Robust Combiners and Universal Constructions for Quantum Cryptography

Taiga Hiroka[⋆], Fuyuki Kitagawa[†], Ryo Nishimaki[†], Takashi Yamakawa[†,⋆]

[⋆]Yukawa Institute for Theoretical Physics, Kyoto University, Japan
taiga.hiroka@yukawa.kyoto-u.ac.jp
[†]NTT Social Informatics Laboratories, Tokyo, Japan
{fuyuki.kitagawa,ryo.nishimaki,takashi.yamakawa}@ntt.com

November 16, 2023

Abstract

A robust combiner combines many candidates for a cryptographic primitive and generates a new candidate for the same primitive. Its correctness and security hold as long as one of the original candidates satisfies correctness and security. A universal construction is a closely related notion to a robust combiner. A universal construction for a primitive is an explicit construction of the primitive that is correct and secure as long as the primitive exists. It is known that a universal construction for a primitive can be constructed from a robust combiner for the primitive in many cases.

Although robust combiners and universal constructions for classical cryptography are widely studied, robust combiners and universal constructions for quantum cryptography have not been explored so far. In this work, we define robust combiners and universal constructions for several quantum cryptographic primitives including one-way state generators, public-key quantum money, quantum bit commitments, and unclonable encryption, and provide constructions of them.

On a different note, it was an open problem how to expand the plaintext length of unclonable encryption. In one of our universal constructions for unclonable encryption, we can expand the plaintext length, which resolves the open problem.

Contents

1 Introduction

1.1 Background

The ultimate goal of theoretical cryptography is to construct interesting cryptographic primitives unconditionally. Over the past years, many computational assumptions have been proposed, and many interesting cryptographic primitives have been constructed under the computational assumptions. However, none of the computational assumptions are proven. Indeed, we do not even know how to prove $\mathbf{P} \neq \mathbf{NP}$ while it is a necessary condition to construct interesting classical cryptographic primitives unconditionally. Moreover, given many candidates for a primitive, we cannot often decide which candidate is the most secure one. For example, we can construct public-key encryption (PKE) from decisional Diffie-Hellman (DDH) [DH76, ElG85] or learning with errors (LWE) [Reg05], but currently, we do not know which computational assumption is the weaker assumption. This causes the problem in the following realistic scenario. Suppose we have two candidates for PKE, where one is based on DDH and the other is based on LWE, and we want to decide more secure candidate to use. Unfortunately, in the current knowledge, we cannot decide which candidate is the more secure one.

A robust cryptographic combiner [Her05, HKN+05] was introduced to resolve this issue. Given many candidates for a primitive, a cryptographic combiner combines these candidates and produces a new candidate for the same primitive. The new candidate is correct and secure as long as at least one of the original candidates satisfies correctness and security. For example, a robust PKE combiner takes two candidates for PKE, where one's security relies on DDH and the other's security relies on LWE, and produces a new candidate for PKE. The new candidate is correct and secure as long as the DDH or LWE assumption holds. Robust combiner is a well-studied topic in classical cryptography. In fact, robust combiners for many fundamental classical cryptographic primitives such as one-way functions, public-key encryption, and functional encryption are shown to exist [HKN+05, AJN+16, AJS17, ABJ+19, JMS20].

A closely related notion to a robust combiner is a universal construction [Lev85]. A universal construction for a primitive, say OWFs, is an explicit construction of OWFs that is correct and secure as long as OWFs exist. The adversary must be able to break all OWF candidates to break a universal construction. In this sense, a universal construction for OWFs is the most secure one among all possible OWF candidates. In classical cryptography, universal constructions are well-studied topic and are known to exist for many fundamental primitives. First, the pioneering work by Levin introduces a notion of universal construction and shows how to construct a universal construction for OWFs [Lev85]. After decades, Harnik, Kilian, Naor, Reingold, and Rosen [HKN+05] give a universal construction for PKE and they show how to construct a universal construction for a primitive using a robust combiner for the same primitive. Goldwasser and Kalai cast questions about universal constructions for cryptographic primitives related to obfuscation [GTK16]. The following sequence of works [AJN+16, AJS17, ABJ+19] gives universal constructions for functional encryption under some assumptions, and [JMS20] gives it unconditionally.

Although robust combiners and universal constructions are widely studied topics in classical cryptography, those in the quantum world have not been studied so far, where each party can generate, process, and communicate quantum information. It is well known that, even in the quantum world, information-theoretical security is impossible to achieve for many interesting quantum cryptographic primitives [LC97, May97, Aar18], and currently, many interesting quantum cryptographic primitives are constructed under computational assumptions. For example, public-key quantum money is one of the most interesting quantum cryptographic primitives, and many candidate constructions are proposed relying on computational assumptions [AC12, FGH+12, Kan18, Zha19, KSS22, LMZ23, Zha23b]. However, none of them have been proven so far, and moreover, we cannot even decide which assumptions are the weakest assumptions. This inability leads to the problem that we cannot decide the most secure one to use.

If there exists a robust public-key quantum money combiner, then we can combine them and produce a new candidate for public-key quantum money, which is secure as long as at least one of the original candidates is secure. Therefore, it is natural to ask the following first question:

Is it possible to construct robust combiners for fundamental quantum cryptographic primitives?

On a different note, recent works show the possibility that quantum cryptography exists even if classical cryptography does not. A pseudo-random state generator (PRSG) is a quantum analog of a pseudo-random generator [JLS18], and Kretchmer shows the possibility that PRSGs exist even if $\mathbf{BQP} = \mathbf{QMA}$ [Kre21]. Many interesting quantum cryptographic primitives are shown to be constructed from PRSGs [MY22b, MY22a, AQY22, AGQY22, BCQ23].

Among them, one-way state generators (OWSGs) and quantum bit commitments (equivalent to EFI [Yan22, BCQ23]) are considered to be candidates for the necessary assumptions for the existence of quantum cryptography. In the case of classical cryptography, many fundamental primitives have the nice feature of the existence of universal constructions. It is natural to wonder whether quantum cryptographic primitives have universal constructions or not. In fact, some researchers believe that the existence of universal constructions is a nice feature for fundamental cryptographic primitives [Zha23a]. Therefore, we ask the following second question:

Is it possible to construct universal constructions for fundamental quantum cryptographic primitives?

1.2 Our Results

We solve the two questions above affirmatively for several cryptographic primitives. Our contributions to the field are as follows:

1. We formally define robust combiners and universal constructions for many quantum cryptographic primitives including OWSGs, public-key quantum money, quantum bit commitments, and unclonable encryption.

2. We construct a robust combiner and a universal construction for OWSGs without any assumptions. A universal construction is secure as long as there exist OWSGs. In other words, the adversary of a universal construction must be able to break all OWSG candidates. In this sense, our construction for OWSG is the most secure one among all possible OWSG candidates. Before this work, the candidate constructions for OWSGs were based on OWFs, average-case hardness of semi-classical quantum statistical difference [CX22] or random quantum circuits [AQY22, BCQ23] [1].

3. We construct a robust combiner and a universal construction for public-key quantum money without any assumptions. In particular, in this work, we consider the public-key quantum money mini-scheme introduced in [AC12], which can be generically upgraded into full-fledged public-key quantum money by additionally using digital signatures. A universal construction for a public-key quantum money mini-scheme satisfies security as long as a public-key quantum money mini-scheme exists. In other words, the adversary of a universal construction must be able to break all candidates for a public-key quantum money mini-scheme. In this sense, our construction is the most secure one among all possible public-key quantum money mini-scheme candidates. Before this work, many candidate constructions are proposed [AC12, FGH+12, Kan18, Zha19, KSS22, LMZ23, Zha23b].

4. We construct a robust combiner and a universal construction for quantum bit commitment without any assumptions. Note that our results also imply that we can construct a robust combiner and a universal construction for EFI, oblivious transfer, and multi-party computation, which are equivalent to quantum bit commitments [BCQ23]. In our robust combiner, given n-candidates of quantum bit commitments, we can construct a new quantum bit commitment that satisfies statistical binding and computational hiding at least one of n-candidates satisfies computational hiding and computational binding at the same time. A universal construction for quantum bit commitment is secure as long as there exists a quantum bit commitment. In other words, the adversary for a universal construction must be able to break all candidates for quantum bit commitment. In this sense, our construction for quantum bit commitment is the most secure one among all possible quantum bit commitment candidates. Before this work, candidate constructions of quantum bit commitments were based on OWFs, classical oracle [KQST23], or random quantum circuits [AQY22, BCQ23] [2].

5. We construct robust combiners and universal constructions for various kinds of unclonable encryption as follows:

    - We construct robust combiners for (one-time) unclonable secret-key encryption (SKE) and unclonable public-key encryption (PKE) without any computational assumptions.

[1] As discussed in the previous works [AQY22, BCQ23], it is a folklore that a random quantum circuit is PRSGs although there exists no theoretical evidence so far. Since we can construct OWSGs from PRSGs [MY22b, MY22a], we can also construct OWSGs based on random quantum circuits if a random quantum circuit is PRSGs.

[2] It is a folklore that a random quantum circuit is PRSGs although there exists no theoretical evidence so far. Since we can construct quantum bit commitments from PRSGs [MY22b, AQY22], we can also construct quantum bit commitments based on random quantum circuits if a random quantum circuit is PRSGs.

- By using robust combiners, we construct universal constructions for (one-time) unclonable SKE and unclonable PKE without any computational assumptions.

Although the previous work [AKL+22] gives a construction of one-time unclonable SKE with unclonable IND-CPA security in the quantum random oracle model (QROM), it was an open problem to construct it in the standard model. Our universal constructions for (one-time) unclonable SKE (resp. PKE) is the first construction of (one-time) unclonable SKE (resp. PKE) that achieves unclonable IND-CPA security in the standard model, where the security relies on the existence of (one-time) unclonable SKE (resp. PKE) with unclonable IND-CPA security.

6. We give another construction of universal construction for one-time unclonable SKE by additionally using the decomposable quantum randomized encoding [BY22]. Although this construction additionally uses decomposable quantum randomized encoding, it has the following nice three properties that the universal construction via a robust combiner does not have:

   - It was an open problem whether unclonable encryption with single-bit plaintexts implies unclonable encryption with multi-bit plaintexts because standard transformation via bit-wise encryption does not work as pointed out in [AKL+22]. In our universal construction, we can expand the plaintext length of one-time unclonable SKE by additionally using decomposable quantum randomized encoding. This resolves the open problem left by [AKL+22]. Note that this result implies that reusable unclonable SKE and unclonable PKE can expand plaintext length without any additional assumptions because reusable unclonable SKE and unclonable PKE imply decomposable quantum randomized encoding.
   - A universal construction via a robust combiner needs to emulate all possible algorithms, and thus a huge constant is included in the running time. Therefore, it may not be executed in a meaningful amount of time if we want reasonable concrete security. On the other hand, universal construction via decomposable quantum randomized encoding does not emulate all possible algorithms and thus avoids the "galactic inefficiency" tied to such approaches.
   - In a universal construction via a robust combiner, the security relies on the existence of one-time unclonable SKE scheme $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, where $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ are uniform QPT algorithms. On the other hand, in a universal construction via decomposable quantum randomized encoding, the security still holds even if the underlying one-time unclonable SKE $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ are non-uniform algorithms.

1.3 More on Related Work

Fundamental Quantum Cryptographic Primitives. Ji, Liu, and Song [JLS18] introduce a notion of PRSGs, and show that it can be constructed from OWFs. Morimae and Yamakawa [MY22b] introduce the notion of OWSGs, and show how to construct them from PRSGs. In the first definition of OWSGs, the output quantum states are restricted to pure states, and its definition is generalized to mixed states by [MY22a]. In this work, we focus on the mixed-state version.

Bennett and Brassard [BB84] initiate the study of quantum bit commitment. Unfortunately, it turns out that statistically secure quantum bit commitments are impossible to achieve [LC97, May97]. Therefore, later works study a quantum bit commitment with computational security [DMS00, CLS01, Yan22, MY22b, MY22a, AQY22, AGQY22, BCQ23, HMY23]. It was shown that quantum bit commitments can be constructed from PRSGs by [MY22b, AQY22], and that quantum bit commitments are equivalent to EFI, oblivious transfer, and multi-party computation [GLSV21, BCKM21, Yan22, BCQ23].

Recently, Khurana and Tomer [KT23] showed that quantum bit commitments can be constructed from OWSGs with pure state. Although their main result is not a combiner for quantum bit commitment, they construct some sort of a combiner for quantum bit commitments as an intermediate tool for achieving their result. In their construction, they construct a uniform quantum bit commitment from a non-uniform one. At this step, they combine quantum bit commitments in the following sense. In their construction, they combine $(n+1)$-quantum bit commitments and generate a new quantum bit commitment. Its hiding and binding property holds as long as one of the original candidates satisfies hiding and binding at the same time and other n candidates also satisfy either hiding or binding. Compared to their

technique, our robust combiner does not need to assume other n candidates satisfy hiding or binding. Therefore, our robust combiner can be applied in a more general setting than their technique. Though our construction partially shares a similarity with theirs, we rely on additional ideas to deal with candidate schemes that do not satisfy either binding or hiding.

Unclonable Encryption. Broadbent and Lord [BL20] introduced a notion of unclonable encryption. They considered two security definitions for unclonable encryption. One is one-wayness against cloning attacks 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, Kaleoglu, Li, Liu, and Zhandry [AKL+22] proposed the first indistinguishable-secure unclonable encryption in the QROM. Ananth and Kaleoglu [AK21] construct unclonable PKE from unclonable encryption and PKE with "classical" ciphertexts. Note that it is unclear how to apply their technique for PKE with quantum ciphertexts. The technique of [HMNY21] can be used to construct unclonable PKE from unclonable encryption and PKE with quantum ciphertexts, which we use in this work.

Combiner for Classical Cryptography. It is known that robust combiners are known to exist for many fundamental classical cryptographic primitives. Oblivious transfer (OT) is an example of exceptions. It is an open problem how to construct a robust combiner for classical OT and some black-box impossibilities are known [HKN+05]. Interestingly, our result implies that a robust combiner for quantum OT exists although a robust combiner for classical OT is still an open problem.

1.4 Organization

In Section 2, we give a technical overview. In Section 3, we define the notations and preliminaries that we require in this work. In Section 4, we define the notions of robust OWSG combiners and a universal construction for OWSGs and provide constructions. We provide some proof in Appendix A. In Section 5, we define the notions of a robust combiner and a universal construction for public-key quantum money mini-scheme and provide constructions. We provide some proof in Appendix B. In Section 6, we define the notions of a robust canonical quantum bit commitment combiner and a universal construction for canonical quantum bit commitment and provide constructions. We provide some proof in Appendix C. In Section 7, we define the notions of robust combiners for unclonable encryption and universal constructions for unclonable encryption and provide constructions. We provide some proof in Appendices D and E. In Section 8, we provide another universal construction for unclonable encryption. We provide some proof in Appendix F. In this construction, we can expand the plaintext length of unclonable encryption.

2 Technical Overview

First of all, let us recall the definition of robust combiner. A robust combiner for a primitive P is a deterministic classical polynomial-time Turing machine $\mathsf{RobComb}.\mathcal{M}_P$ that takes as input n-candidates $\{\Sigma[i]\}_{i\in[n]}$ for P, and produces a new candidate Σ for P. Σ is correct and secure as long as at least one of the candidates $\{\Sigma[i]\}_{i\in[n]}$ for P is correct and secure. Here, the point is that $\{\Sigma[i]\}_{i\in[n]}$ are not promised to satisfy even correctness other than one of them. In the following, we will explain the case where only two candidates $\Sigma[1]$ and $\Sigma[2]$ are given for simplicity. Remark that the same argument goes through in the general case, where n candidates $\{\Sigma[i]\}_{i\in[n]}$ are given.

2.1 Robust Combiner for One-Way State Generators and Public-Key Quantum Money

In this section, we explain a robust combiner for OWSGs. In the same way as OWSGs, a robust combiner for public-key quantum money can be constructed.

Definition of One-Way State Generators. OWSG is a quantum generalization of OWFs and consists of a tuple of quantum polynomial-time algorithms $\Sigma_{\mathsf{OWSG}} := (\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Vrfy})$. The KeyGen algorithm takes as input a security parameter 1^λ, and generates a classical key k, the $\mathsf{StateGen}$ algorithm takes as input a classical key k and outputs a quantum state ψ_k, and the Vrfy algorithm takes as input a classical key k and a quantum state ψ_k and outputs 1 indicating acceptance or 0 indicating rejection. We require that OWSG Σ satisfies correctness and security. The correctness guarantees that $\mathsf{Vrfy}(k, \psi_k)$ outputs 1 indicating acceptance with overwhelming probability, where $k \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $\psi_k \leftarrow \mathsf{StateGen}(k)$. The security guarantees that no QPT adversaries given polynomially many copies of ψ_k cannot generate k^* such that $1 \leftarrow \mathsf{Vrfy}(k^*, \psi_k)$, where $k \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $\psi_k \leftarrow \mathsf{StateGen}(k)$.

Robust Combiner. First, we consider the simpler case, where given OWSG candidates $\Sigma_{\mathsf{OWSG}}[1] = (\mathsf{KeyGen}[1], \mathsf{StateGen}[1], \mathsf{Vrfy}[1])$ and $\Sigma_{\mathsf{OWSG}}[2] = (\mathsf{KeyGen}[2], \mathsf{StateGen}[2], \mathsf{Vrfy}[2])$ are promised to satisfy at least correctness. In this case, we can construct a combiner for OWSGs in the same way as OWFs. Namely, a combined protocol $\mathsf{Comb}.\Sigma_{\mathsf{OWSG}} = (\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Vrfy})$ simply runs $\Sigma[1]$ and $\Sigma[2]$ in parallel.

Does the same strategy work for the general setting, where original candidates are not promised to satisfy correctness? Unfortunately, the simple parallel protocol works only when both $\Sigma_{\mathsf{OWSG}}[1]$ and $\Sigma_{\mathsf{OWSG}}[2]$ satisfy correctness because $\mathsf{Comb}.\Sigma_{\mathsf{OWSG}}$ does not satisfy correctness otherwise. We observe that given an OWSG candidate Σ_{OWSG}, we can construct Σ^*_{OWSG} with the following properties:

- Σ^*_{OWSG} satisfies correctness regardless of Σ_{OWSG}.
- Σ^*_{OWSG} satisfies security as long as Σ_{OWSG} satisfies correctness and security.

Once we have obtained such a transformation, we can construct a robust OWSG combiner $\mathsf{RobComb}.\mathcal{M}_{\mathsf{OWSG}}$ as follows. Given two OWSGs candidates $\Sigma_{\mathsf{OWSG}}[1]$ and $\Sigma_{\mathsf{OWSG}}[2]$, our robust OWSG combiner $\mathsf{RobComb}.\mathcal{M}_{\mathsf{OWSG}}$ first transforms them into $\Sigma_{\mathsf{OWSG}}[1]^*$ and $\Sigma_{\mathsf{OWSG}}[2]^*$, respectively, and then outputs $\mathsf{Comb}.\Sigma_{\mathsf{OWSG}}$ which runs $\Sigma_{\mathsf{OWSG}}[1]^*$ and $\Sigma_{\mathsf{OWSG}}[2]^*$ in parallel. $\mathsf{Comb}.\Sigma_{\mathsf{OWSG}}$ satisfies correctness because $\Sigma_{\mathsf{OWSG}}[1]^*$ and $\Sigma_{\mathsf{OWSG}}[2]^*$ satisfies correctness no matter what $\Sigma_{\mathsf{OWSG}}[1]$ and $\Sigma_{\mathsf{OWSG}}[2]$ are. $\mathsf{Comb}.\Sigma_{\mathsf{OWSG}}$ satisfies security as long as either $\Sigma_{\mathsf{OWSG}}[1]$ or $\Sigma_{\mathsf{OWSG}}[2]$ satisfy correctness and security because either $\Sigma_{\mathsf{OWSG}}[1]^*$ or $\Sigma_{\mathsf{OWSG}}[2]^*$ satisfies security as long as either $\Sigma_{\mathsf{OWSG}}[1]$ or $\Sigma_{\mathsf{OWSG}}[2]$ satisfies correctness and security.

Transform Incorrect Candidate into Correct One. Now, we consider how to obtain such a transformation. In the previous work [HKN+05], it was shown that we can transform PKE Σ_{PKE} into Σ^*_{PKE} that satisfies correctness regardless of Σ_{PKE} and satisfies security as long as Σ_{PKE} satisfies correctness and security. In the same way as [HKN+05], we can obtain such transformation for OWSGs. However, in this work, we take a different approach because the technique by [HKN+05] does not work for unclonable encryption.

First, we observe that without loss of generality, $\mathsf{Vrfy}(k, \psi)$ can be considered working as follows: It appends $|0\rangle\langle 0|$ to ψ, applies U_k to $\psi \otimes |0\rangle\langle 0|$, measures the first qubit of $U_k(\psi \otimes |0\rangle\langle 0|)U_k^\dagger$, and outputs the measurement outcome. Now, we describe $\Sigma^*_{\mathsf{OWSG}} = (\mathsf{KeyGen}^*, \mathsf{StateGen}^*, \mathsf{Vrfy}^*)$. KeyGen^* is the same as the original KeyGen. $\mathsf{StateGen}^*(k)$ first runs $\psi_k \leftarrow \mathsf{StateGen}(k)$, then measures the first qubit of $U_k(\psi_k \otimes |0\rangle\langle 0|)U_k^\dagger$ in the computational basis, and obtains b. If $b = 1$, $\mathsf{StateGen}^*(k)$ rewinds its register and outputs the register as ψ^*_k. Otherwise, output $\psi^*_k = \bot$, where $\bot$ is a special symbol. $\mathsf{Vrfy}^*(k, \psi)$ first checks the form of ψ. If $\psi = \bot$, $\mathsf{Vrfy}^*(k, \psi)$ outputs 1. Otherwise, $\mathsf{Vrfy}^*(k, \psi)$ applies U_k to ψ, then measures the first qubit of $U_k\psi U_k^\dagger$, and finally outputs the measurement outcome. We can see that Σ^* satisfies correctness. If $\mathsf{StateGen}^*(k)$ outputs $\psi^*_k = \bot$, then Vrfy^* always outputs 1. On the other hand, if $\psi^* \neq \bot$, then $\mathsf{StateGen}^*(k)$ outputs ψ^*_k with the form $U_k^\dagger(|1\rangle\langle 1| \otimes \rho)U_k$ for some quantum state ρ. Therefore, $\mathsf{Vrfy}^*(k, \psi^*_k)$ outputs 1 since $U_k\psi^*_k U_k^\dagger = |1\rangle\langle 1| \otimes \rho$. Moreover, we can see that Σ^* satisfies security as long as Σ satisfies correctness and security. As long as Σ satisfies correctness, if we measure the first qubits of $U_k(\psi_k \otimes |0\rangle\langle 0|)U_k^\dagger$ in the computational basis, then the measurement result is 1 with overwhelming probability, where $k \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $\psi_k \leftarrow \mathsf{StateGen}(k)$. This indicates that the measurement does not disturb the quantum state $U_k(\psi_k \otimes |0\rangle\langle 0|)U_k^\dagger$ from gentle measurement lemma. Therefore, ψ^*_k is statistically close to $\psi_k \otimes |0\rangle\langle 0|$ as long as Σ satisfies correctness. In particular, this implies that we can reduce the security of Σ^* to that of Σ as long as Σ satisfies correctness.

2.2 Robust Combiner for Unclonable Encryption

In this section, we explain how to obtain a robust combiner for unclonable SKE. As a corollary, we can obtain a robust combiner for unclonable PKE. This is because we can construct unclonable PKE from unclonable SKE and PKE with quantum ciphertexts [HMNY21, AK21], and a robust combiner for PKE with quantum ciphertexts can be constructed in the same way as the classical ciphertexts case [HKN+05].

Definition of Unclonable SKE. First of all, we explain the definition of unclonable SKE. Unclonable SKE $\Sigma_{\mathsf{unclone}}$ is the same as standard SKE Σ_{SKE} except that the ciphertext of unclonable SKE is a quantum state and it satisfies unclonable IND-CPA security in addition to standard IND-CPA security. In unclonable IND-CPA security, the cloning adversary $\mathcal{A}$ with oracle $\mathsf{Enc}(\mathsf{sk},\cdot)$ first sends the challenge plaintext (m_0,m_1), then receives a ciphertext CT_b, where $\mathsf{CT}_b \leftarrow \mathsf{Enc}(\mathsf{sk},m_b)$, and finally generates a quantum state $\rho_{\mathcal{B},\mathcal{C}}$ over the $\mathcal{B}$ and $\mathcal{C}$ registers. The adversary $\mathcal{B}$ (resp. $\mathcal{C}$) receives the $\mathcal{B}$ register (resp. the $\mathcal{C}$ register) and the secret-key sk, and outputs $b_{\mathcal{B}}$ (resp. $b_{\mathcal{C}}$) which is a guess of b. The unclonable IND-CPA security guarantees that for any QPT adversaries $(\mathcal{A},\mathcal{B},\mathcal{C})$, we have

$$\Pr[b=b_{\mathcal{B}}=b_{\mathcal{C}}] \leq \frac{1}{2}+\mathsf{negl}(\lambda).$$

Robust Combiner. First, we consider the simpler case, where given candidates $\Sigma_{\mathsf{unclone}}[1]=(\mathsf{KeyGen}[1],\mathsf{Enc}[1],\mathsf{Dec}[1])$ and $\Sigma_{\mathsf{unclone}}[2]=(\mathsf{KeyGen}[2],\mathsf{Enc}[2],\mathsf{Dec}[2])$ are promised to satisfy at least correctness. In that case, a combined unclonable SKE scheme $\mathsf{Comb}.\Sigma_{\mathsf{unclone}}=(\mathsf{KeyGen},\mathsf{Enc},\mathsf{Dec})$ simply runs $\Sigma_{\mathsf{unclone}}[1]$ and $\Sigma_{\mathsf{unclone}}[2]$ by using X-OR secret sharing. In other words, for encrypting bit b, $\mathsf{Comb}.\Sigma_{\mathsf{unclone}}$ first samples $r[1]$ and $r[2]$ such that $r[1]+r[2]=b$, and encrypts $r[1]$ by using $\Sigma_{\mathsf{unclone}}[1]$ and $r[2]$ by using $\Sigma_{\mathsf{unclone}}[2]$. Clearly, $\mathsf{Comb}.\Sigma_{\mathsf{unclone}}$ satisfies correctness and security as long as both $\Sigma_{\mathsf{unclone}}[1]$ and $\Sigma_{\mathsf{unclone}}[2]$ satisfy correctness and either $\Sigma_{\mathsf{unclone}}[1]$ or $\Sigma_{\mathsf{unclone}}[2]$ satisfies security.

Does the same strategy work for the general setting, where original candidates are not promised to satisfy even correctness? Unfortunately, the simple X-OR protocol above works only when both $\Sigma_{\mathsf{unclone}}[1]$ and $\Sigma_{\mathsf{unclone}}[2]$ satisfy correctness because $\mathsf{Comb}.\Sigma_{\mathsf{unclone}}$ does not satisfy correctness otherwise. Our key observation is that given a candidate of unclonable SKE $\Sigma_{\mathsf{unclone}}$ we can construct a new candidate $\Sigma^*_{\mathsf{unclone}}$ with the following properties:

- $\Sigma^*_{\mathsf{unclone}}$ satisfies correctness regardless of $\Sigma_{\mathsf{unclone}}$.
- $\Sigma^*_{\mathsf{unclone}}$ satisfies security as long as Σ satisfies correctness and security.

Once we have obtained such a transformation, we can construct a robust combiner for unclonable SKE as follows. Given two unclonable SKE candidates $\Sigma_{\mathsf{unclone}}[1]$ and $\Sigma_{\mathsf{unclone}}[2]$, a robust combiner for unclonable SKE first transforms $\Sigma_{\mathsf{unclone}}[1]$ and $\Sigma_{\mathsf{unclone}}[2]$ into $\Sigma_{\mathsf{unclone}}[1]^*$ and $\Sigma_{\mathsf{unclone}}[2]^*$, respectively, and then outputs $\mathsf{Comb}.\Sigma_{\mathsf{unclone}}$ which runs $\Sigma_{\mathsf{unclone}}[1]^*$ and $\Sigma_{\mathsf{unclone}}[2]^*$ by using X-OR secret sharing. $\mathsf{Comb}.\Sigma_{\mathsf{unclone}}$ satisfies correctness because $\Sigma_{\mathsf{unclone}}[1]^*$ and $\Sigma_{\mathsf{unclone}}[2]^*$ satisfy correctness no matter what $\Sigma_{\mathsf{unclone}}[1]$ and $\Sigma_{\mathsf{unclone}}[2]$ are. Moreover, $\mathsf{Comb}.\Sigma_{\mathsf{unclone}}$ satisfies security as long as either $\Sigma_{\mathsf{unclone}}[1]$ or $\Sigma_{\mathsf{unclone}}[2]$ satisfies correctness and security. This is because either $\Sigma^*_{\mathsf{unclone}}[1]$ or $\Sigma_{\mathsf{unclone}}[2]^*$ satisfies security as long as either $\Sigma_{\mathsf{unclone}}[1]$ or $\Sigma_{\mathsf{unclone}}[2]$ satisfies correctness and security.

Transform Incorrect Candidate into Correct One. Now, we consider how to obtain such a transformation. It is known that we can obtain such a transformation for PKE [HKN+05]. In their technique, they use parallel repetition to amplify correctness. We emphasize that we cannot apply their technique for unclonable encryption because correctness amplification via parallel repetition does not work for unclonable encryption. Therefore, we take a different approach, whose idea is the same as OWSGs. Without loss of generality, we can assume that $\mathsf{Dec}(\mathsf{sk},\mathsf{CT})$ first appends $|0\rangle\langle 0|$ to CT, applies U_{sk} to $\mathsf{CT}\otimes|0\rangle\langle 0|$, measures the first $|m|$-bit of $U_{\mathsf{sk}}(\mathsf{CT}\otimes|0\rangle\langle 0|)U_{\mathsf{sk}}^\dagger$, and outputs the measurement outcome. Now, we describe $\Sigma^*_{\mathsf{unclone}}=(\mathsf{KeyGen}^*,\mathsf{Enc}^*,\mathsf{Dec}^*)$. KeyGen^* is the same as the original KeyGen. $\mathsf{Enc}^*(\mathsf{sk},m)$ first runs $\mathsf{CT}\leftarrow\mathsf{Enc}(\mathsf{sk},m)$, then measures the first $|m|$-bit of $U_{\mathsf{sk}}(\mathsf{CT}\otimes|0\rangle\langle 0|)U_{\mathsf{sk}}^\dagger$ in the computational basis, obtains m^*, and checks whether $m=m^*$. If $m=m^*$, $\mathsf{Enc}^*(\mathsf{sk},\mathsf{CT})$ rewinds its register and outputs the register as the quantum ciphertext CT^*. Otherwise, output $\mathsf{CT}^*=(\bot,m)$, where $\bot$ is a special symbol. $\mathsf{Dec}^*(\mathsf{sk},\mathsf{CT}^*)$ first checks the form of CT^*, and outputs m if CT^* is of the form $(\bot,m)$. Otherwise, $\mathsf{Dec}^*(\mathsf{sk},\mathsf{CT}^*)$ applies U_{sk} to CT^*, and outputs the

measurement outcome of first $|m|$-qubits of $U_{\mathsf{sk}}\mathsf{CT}^*U_{\mathsf{sk}}^\dagger$. Clearly, the new construction $\Sigma^*_{\mathsf{unclone}}$ satisfies correctness in the same reason as OWSG. Furthermore, $\Sigma^*_{\mathsf{unclone}}$ satisfies security as long as $\Sigma_{\mathsf{unclone}}$ satisfies correctness and security. This is because CT^* is statistically close to $\mathsf{CT} \otimes |0\rangle\langle 0|$ as long as $\Sigma_{\mathsf{unclone}}$ satisfies correctness, and thus we can reduce the security of $\Sigma^*_{\mathsf{unclone}}$ to that of $\Sigma_{\mathsf{unclone}}$.

2.3 Robust Combiner for Quantum Bit Commitment

Definition of Quantum Bit Commitment. In the following, we consider a robust combiner for quantum bit commitment. In this work, we consider a canonical quantum bit commitment. Any quantum bit commitment can be written in the following canonical form [Yan22]. A canonical quantum bit commitment scheme is a pair of unitaries (Q_0, Q_1) acting on the registers $\mathbf{C}$ called the commitment register and $\mathbf{R}$ called the reveal register, and works as follows.

Commit Phase: A sender runs $Q_b |0\rangle_{\mathbf{C},\mathbf{R}}$ and sends the $\mathbf{C}$ to a receiver for committing a bit $b \in \{0, 1\}$.

Reveal Phase: For revealing the committed bit b, the sender sends b and the $\mathbf{R}$ register to the receiver. The receiver applies $Q_b^\dagger$ to the $\mathbf{C}$ and $\mathbf{R}$ register and measures both registers in the computational basis. The receiver accepts if the measurement outcomes are all 0, and rejects otherwise.

We require that a canonical quantum bit commitment satisfies hiding and binding. The computational (resp. statistical) hiding requires that no quantum polynomial-time (resp. unbounded) adversaries distinguish $Q_0 |0\rangle_{\mathbf{C},\mathbf{R}}$ from $Q_1 |0\rangle_{\mathbf{C},\mathbf{R}}$ without touching the $\mathbf{R}$ register with non-negligible probability.

The binding requires that no adversaries can map an honestly generated quantum bit commitment of 0 (i.e. $Q_0 |0\rangle_{\mathbf{C},\mathbf{R}}$) to that of 1 (i.e. $Q_1 |0\rangle_{\mathbf{C},\mathbf{R}}$) without touching $\mathbf{C}$ registers. More formally, computational (resp. statistical) binding requires that for any quantum polynomial-time (resp. unbounded) unitary $U_{\mathbf{R},\mathbf{Z}}$ acting on the $\mathbf{R}$ and $\mathbf{Z}$ register and any quantum state $|\tau\rangle_{\mathbf{Z}}$ on $\mathbf{Z}$ register, we have

$$\left\| (Q_1 |0\rangle\langle 0| Q_1^\dagger)_{\mathbf{C},\mathbf{R}} (I_{\mathbf{C}} \otimes U_{\mathbf{R},\mathbf{Z}})(Q_0 |0\rangle_{\mathbf{C},\mathbf{R}} |\tau\rangle_{\mathbf{Z}}) \right\| \leq \mathsf{negl}(\lambda).$$

It was shown that we can change the flavor of quantum bit commitment [Yan22, HMY23]. More formally, if we have a canonical quantum bit commitment (Q_0, Q_1) that satisfies X-hiding and Y-binding, then we can construct a canonical quantum bit commitment $(\widetilde{Q_0}, \widetilde{Q_1})$ that satisfies X-binding and Y-hiding for $X, Y \in \{\text{statistical, computational}\}$.

Robust Combiner. First, let us clarify our final goal. Given two candidates of canonical quantum bit commitments $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$, our robust combiner $\mathsf{RobComb}.\mathcal{M}_{\mathsf{Commit}}$ generates a new candidate $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ that satisfies hiding and binding as long as either $(Q_0[1], Q_1[1])$ or $(Q_0[2], Q_1[2])$ satisfies hiding and binding. More formally, our robust combiner $\mathsf{RobComb}.\mathcal{M}_{\mathsf{Commit}}$ outputs $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ with the following properties:

- $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ satisfies statistical binding regardless of $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$.
- $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ satisfies computational hiding as long as either $(Q_0[1], Q_1[1])$ or $(Q_0[2], Q_1[2])$ satisfies computational hiding and computational binding.

To achieve this final goal, let us consider the following simpler goal first, where both candidates $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$ satisfy at least statistical binding. More formally, given candidates $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$, we consider constructing a new candidate $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ with the following properties:

- $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ satisfies statistical binding as long as both $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$ satisfies statistical binding.
- $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ satisfies computational hiding as long as either $(Q_0[1], Q_1[1])$ or $(Q_0[2], Q_1[2])$ satisfies computational hiding.

We can construct such $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ by simply using X-OR secret sharing. More formally, for $b \in \{0,1\}$, $\mathsf{Comb}.Q_b$ first samples $r[1]$ and $r[2]$ conditioned on $r[1] + r[2] = b$, and then commits $r[1]$ by using $(Q_0[1], Q_1[1])$ and commits $r[2]$ by using $(Q_0[2], Q_1[2])$. Our construction satisfies statistical binding as long as both $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$ satisfy statistical binding. The intuitive reason is that the adversary of $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ needs to change $r[1]$ or $r[2]$ after sending the commitment register to break binding of $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$, but the adversary cannot do this because both $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$ satisfy statistical binding. Furthermore, $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ satisfies computational hiding as long as either $(Q_0[1], Q_1[1])$ or $(Q_0[2], Q_1[2])$ satisfies computational hiding. The intuitive reason is that the adversary of $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ needs to obtain both $r[1]$ and $r[2]$ from the commitment register of $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$, but the adversary cannot do this because either $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$ satisfies computational hiding.

Does the same strategy work for a robust quantum bit commitment combiner $\mathsf{RobComb}.\mathcal{M}_{\mathsf{Commit}}$? Unfortunately, the simple X-OR protocol above works only when both $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$ satisfy statistical binding because $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ does not satisfy statistical binding otherwise. Our key observation is that, given a candidate of canonical quantum bit commitment (Q_0, Q_1), we can construct a new candidate (Q_0^*, Q_1^*) that satisfies at least statistical binding regardless of (Q_0, Q_1). More formally, we can construct (Q_0^*, Q_1^*) with the following properties:

- (Q_0^*, Q_1^*) satisfies statistical binding regardless of (Q_0, Q_1).
- (Q_0^*, Q_1^*) satisfies computational hiding if (Q_0, Q_1) satisfies computational hiding and computational binding.

Once we have obtained such a transformation, we can construct a robust quantum bit commitment combiner $\mathsf{RobComb}.\mathcal{M}_{\mathsf{Commit}}$. Given two candidates of canonical quantum bit commitment $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$, $\mathsf{RobComb}.\mathcal{M}_{\mathsf{Commit}}$ first transforms $(Q_0[1], Q_1[1])$ and $(Q_0[2], Q_1[2])$ into $(Q_0[1]^*, Q_1[1]^*)$ and $(Q_0[2]^*, Q_1[2]^*)$, respectively and then outputs $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$, which runs $(Q_0[1]^*, Q_1[1]^*)$ and $(Q_0[2]^*, Q_1[2]^*)$ by using X-OR secret sharing. Clearly, $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ satisfies statistical binding. Moreover, $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ satisfies computational hiding as long as either $(Q_0[1], Q_1[1])$ or $(Q_0[2], Q_1[2])$ satisfies computational hiding and computational binding.

Transform Candidate without Statistical Binding into One with Statistical Binding. Now, we consider how to obtain such a transformation. Our first observation is that either (Q_0, Q_1) or $(\widetilde{Q_0}, \widetilde{Q_1})$, which is the flavor conversion of (Q_0, Q_1) obtained by [HMY23], satisfies statistical binding in a possibly weak sense. To see this let us denote $\rho_b \coloneqq \mathrm{Tr}_{\mathbf{R}}(Q_b |0\rangle_{\mathbf{C},\mathbf{R}})$. Then, there exists some constant f such that

$$F(\rho_0, \rho_1) = f,$$

where $F(\rho_0, \rho_1)$ is the fidelity between ρ_0 and ρ_1. If f is small, then (Q_0, Q_1) satisfies statistical binding in a possibly weak sense from Uhlmann's theorem. On the other hand, if f is large, then (Q_0, Q_1) does not satisfy statistical binding, but $(\widetilde{Q_0}, \widetilde{Q_1})$ satisfies statistical binding instead. This is because if f is large, then (Q_0, Q_1) satisfies statistical hiding, and thus $(\widetilde{Q_0}, \widetilde{Q_1})$ satisfies statistical binding. Therefore, either (Q_0, Q_1) or $(\widetilde{Q_0}, \widetilde{Q_1})$ satisfies statistical binding in a possibly weak sense regardless of (Q_0, Q_1). Furthermore, we observe that such a possibly weak binding property can be amplified to a strong one by parallel repetition.

Based on these observations, we construct our transformation. Given a candidate of canonical quantum bit commitment (Q_0, Q_1), our transformation outputs a new candidate (Q_0^*, Q_1^*) working as follows.

- If we write $\mathbf{C}$ and $\mathbf{R}$ to mean the commitment register and the reveal register of (Q_0, Q_1), and write $\widetilde{\mathbf{C}}$ and $\widetilde{\mathbf{R}}$ to mean the commitment and the reveal register of $(\widetilde{Q_0}, \widetilde{Q_1})$, then the commitment register $\mathbf{C}^*$ of (Q_0^*, Q_1^*) is $(\mathbf{C}^{\otimes\lambda}, \widetilde{\mathbf{C}}^{\otimes\lambda})$, and the reveal register $\mathbf{R}^*$ of (Q_0^*, Q_1^*) is $(\mathbf{R}^{\otimes\lambda}, \widetilde{\mathbf{R}}^{\otimes\lambda})$.
- For $b \in \{0,1\}$, Q_b^* works as follows:

  $$Q_b^* \coloneqq (Q_b \otimes \widetilde{Q_b})^{\otimes\lambda}.$$

  Note that we have

  $$Q_b^* |0\rangle_{\mathbf{C}^*,\mathbf{R}^*} = (Q_b |0\rangle_{\mathbf{C},\mathbf{R}})^{\otimes\lambda} \otimes (\widetilde{Q_b} |0\rangle_{\widetilde{\mathbf{C}},\widetilde{\mathbf{R}}})^{\otimes\lambda}.$$

We can see that (Q_0^*, Q_1^*) satisfies statistical binding regardless of (Q_0, Q_1). If we write $\rho_b := \mathrm{Tr}_{\mathbf{R}}(Q_b |0\rangle_{\mathbf{C},\mathbf{R}})$, there exists some constant $0 \leq f \leq 1$ such that

$$F(\rho_0, \rho_1) = f.$$

If we write $\widetilde{\rho_b} := \mathrm{Tr}_{\widetilde{\mathbf{R}}}(\widetilde{Q_b} |0\rangle_{\widetilde{\mathbf{C}},\widetilde{\mathbf{R}}})$, then we can show that

$$F(\widetilde{\rho_0}, \widetilde{\rho_1}) \leq (1-f)^{1/2}$$

by using the technique by [HMY23]. Therefore, if we write $\rho_b^* := \mathrm{Tr}_{\mathbf{R}^*}(Q_b^* |0\rangle_{\mathbf{C}^*,\mathbf{R}^*})$, we have

$$F(\rho_0^*, \rho_1^*) = F((\rho_0 \otimes \widetilde{\rho_0})^{\otimes\lambda}, (\rho_1 \otimes \widetilde{\rho_1})^{\otimes\lambda}) \leq F(\rho_0, \rho_1)^\lambda F(\widetilde{\rho_0}, \widetilde{\rho_1})^\lambda \leq f^\lambda (1-f)^{\lambda/2} \leq 2^{-\lambda/2}.$$

This implies that (Q_0^*, Q_1^*) satisfies statistical binding regardless of (Q_0, Q_1) from Uhlmann's Theorem.

Moreover, we can see that (Q_0^*, Q_1^*) satisfies computational hiding as long as (Q_0, Q_1) satisfies computational hiding and computational binding. The hiding QPT adversary of (Q_0^*, Q_1^*) needs to obtain b from $\rho_b^* = (\rho_b \otimes \widetilde{\rho_b})^{\otimes\lambda}$. For that, the adversary needs to obtain b from ρ_b or $\widetilde{\rho_b}$. Because (Q_0, Q_1) satisfies computational hiding, the QPT adversary cannot obtain b from ρ_b. Furthermore, $(\widetilde{Q_0}, \widetilde{Q_1})$ also satisfies computational hiding because $(\widetilde{Q_0}, \widetilde{Q_1})$ is a flavor conversion of (Q_0, Q_1). Therefore, the QPT adversary cannot obtain b from $\widetilde{\rho_b}$.

2.4 Universal Constructions

Let us recall the definition of universal construction. A universal construction for a primitive P is an explicit construction of P, which satisfies correctness and security as long as P exists. In this section, we explain how to provide universal constructions via robust combiners. In particular, we explain how to construct a universal construction for OWSGs by using a robust OWSG combiner $\mathsf{RobComb}.\mathcal{M}_{\mathsf{OWSG}}$. We can give universal constructions for other cryptographic primitives in the same way.

In a nutshell, the idea of universal construction via robust combiner [HKN+05] is to think of all descriptions of algorithms as OWSG candidates and combine them. For a set of classical Turing machines $\mathcal{M} = (x, y, z)$, we write $(\mathsf{KeyGen}[x], \mathsf{StateGen}[y], \mathsf{Vrfy}[z])$ to mean a OWSG candidate described by (x, y, z). For simplicity, we assume that $(\mathsf{KeyGen}[x], \mathsf{StateGen}[y], \mathsf{Vrfy}[z])$ are efficient for all $x, y, z \in \mathbb{N}$. The universal construction $(\mathsf{KeyGen}_{\mathsf{Univ}}(1^\lambda), \mathsf{StateGen}_{\mathsf{Univ}}(k), \mathsf{Vrfy}_{\mathsf{Univ}}(k, \psi_k))$ works as follows:

- $\mathsf{KeyGen}_{\mathsf{Univ}}(1^\lambda)$ first runs

  $$(\mathsf{KeyGen}_\lambda, \mathsf{StateGen}_\lambda, \mathsf{Vrfy}_\lambda) \leftarrow \mathsf{RobComb}.\mathcal{M}_{\mathsf{OWSG}}(\{\mathsf{KeyGen}[x], \mathsf{StateGen}[y], \mathsf{Vrfy}[z]\}_{x,y,z\in[\lambda]}),$$

  where $[\lambda] = \{1, \cdots, \lambda\}$. Then, $\mathsf{KeyGen}_{\mathsf{Univ}}(1^\lambda)$ runs $k \leftarrow \mathsf{KeyGen}_\lambda(1^\lambda)$, and outputs k.
- $\mathsf{StateGen}_{\mathsf{Univ}}(k)$ runs $\psi_k \leftarrow \mathsf{StateGen}_\lambda(1^\lambda, k)$, and outputs ψ_k.
- $\mathsf{Vrfy}_{\mathsf{Univ}}(k, \psi_k)$ runs $\mathsf{Vrfy}_\lambda(1^\lambda, k, \psi_k)$, and outputs its output.

Assume that there exist OWSGs, then there also exists a set of classical Turing machine $\mathcal{M}^* = (x^*, y^*, z^*)$ such that the OWSG scheme $(\mathsf{KeyGen}[x^*], \mathsf{StateGen}[y^*], \mathsf{Vrfy}[z^*])$ satisfies correctness and security. For all sufficiently large $\lambda \in \mathbb{N}$, one of $\{\mathsf{KeyGen}[x], \mathsf{StateGen}[y], \mathsf{Vrfy}[z]\}_{x,y,z\in[\lambda]}$ includes a correct and secure OWSG scheme $(\mathsf{KeyGen}[x^*], \mathsf{StateGen}[y^*], \mathsf{Vrfy}[z^*])$ as long as OWSGs exist. Therefore, $(\mathsf{KeyGen}_\lambda, \mathsf{StateGen}_\lambda, \mathsf{Vrfy}_\lambda)$ satisfies correctness and security for all sufficiently large $\lambda \in \mathbb{N}$ as long as OWSGs exist. Because $(\mathsf{KeyGen}_{\mathsf{Univ}}, \mathsf{StateGen}_{\mathsf{Univ}}, \mathsf{Vrfy}_{\mathsf{Univ}})$ emulates $(\mathsf{KeyGen}_\lambda, \mathsf{StateGen}_\lambda, \mathsf{Vrfy}_\lambda)$, it also satisfies correctness and security

2.5 Universal Plaintext Expansion for Unclonable Encryption

We give another universal construction for one-time unclonable SKE assuming decomposable quantum randomized encoding whose construction is inspired by [WW23]. Although we additionally use a decomposable quantum randomized encoding for this construction, we can expand the plaintext of one-time unclonable SKE. Note that it was an open problem to expand the plaintext of unclonable encryption since a standard transformation via bit-wise encryption does not work as pointed out in [AKL+22].

First, let us recall the decomposable quantum randomized encoding $\Sigma_{\mathsf{RE}} = \mathsf{RE}.(\mathsf{Enc}, \mathsf{Dec})$ given in [BY22]. In their decomposable quantum randomized encoding, $\mathsf{RE}.\mathsf{Enc}$ takes as input a quantum circuit F, λ-length possibly quantum input q and λ-length classical input x, and outputs $\widehat{F(q,x)}$. Let $q[i]$ and $x[i]$ be the i-th qubit and bit of q and x, respectively. Decomposability guarantees that $\widehat{F(q,x)}$ can be separated into the offline encoding part $\widehat{F}_{\mathsf{off}}$ and online encoding parts $\left(\{\mathsf{lab}_i(q[i])\}_{i\in\{1,\cdots,\lambda\}}, \{\mathsf{lab}_{i+\lambda}(x[i])\}_{i\in\{1,\cdots,\lambda\}}\right)$ as follows:

$$\widehat{F(q,x)} \coloneqq \left(\widehat{F}_{\mathsf{off}}, \mathsf{lab}_1(q[1]), \cdots, \mathsf{lab}_\lambda(q[\lambda]), \mathsf{lab}_{\lambda+1}(x[1]), \cdots, \mathsf{lab}_{2\lambda}(x[\lambda])\right),$$

where $\widehat{F}_{\mathsf{off}}$ does not depend on q and x, $\mathsf{lab}_i(q[i])$ depends on only $q[i]$ for $i \in [\lambda]$ and $\mathsf{lab}_{i+\lambda}(x[i])$ depends on only $x[i]$ for $i \in [\lambda]$. $\mathsf{RE}.\mathsf{Dec}$ takes as input $\widehat{F(q,x)}$ and outputs $F(q,x)$. The security roughly guarantees that for any quantum circuits F_1, F_2 with the same size, and any quantum and classical inputs $(\{q_1, x_1\}, \{q_2, x_2\})$ such that $F_1(q_1, x_1) = F_2(q_2, x_2)$, $\widehat{F_1(q_1, x_1)}$ is computationally indistinguishable from $\widehat{F_2(q_2, x_2)}$.

Now, we describe our one-time unclonable SKE $\Sigma_{\mathsf{Univ}} = (\mathsf{KeyGen}_{\mathsf{Univ}}, \mathsf{Enc}_{\mathsf{Univ}}, \mathsf{Dec}_{\mathsf{Univ}})$:

$\mathsf{KeyGen}_{\mathsf{Univ}}(1^\lambda)$: Our key generation algorithm $\mathsf{KeyGen}_{\mathsf{Univ}}(1^\lambda)$ first samples $x \leftarrow \{0,1\}^\lambda$. Then, it samples $R[i] \leftarrow \{0,1\}^{\ell(\lambda)}$ for $i \in [\lambda]$, and outputs $\mathsf{sk} \coloneqq (x, \{R[i]\}_{i\in[\lambda]})$. Here, $\ell(\lambda)$ is the size of online encoding of $\mathsf{RE}.\mathsf{Enc}$.

$\mathsf{Enc}_{\mathsf{Univ}}(\mathsf{sk}, m)$: Our encryption algorithm $\mathsf{Enc}_{\mathsf{Univ}}(\mathsf{sk}, m)$ first generates a quantum circuit $C[m]$ that outputs m for any inputs, where the quantum circuit is padded to an appropriate size, which we will specify later. Then, $\mathsf{Enc}_{\mathsf{Univ}}(\mathsf{sk}, m)$ computes $\widehat{C[m]}_{\mathsf{off}}$, which is the offline encoding of $C[m]$. Next, it computes $\mathsf{lab}_i(0)$ for $i \in [\lambda]$ and $\mathsf{lab}_{\lambda+i}(b)$ for $i \in [\lambda]$ and $b \in \{0,1\}$. Finally, it samples $S[i] \leftarrow \{0,1\}^\lambda$, and computes $\mathsf{Lab.CT}[i, x[i]] = R[i] + \mathsf{lab}_{\lambda+i}(x[i])$ and $\mathsf{Lab.CT}[i, 1-x[i]] = S[i] + \mathsf{lab}_{\lambda+i}(1-x[i])$ for all $i \in [\lambda]$. The ciphertext of $\mathsf{Enc}_{\mathsf{Univ}}(\mathsf{sk}, m)$ is

$$\widehat{C[m]}_{\mathsf{off}}, \{\mathsf{lab}_i(0)\}_{i\in[\lambda]}, \{\mathsf{Lab.CT}[i,b]\}_{i\in[\lambda],b\in\{0,1\}}.$$

$\mathsf{Dec}_{\mathsf{Univ}}(\mathsf{sk}, \mathsf{CT})$: Our decryption algorithm $\mathsf{Dec}_{\mathsf{Univ}}(\mathsf{sk}, \mathsf{CT})$ works as follows. First, let $\mathsf{sk} = (x, \{R[i]\}_{i\in[\lambda]})$ and $\mathsf{CT} = \left(\widehat{C[m]}_{\mathsf{off}}, \{\mathsf{lab}_i(0)\}_{i\in[\lambda]}, \{\mathsf{Lab.CT}[i,b]\}_{i\in[\lambda],b\in\{0,1\}}\right)$. $\mathsf{Dec}_{\mathsf{Univ}}(\mathsf{sk}, \mathsf{CT})$ first computes $\mathsf{lab}_{\lambda+i}(x[i]) = R[i] + \mathsf{Lab.CT}[i, x[i]]$ for all $i \in [\lambda]$, and runs $\mathsf{RE}.\mathsf{Dec}(\widehat{C[m]}_{\mathsf{off}}, \{\mathsf{lab}_i(0)\}_{i\in[\lambda]}, \{\mathsf{lab}_{i+\lambda}(x[i])\}_{i\in[\lambda]})$.

Clearly, our encryption algorithm can encrypt arbitrary-length plaintext. We can see that our construction satisfies correctness. More formally, $\mathsf{Dec}_{\mathsf{Univ}}(\mathsf{sk}, \mathsf{CT}_m)$ outputs m with high probability if $\mathsf{sk} \leftarrow \mathsf{KeyGen}_{\mathsf{Univ}}(1^\lambda)$ and $\mathsf{CT}_m \leftarrow \mathsf{Enc}_{\mathsf{Univ}}(\mathsf{sk}, m)$. From our construction, $\mathsf{Dec}_{\mathsf{Univ}}(\mathsf{sk}, \mathsf{CT}_m)$ outputs the output of $\mathsf{RE}.\mathsf{Dec}(\widehat{C[m]}_{\mathsf{off}}, \{\mathsf{lab}_i(0)\}_{i\in[\lambda]}, \{\mathsf{lab}_{i+\lambda}(x[i])\}_{i\in[\lambda]})$, where $\left(\widehat{C[m]}_{\mathsf{off}}, \{\mathsf{lab}_i(0)\}_{i\in[\lambda]}, \{\mathsf{lab}_{i+\lambda}(x[i])\}_{i\in[\lambda]}\right) \leftarrow \mathsf{RE}.\mathsf{Enc}(C, 0^\lambda, x)$. From the correctness of decomposable quantum randomized encoding, $\mathsf{RE}.\mathsf{Dec}(\widehat{C[m]}_{\mathsf{off}}, \{\mathsf{lab}_i(0)\}_{i\in[\lambda]}, \{\mathsf{lab}_{i+\lambda}(x[i])\}_{i\in[\lambda]})$ outputs $C[m](0^\lambda, x)$, which is equal to m.

Furthermore, our construction Σ_{Univ} satisfies unclonable IND-CPA security as long as the underlying decomposable quantum randomized encoding Σ_{RE} satisfies security and there exists a one-time unclonable SKE for single-bit plaintexts. To see this, we introduce some notations and observations. We write $\Sigma_{\mathsf{unclone}} = \mathsf{Unclone}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ to mean a one-time unclonable SKE for single-bit plaintexts, which we assume to exist. Without loss of generality, we can assume that the secret key sk generated by $\mathsf{Unclone.KeyGen}(1^\lambda)$ is uniformly randomly sampled and $|\mathsf{sk}| = |\mathsf{CT}| = \lambda$ for all security parameters λ. Moreover, we can assume that for a security parameter λ, $\mathsf{Unclone.Dec}(\mathsf{sk}, \mathsf{CT})$ is a quantum algorithm that runs some quantum circuit $\mathsf{Unclone.Dec}_\lambda$ on CT and sk, and outputs its output. We introduce a

quantum circuit $D_\lambda[m_0, m_1]$ that takes as input CT and sk, and runs the quantum circuit $\mathsf{Unclone.Dec}_\lambda$ on CT and sk, obtains b and outputs m_b. The size of $C[m]$ is padded so that its size is equal to $D_\lambda[m_0, m_1]$.

Now, we can see that our construction Σ_{Univ} satisfies one-time unclonable IND-CPA security. In the first step of the proof, we switch the following real ciphertext for message m_b

$$\mathsf{CT}_b = \left(\widehat{C[m_b]}_{\mathsf{off}}, \{\mathsf{lab}_i(0)\}_{i\in[\lambda]}, \{\mathsf{Lab.CT}[i,\beta]\}_{i\in[\lambda],\beta\in\{0,1\}}\right)$$

to the following modified ciphertext

$$\widetilde{\mathsf{CT}_b} = \left(\widehat{D[m_0, m_1]}_{\mathsf{off}}, \{\mathsf{lab}_i(\mathsf{unclone.CT}_b[i])\}_{i\in[\lambda]}, \{\mathsf{Lab.CT}[i,\beta]\}_{i\in[\lambda],\beta\in\{0,1\}}\right),$$

where $\mathsf{unclone.CT}_b \leftarrow \mathsf{Unclone.Enc}(x, b)$ and $\mathsf{unclone.CT}_b[i]$ is the i-th qubit of $\mathsf{unclone.CT}_b$ and $x \leftarrow \{0,1\}^\lambda$. This change does not affect the output of the security experiment because Σ_{RE} satisfies security and we have

$$D[m_0, m_1](\mathsf{unclone.CT}_b, x) = C[m_b](0^\lambda, x) = m_b.$$

In the next step, we can reduce the security of our construction Σ_{Univ} to that of one-time unclonable SKE for single-bit plaintexts $\Sigma_{\mathsf{unclone}}$. This is because the adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ of $\Sigma_{\mathsf{unclone}}$ can simulate the challenger of Σ_{Univ} sicne $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ can simulate $\widetilde{\mathsf{CT}_b}$ by using $\mathsf{unclone.CT}_b$.

3 Preliminaries

3.1 Notations

Here we introduce basic notations we will use in this paper. $x \leftarrow X$ denotes selecting an element x from a finite set X uniformly at random, and $y \leftarrow \mathcal{A}(x)$ denotes assigning to y the output of a quantum or probabilistic or deterministic algorithm $\mathcal{A}$ on an input x. When we explicitly write that $\mathcal{A}$ uses randomness r, we write $y \leftarrow \mathcal{A}(x; r)$. Let $[n] := \{1, \cdots, n\}$. For $x \in \{0,1\}^n$ and $i \in [n]$, x_i and $x[i]$ are the i-th bit value of x. For an n-qubit state ρ and $i \in [n]$, we write ρ_i and $\rho[i]$ to mean a quantum state that traces out all states other than the i-th qubit of ρ. QPT stands for quantum polynomial time. A function $f : \mathbb{N} \to \mathbb{R}$ is a negligible function if, for any constant c, there exists $\lambda_0 \in \mathbb{N}$ such that for any $\lambda > \lambda_0$, $f(\lambda) < 1/\lambda^c$. We write $f(\lambda) \leq \mathsf{negl}(\lambda)$ to denote $f(\lambda)$ being a negligible function.

For simplicity, we often write $|0\rangle$ to mean $|0\cdots 0\rangle$. For any two quantum states ρ_1 and ρ_2, $F(\rho_1, \rho_2)$ is the fidelity between them, and $\mathsf{TD}(\rho_1, \rho_2)$ is the trace distance between them.

For a quantum algorithm $\mathcal{A}$, and quantum states ρ and σ, we say that $\mathcal{A}$ distinguishes ρ from σ with advantage Δ if

$$|\Pr[1 \leftarrow \mathcal{A}(\rho)] - \Pr[1 \leftarrow \mathcal{A}(\sigma)]| = \Delta.$$

We say that ρ is c-computationally indistinguishable (resp. c-statistically indistinguishable) from σ if no QPT algorithms (resp. unbounded algorithms) can distinguish ρ from σ with advantage greater than c.

Quantum Circuits For convenience, we assume that all quantum circuits use gates from the universal gate set $\{I, H, CNOT, T\}$. A unitary quantum circuit is one that consists only of gates from this gate set. A general quantum circuit is a quantum circuit that can additionally have non-unitary gates that (a) introduce new qubits initialized in the zero state, (b) trace them out, or (c) measure them in the computational basis. We say that a general quantum circuit has size s if the total number of gates is at most s.

Definition 3.1 (Uniform Quantum Polynomial Time Algorithm). *We say that an algorithm $\mathcal{A}$ is a uniform quantum polynomial time (QPT) algorithm if $\mathcal{A}$ works as follows: For any pair of classical and quantum input (x, ρ), $\mathcal{A}$ runs some deterministic classical polynomial-time Turing machine $\mathcal{M}$ on $(x, |\rho|)$, and obtains a general quantum circuit $C_{x,|\rho|}$ within $\mathrm{poly}(|x|, |\rho|)$ steps, and outputs the output of $C_{x,|\rho|}(\rho)$.*

We say that the sequence of unitaries $\{U_\lambda\}_{\lambda\in\mathbb{N}}$ is a uniform QPT unitary if U_λ is the output of $\mathcal{M}(1^\lambda)$ for all $\lambda \in \mathbb{N}$, where $\mathcal{M}$ is a classical Turing machine that halts within $\mathrm{poly}(\lambda)$ steps for any input $\lambda \in \mathbb{N}$.

Remark 3.2. We consider many algorithms as uniform QPT algorithms, and thus an algorithm Alg is represented as a classical Turing machine that generates general quantum circuits. If $x \in \{0,1\}^*$ is a classical Turing machine that represents Alg, then we sometimes explicitly write $\mathsf{Alg}[x]$.

Definition 3.3 (Non-Uniform Quantum Polynomial Time Algorithm). *We say that an algorithm $\mathcal{A}$ is a non-uniform quantum polynomial time algorithm if $\mathcal{A}$ works as follows: For any pair of classical and quantum input (x, ρ), $\mathcal{A}$ runs a general quantum circuit C with size $\mathrm{poly}(|x|, |\rho|)$ on (x, ρ) and a quantum advice ψ with size $\mathrm{poly}(|x|, |\rho|)$, and outputs its output.*

Remark 3.4. Throughout this work, we model adversaries as non-uniform QPT algorithms. Note that all results except for Section 8 hold in the uniform adversary setting with appropriate modifications.

Other Notions:

Lemma 3.5 (Gentle Measurement Lemma). *Let ρ be a mixed state, and let E be a measurement operator. Suppose that $\mathrm{Tr}(E\rho) \geq 1 - \epsilon$, where $0 < \epsilon \leq 1$. Then, the post-measurement quantum state $\rho' := \frac{\sqrt{E}\rho\sqrt{E}}{\mathrm{Tr}(E\rho)}$ satisfies:*

$$\|\rho - \rho'\|_1 \leq 2\sqrt{\epsilon}.$$

Theorem 3.6 (Uhlmann's Theorem). *Let $|\psi\rangle_{\mathbf{C},\mathbf{R}}$ and $|\phi\rangle_{\mathbf{C},\mathbf{R}}$ be quantum states over the $\mathbf{C}$ and $\mathbf{R}$ registers. Then, for any unitary $U_{\mathbf{R}}$ acting over $\mathbf{R}$ register, we have*

$$F(\rho, \sigma) = \left| \langle\psi|_{\mathbf{C},\mathbf{R}} \left(I_{\mathbf{C}} \otimes U_{\mathbf{R}}\right) |\phi\rangle_{\mathbf{C},\mathbf{R}} \right|^2,$$

where $\rho = \mathrm{Tr}_{\mathbf{R}}(|\psi\rangle \langle\psi|_{\mathbf{C},\mathbf{R}})$ and $\sigma = \mathrm{Tr}_{\mathbf{R}}(|\phi\rangle \langle\phi|_{\mathbf{C},\mathbf{R}})$.

3.2 Cryptographic Tools

In this section, we introduce cryptographic tools which we will use.

One-Way State Generators. In this work, we consider the mixed-state output version of one-way state generators introduced in [MY22a].

Definition 3.7 (One-way state generators(OWSGs)). *A one-way state generator (OWSG) candidate is a set of algorithms $\Sigma := (\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Vrfy})$ such that:*

$\mathsf{KeyGen}(1^\lambda)$*: It takes a security parameter 1^λ, and outputs a classical string k.*

$\mathsf{StateGen}(1^\lambda, k)$*: It takes a security parameter 1^λ and k, and outputs a quantum state ψ_k.*

$\mathsf{Vrfy}(1^\lambda, k, \psi_k)$*: It takes a security parameter 1^λ, k and ψ_k, and outputs $\top$ or $\bot$.*

We say that a candidate Σ is a OWSG scheme if Σ satisfies the following efficiency, correctness, and security properties.

Efficiency. *The algorithms $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Vrfy})$ are uniform QPT algorithms.*

Correctness. *We have*

$$\Pr\left[\top \leftarrow \mathsf{Vrfy}(1^\lambda, k, \psi_k) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \psi_k \leftarrow \mathsf{StateGen}(1^\lambda, \psi_k)\right] \geq 1 - \mathsf{negl}(\lambda).$$

Security. *For any non-uniform QPT algorithm $\mathcal{A}$ and any polynomial $t(\cdot)$,*

$$\Pr\left[\top \leftarrow \mathsf{Vrfy}(1^\lambda, k^*, \psi_k) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \psi_k \leftarrow \mathsf{StateGen}(1^\lambda, k), k^* \leftarrow \mathcal{A}(\psi_k^{\otimes t(\lambda)})\right] \leq \mathsf{negl}(\lambda).$$

Remark 3.8. If a OWSG scheme $(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Vrfy})$ satisfies

$$\Pr\left[\top \leftarrow \mathsf{Vrfy}(1^\lambda, k, \psi_k) : k \leftarrow \mathsf{KeyGen}(1^\lambda), \psi_k \leftarrow \mathsf{StateGen}(1^\lambda, \psi_k)\right] = 1$$

for all security parameters $\lambda \in \mathbb{N}$, then we say that the OWSG scheme satisfies perfect correctness.

Public-Key Quantum Money Mini-Scheme. In this work, we consider public-key quantum money mini-scheme.

Definition 3.9 (Public-Key Quantum Money Mini-Scheme [AC12]). *A public-key quantum money mini-scheme candidate is a set of algorithms $\Sigma \coloneqq (\mathsf{Mint}, \mathsf{Vrfy})$ such that:*

$\mathsf{Mint}(1^\lambda)$*: It takes a security parameter 1^λ, and outputs a serial number s and a quantum state ρ_s.*

$\mathsf{Vrfy}(1^\lambda, s, \rho_s)$*: It takes a security parameter 1^λ, s, and ρ_s, and outputs $\top$ or $\bot$.*

We say that a candidate Σ is a public-key quantum money mini-scheme if it satisfies the following efficiency, correctness, and security properties.

Efficiency. *The algorithms $(\mathsf{Mint}, \mathsf{Vrfy})$ are uniform QPT algorithms.*

Correctness. *We have*

$$\Pr\left[\top \leftarrow \mathsf{Vrfy}(1^\lambda, s, \rho_s) : (s, \rho_s) \leftarrow \mathsf{Mint}(1^\lambda)\right] \geq 1 - \mathsf{negl}(\lambda).$$

Security. *For any non-uniform QPT algorithm $\mathcal{A}$ and any polynomial $t(\cdot)$,*

$$\Pr\left[\top \leftarrow \mathsf{Vrfy}(1^\lambda, s, \sigma[1]) \wedge \top \leftarrow \mathsf{Vrfy}(1^\lambda, s, \sigma[2]) : (s, \rho_s) \leftarrow \mathsf{Mint}(1^\lambda), \sigma \leftarrow \mathcal{A}(\rho_s)\right] \leq \mathsf{negl}(\lambda),$$

where σ is a quantum state on 2 registers, R_1, R_2 each of which is of $|\rho_s|$ qubits and where $\sigma[1] \coloneqq \mathrm{Tr}_{R[2]}(\sigma)$ and $\sigma \coloneqq \mathrm{Tr}_{R[1]}(\sigma)$.

Remark 3.10. If a public-key quantum money mini-scheme $(\mathsf{Mint}, \mathsf{Vrfy})$ satisfies

$$\Pr\left[\top \leftarrow \mathsf{Vrfy}(1^\lambda, s, \rho_k) : (s, \rho_s) \leftarrow \mathsf{Mint}(1^\lambda)\right] = 1$$

for all security parameters $\lambda \in \mathbb{N}$, then we say that the public-key quantum money mini-scheme satisfies perfect correctness.

We note that a public-key quantum money mini-scheme can be upgraded into a full-fledged public-key quantum money additionally using standard digital signatures [AC12].

Canonical Quantum Bit Commitment.

Definition 3.11 (Canonical Quantum Bit Commitment [Yan22]). *A candidate for canonical quantum bit commitment is a set of uniform QPT unitaries $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ acting on the register $\mathbf{C}$ and $\mathbf{R}$. We consider the following two properties.*

Hiding. *We say that a candidate for canonical quantum bit commitment $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies c-statistical hiding (resp. c-computational hiding) if $\mathrm{Tr}_{\mathbf{R}}(Q_0(\lambda)\left|0\right\rangle_{\mathbf{CR}})$ is c-statistically indistinguishable (resp. c-computationally indistinguishable) from $\mathrm{Tr}_{\mathbf{R}}(Q_1(\lambda)\left|0\right\rangle_{\mathbf{CR}})$ for all sufficiently large $\lambda \in \mathbb{N}$.*

If a candidate for canonical quantum bit commitment satisfies $\mathsf{negl}(\lambda)$-statistical hiding (resp. $\mathsf{negl}(\lambda)$-computational hiding), then we say that the candidate satisfies statistical hiding (resp. computational hiding).

Binding. *We say that a candidate for canonical quantum bit commitment $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies c-statistical binding (resp. c-computational binding) if for all sufficiently large security parameters $\lambda \in \mathbb{N}$, any unbounded-time (resp. QPT) unitary U over $\mathbf{R}$ and an additional register $\mathbf{Z}$ and any polynomial-size $|\tau\rangle$, it holds that*

$$\left\|(\langle 0|\, Q_1^\dagger(\lambda))_{\mathbf{C},\mathbf{R}}(I_{\mathbf{C}} \otimes U_{\mathbf{R},\mathbf{Z}})((Q_0(\lambda)\left|0\right\rangle_{\mathbf{C},\mathbf{R}})\left|\tau\right\rangle_{\mathbf{Z}})\right\| \leq c.$$

If a candidate for canonical quantum bit commitment satisfies $\mathsf{negl}(\lambda)$-statistical binding (resp. $\mathsf{negl}(\lambda)$-computational binding), then we say that the candidate satisfies statistical binding (resp. computational binding).

It was shown that we can convert the flavor of quantum bit commitment as follows.

Lemma 3.12 (Converting Flavors:[HMY23]). *Let $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ be a candidate of canonical quantum bit commitment. Let $\{\widetilde{Q_0}(\lambda), \widetilde{Q_1}(\lambda)\}_{\lambda\in\mathbb{N}}$ be a candidate of canonical quantum bit commitment described as follows:*

- *The role of commitment and reveal registers are swapped from $(Q_0(\lambda), Q_1(\lambda))$ and the commitment register is augmented by an additional one-qubit register which we denote $\mathbf{D}$. In other words, if $\mathbf{C}$ and $\mathbf{R}$ are the commitment and reveal registers of $(Q_0(\lambda), Q_1(\lambda))$, then the commitment and reveal registers of $(\widetilde{Q_0}(\lambda), \widetilde{Q_1}(\lambda))$ are defined as $\widetilde{\mathbf{C}} := (\mathbf{R}, \mathbf{D})$ and $\widetilde{\mathbf{R}} := \mathbf{C}$, where $\mathbf{D}$ is an additional one-qubit register.*
- *For $b \in \{0,1\}$, the unitary $\widetilde{Q_b}(\lambda)$ is defined as follows:*

$$\widetilde{Q_b}(\lambda) := (Q_0(\lambda) \otimes |0\rangle\langle 0|_{\mathbf{D}} + Q_1(\lambda) \otimes |1\rangle\langle 1|_{\mathbf{D}}) \left(I_{\mathbf{RC}} \otimes Z_{\mathbf{D}}^b H_{\mathbf{D}}\right).$$

The following holds for $X, Y \in \{$statistical, computational$\}$.

1. *If $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies c-X hiding, then $\{\widetilde{Q_0}(\lambda), \widetilde{Q_1}(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies $\sqrt{c}$-X binding.*
2. *If $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies $\mathsf{negl}(\lambda)$-Y binding, then $\{\widetilde{Q_0}(\lambda), \widetilde{Q_1}(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies $\mathsf{negl}(\lambda)$-Y hiding.*

Remark 3.13. The previous work [HMY23] considered the case where the original commitment $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies $\mathsf{negl}(\lambda)$-X hiding. However, for our purpose, we need to analyze the case where the original commitment $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies c-X hiding for some constant c instead of $\mathsf{negl}(\lambda)$-X hiding. For the reader's convenience, we describe the proof in Appendix C. Remark that the proof is the same as the previous work.

Unclonable Encryption. In this work, we consider unclonable encryption with unclonable IND-CPA security.

Definition 3.14 (Unclonable Secret-Key Encryption [BL20]). *A candidate for unclonable secret-key encryption for $n(\lambda)$-bit plaintexts is a set of algorithms $\Sigma := (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ such that:*

$\mathsf{KeyGen}(1^\lambda)$*: It takes as input a security parameter 1^λ, and outputs a classical secret-key* sk.

$\mathsf{Enc}(1^\lambda, \mathsf{sk}, m)$*: It takes as input a security parameter 1^λ,* sk *and $m \in \{0,1\}^{n(\lambda)}$, and outputs a quantum ciphertext* CT.

$\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT})$*: It takes as input a security parameter 1^λ,* sk *and* CT*, and outputs m.*

We say that a candidate Σ is an unclonable SKE scheme if it satisfies the following efficiency, correctness, IND-CPA security, and unclonable IND-CPA security.

Efficiency. *The algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *are uniform QPT algorithms.*

Correctness. *We have*

$$\Pr\left[m \leftarrow \mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT}) : \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda), \mathsf{CT} \leftarrow \mathsf{Enc}(1^\lambda, \mathsf{sk}, m)\right] \geq 1 - \mathsf{negl}(\lambda).$$

Unclonable IND-CPA Security. *We require that Σ satisfies standard IND-CPA security. In addition to the standard IND-CPA security, we require that Σ satisfies the unclonable IND-CPA security defined below. Given an unclonable encryption Σ, we consider the unclonable IND-CPA security experiment $\mathsf{Exp}^{\mathsf{unclone}}_{\Sigma,(\mathcal{A},\mathcal{B},\mathcal{C})}(\lambda)$ against $(\mathcal{A}, \mathcal{B}, \mathcal{C})$.*

1. *The challenger runs* $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2. *$\mathcal{A}$ can query* $\mathsf{Enc}(1^\lambda, \mathsf{sk}, \cdot)$ *polynomially many times.*
3. *$\mathcal{A}$ sends (m_0, m_1) to the challenger.*

4. *The challenger samples* $b \leftarrow \{0,1\}$*, runs* $\mathsf{CT}_b \leftarrow \mathsf{Enc}(1^\lambda, \mathsf{sk}, m_b)$*, and sends* CT_b *to* $\mathcal{A}$*.*

5. $\mathcal{A}$ *produces* $\rho_{\mathcal{B},\mathcal{C}}$ *and sends the corresponding registers to* $\mathcal{B}$ *and* $\mathcal{C}$*.*

6. $\mathcal{B}$ *and* $\mathcal{C}$ *receive* sk *and output* $b_{\mathcal{B}}$ *and* $b_{\mathcal{C}}$*.*

7. *The experiment outputs* 1 *indicating win if* $b_{\mathcal{B}} = b_{\mathcal{C}} = b$*, and otherwise* 0*.*

We say that Σ *is unclonable IND-CPA secure if for all sufficiently large security parameters* $\lambda \in \mathbb{N}$*, for all non-uniform QPT adversaries* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$*,*

$$\Pr\left[\mathsf{Exp}^{\mathsf{unclone}}_{\Sigma,(\mathcal{A},\mathcal{B},\mathcal{C})}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Remark 3.15. We also consider one-time unclonable secret-key encryption. It is the same as unclonable secret-key encryption except that it satisfies one-time IND-CPA security and one-time unclonable IND-CPA security instead of IND-CPA security and unclonable IND-CPA security. The one-time unclonable IND-CPA security is the same as unclonable IND-CPA security except that the adversary is not allowed to query the encryption oracle.

Remark 3.16. If an unclonable SKE scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies

$$\Pr\left[m \leftarrow \mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT}) : \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda), \mathsf{CT} \leftarrow \mathsf{Enc}(1^\lambda, \mathsf{sk}), m\right] = 1$$

for all security parameters $\lambda \in \mathbb{N}$ and all $m \in \mathcal{M}_\lambda$, then we say that the unclonable SKE scheme satisfies perfect correctness.

We also consider unclonable PKE. For clarity, we describe unclonable PKE with unclonable IND-CPA security.

Definition 3.17 (Unclonable Public-Key Encryption[AK21]). *A candidate for unclonable public-key encryption for* $n(\lambda)$*-bit plaintexts is a set of algorithms* $\Sigma := (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that:*

$\mathsf{KeyGen}(1^\lambda)$*: It takes as input a security parameter* 1^λ*, and outputs a classical secret-key* sk *and a classical public-key* pk*.*

$\mathsf{Enc}(1^\lambda, \mathsf{pk}, m)$*: It takes as input a security parameter* 1^λ*,* pk *and* $m \in \{0,1\}^{n(\lambda)}$*, and outputs a quantum ciphertext* CT*.*

$\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT})$*: It takes as input a security parameter* 1^λ*,* sk *and* CT*, and outputs* m*.*

We say that a candidate Σ *satisfies efficiency, correctness, IND-CPA security, and unclonable IND-CPA security, respectively if* Σ *satisfies the following efficiency, correctness, IND-CPA security, and unclonable IND-CPA security property, respectively.*

Efficiency. *The algorithms* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *are uniform QPT algorithms.*

Correctness. *We have*

$$\Pr\left[m \leftarrow \mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT}) : (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda), \mathsf{CT} \leftarrow \mathsf{Enc}(1^\lambda, \mathsf{pk}, m)\right] \geq 1 - \mathsf{negl}(\lambda).$$

Unclonable IND-CPA Security. *We require that* Σ *satisfies standard IND-CPA security. In addition to the standard IND-CPA security, we require that* Σ *satisfies the unclonable IND-CPA security defined below. Given an unclonable encryption* Σ*, we consider the unclonable IND-CPA security experiment* $\mathsf{Exp}^{\mathsf{unclone}}_{\Sigma,(\mathcal{A},\mathcal{B},\mathcal{C})}(\lambda)$ *against* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$*.*

1. *The challenger runs* $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$*, and sends* pk *to* $\mathcal{A}$*.*

2. $\mathcal{A}$ *sends* (m_0, m_1) *to the challenger.*

3. *The challenger samples* $b \leftarrow \{0,1\}$*, runs* $\mathsf{CT}_b \leftarrow \mathsf{Enc}(1^\lambda, \mathsf{pk}, m_b)$*, and sends* CT_b *to* $\mathcal{A}$*.*

4. *$\mathcal{A}$ produces $\rho_{\mathcal{B},\mathcal{C}}$ and sends the corresponding registers to $\mathcal{B}$ and $\mathcal{C}$.*

5. *$\mathcal{B}$ and $\mathcal{C}$ receive sk and output $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$.*

6. *The experiment outputs 1 indicating win if $b_{\mathcal{B}} = b_{\mathcal{C}} = b$, and otherwise 0.*

We say that Σ is unclonable IND-CPA secure if for all sufficiently large security parameters $\lambda \in \mathbb{N}$, for all non-uniform QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$,

$$\Pr\left[\mathsf{Exp}^{\mathsf{unclone}}_{\Sigma,(\mathcal{A},\mathcal{B},\mathcal{C})}(\lambda) = 1\right] \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

Remark 3.18. We say that (one-time) unclonable SKE (resp. PKE) Σ is unclonable SKE (resp. SKE) for single-bit plaintexts if a plaintext space $\mathcal{M}_\lambda$ is $\mathcal{M}_\lambda := \{0,1\}$ for all security parameters $\lambda \in \mathbb{N}$. Note that we cannot expand the plaintext space by bit-wise encryption.

Decomposable Quantum Randomized Encoding.

Definition 3.19 (Decomposable Quantum Randomized Encoding(DQRE) [BY22]). *A DQRE scheme is a tuple of algorithms $(\mathsf{Enc}, \mathsf{Dec})$ such that:*

$\mathsf{Enc}(1^\lambda, F, x)$*: It takes 1^λ with $\lambda \in \mathbb{N}$, a general quantum circuit F and a possibly quantum input x as inputs, and outputs $\widehat{F(x)}$.*

$\mathsf{Dec}(1^\lambda, \widehat{F(x)})$*: It takes as input 1^λ, and $\widehat{F(x)}$, and outputs $F(x)$.*

We require the following four properties:

Efficiency. $(\mathsf{Enc}, \mathsf{Dec})$ *are uniform QPT algorithms.*

Correctness. *For all quantum states (x, q) and randomness r, it holds that $(F(x), q) = (\mathsf{Dec}(1^\lambda, \widehat{F}(x; r)), q)$, where $\widehat{F}(x; r)$ is an output of $\mathsf{Enc}(1^\lambda, F, x; r)$.*

Security. *There exists a uniform QPT algorithm Sim such that for all quantum states (x, q) and non-uniform QPT adversary $\mathcal{A}$, there exists some negligible function negl that satisfies,*

$$\left|\Pr\left[1 \leftarrow \mathcal{A}(\widehat{F}(x; r), q)\right] - \Pr\left[1 \leftarrow \mathcal{A}(\mathsf{Sim}(1^\lambda, |F|, F(x)), q)\right]\right| \leq \mathsf{negl}(\lambda),$$

where the state on the left-hand side is averaged over r and $|F|$ is the size of the general quantum circuit F.

Remark 3.20. In the security of the original paper [BY22], the simulator Sim takes the topology of F as input. Without loss of generality, we can replace the topology of F with the size of F because we can hide the topology of F by using a universal quantum circuit.

Decomposability. *There exists a quantum state e (called the resource state of the encoding), and operation $\widehat{F}_{\mathsf{off}}$ (called the offline part of the encoding) and a collection of input encoding operations $\widehat{F_1}, \cdots, \widehat{F_n}$ such that for all inputs $x = (x_1, \cdots, x_n)$,*

$$\widehat{F}(x; r) = \left(\widehat{F}_{\mathsf{off}}, \widehat{F}_1, \widehat{F_2}, \cdots, \widehat{F_n}\right)(x, r, e)$$

where the functions $\widehat{F}_{\mathsf{off}}, \widehat{F_1}, \cdots, \widehat{F_n}$ act on disjoint subsets of qubits from e, x (but can depend on all bits of r), each $\widehat{F}_i$ acts on a single qubit x_i and $\widehat{F}_{\mathsf{off}}$ does not act on any of the qubits of x.

Classical Labels. *If x_i is a classical bit, then $\widehat{F_i}(x_i, r)$ is a classical string as well.*

Theorem 3.21 ([BY22]). *Decomposable quantum randomized encoding exists if OWFs exist.*

Proposition 3.22. *Let $\Sigma := (\mathsf{Enc}, \mathsf{Dec})$ be a decomposable quantum randomized encoding. Then, for any quantum circuits F_0, F_1 with the same size, for any possibly quantum input x_0 and x_1 such that $F_0(x_0) = F_1(x_1)$, $\widehat{F_0}(x_0; r_0)$ is computationally indistinguishable from $\widehat{F_1}(x_1; r_1)$, where both quantum states are averaged over the randomness r_0 and r_1.*

This can be shown by a standard hybrid argument, and thus we omit the proof.

4 Robust OWSGs Combiner

Definition 4.1 (Robust OWSGs Combiner). *A robust OWSGs combiner is a deterministic classical polynomial-time Turing machine $\mathcal{M}$ with the following properties:*

- *$\mathcal{M}$ takes as input 1^n with $n \in \mathbb{N}$ and n-candidates OWSGs $\{\Sigma_i := (\mathsf{KeyGen}_i, \mathsf{StateGen}_i, \mathsf{Vrfy}_i)\}_{i\in[n]}$ promised that all candidates satisfy efficiency, and outputs a single set of algorithms $\Sigma := (\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Vrfy})$.*
- *If all of $\{\Sigma_i\}_{i\in[n]}$ satisfy efficiency and at least one of $\{\Sigma_i\}_{i\in[n]}$ satisfies both correctness and security, then Σ is an OWSG scheme that satisfies efficiency, correctness, and security.*

Remark 4.2. In the previous work [HKN+05], they define robust combiners in a similar way where n is treated as an arbitrary function in the security parameter. However, it is unclear what is meant by the definition where n is a super-constant. This is because the security parameter for the scheme Σ obtained by a robust combiner is an arbitrary non-negative integer after combining n candidates $\{\Sigma_i\}_{i\in[n]}$. Therefore, in the definition above, we consider n as a constant in λ. On the other hand, Definition 4.1 is not sufficient to construct universal construction since n is constant in λ. Therefore, we also introduce another definition (Definition 4.10) of a robust combiner, where n can be dependent on λ. Although our construction actually satisfies Definition 4.10, here we consider Definition 4.1 for simplicity.

Theorem 4.3. *A robust OWSGs combiner exists.*

For proving Theorem 4.3, we introduce the following Lemma 4.4.

Lemma 4.4. *Let $\Sigma = (\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Vrfy})$ be a candidate of OWSG. From Σ, we can construct a OWSG scheme Σ^* with the following properties:*

1. *If Σ is uniform QPT algorithm, Σ^* is uniform QPT algorithm.*
2. *Σ^* satisfies perfect correctness.*
3. *If Σ is a uniform QPT algorithm and satisfies correctness and security, then Σ^* satisfies security.*

Proof of Lemma 4.4. Without loss of generality, $\mathsf{Vrfy}(1^\lambda, k, \psi)$ can be considered as the algorithm working in the following way:

For input $(1^\lambda, k, \psi)$, run a classical Turing machine $\mathcal{M}$ on $(1^\lambda, k, |\psi|)$, obtain $U_{\mathsf{Vrfy},k}$, append auxiliary state $|0\cdots0\rangle\langle0\cdots0|$ to ψ, apply a unitary $U_{\mathsf{Vrfy},k}$ on $\psi \otimes |0\cdots0\rangle\langle0\cdots0|$, and measure the first qubit of $U_{\mathsf{Vrfy},k}(\psi \otimes |0\cdots0\rangle\langle0\cdots0|)U^\dagger_{\mathsf{Vrfy},k}$ with the computational basis and output $\top$ if the measurement result is 1 and $\bot$ otherwise.

We describe the $\Sigma^* := (\mathsf{KeyGen}^*, \mathsf{StateGen}^*, \mathsf{Vrfy}^*)$.

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

- Run $k \leftarrow \mathsf{KeyGen}(1^\lambda)$.
- Output $k^* := k$.

$\mathsf{StateGen}^*(1^\lambda, k^*)$**:**

- Parse $k^* = k$.
- Run $\psi_k \leftarrow \mathsf{StateGen}(1^\lambda, k)$.
- Run $U_{\mathsf{Vrfy},k}$ on $\psi_k \otimes |0\cdots0\rangle\langle 0\cdots0|$, and measures the first qubit of $U_{\mathsf{Vrfy},k}\left(\psi_k \otimes |0\cdots0\rangle\langle 0\cdots0|\right)U_{\mathsf{Vrfy},k}^\dagger$, in the computational basis, and obtains the measurement result b and post-measurement quantum state $\rho_{b,k}$.
  - If the measurement result is 1, then output $\psi_k^* \coloneqq U_{\mathsf{Vrfy},k}^\dagger(|1\rangle\langle 1| \otimes \rho_{1,k})U_{\mathsf{Vrfy},k} \otimes |1\rangle\langle 1|$.
  - If the measurement result is 0, then output $\psi_k^* \coloneqq U_{\mathsf{Vrfy},k}^\dagger(|0\rangle\langle 0| \otimes \rho_{0,k})U_{\mathsf{Vrfy},k} \otimes |0\rangle\langle 0|$.

$\mathsf{Vrfy}^*(1^\lambda, k^*, \psi^*)$**:**

- Parse $k^* = k$ and $\psi^* \coloneqq \rho \otimes |b\rangle\langle b|$.
- Measure the last bit of ψ^* in the computational basis.
  - If 1 is obtained, then measure the first qubit of $U_{\mathsf{Vrfy},k}\rho U_{\mathsf{Vrfy},k}^\dagger$ in the computational basis, and output $\top$ if the measurement outcome is 1 and $\bot$ otherwise.
  - If 0 is obtained, then output $\top$.

The first item and the second item straightforwardly follow, and thus we skip the proof.

Proof of the third item. Assume that Σ^* is not secure for contradiction. More formally, assume that there exists a QPT adversary $\mathcal{A}$ such that the following probability is non-negligible

$$\Pr\left[\top \leftarrow \mathsf{Vrfy}^*(1^\lambda, k', \psi_k^*) : \begin{array}{l} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \psi_k^* \leftarrow \mathsf{StateGen}^*(1^\lambda, k) \\ k' \leftarrow \mathcal{A}(\psi_k^{*\otimes t(\lambda)}) \end{array}\right].$$

Then, construct $\mathcal{B}$ that breaks the security of Σ as follows.

1. $\mathcal{B}$ receives $\psi_k^{\otimes t(\lambda)}$ from $\mathcal{C}$ which is the challenger of Σ.
2. $\mathcal{B}$ sends $\left(\psi_k \otimes |0\cdots0\rangle\langle 0\cdots0| \otimes |1\rangle\langle 1|\right)^{\otimes t}$ to $\mathcal{A}$.
3. $\mathcal{B}$ receives k' from $\mathcal{A}$.
4. $\mathcal{B}$ sends k' to $\mathcal{C}$.

From the construction of $\mathcal{B}$, $\mathcal{B}$ simulates the security experiment of Σ^* except that it uses $\psi_k \otimes |0\cdots0\rangle\langle 0\cdots0| \otimes |1\rangle\langle 1|$ instead of ψ_k^*. Because we assume that Σ satisfies correctness, we have

$$U_{\mathsf{Vrfy},k}(\psi_k \otimes |0\cdots0\rangle\langle 0\cdots0|)U_{\mathsf{Vrfy},k}^\dagger = \mathsf{negl}(\lambda)\,|0\rangle\langle 0| \otimes \rho_{0,k} + (1 - \mathsf{negl}(\lambda))\,|1\rangle\langle 1| \otimes \rho_{1,k}$$

where $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, $\psi_k \leftarrow \mathsf{StateGen}(1^\lambda, k)$, and $\rho_{0,\mathsf{sk}}$ and $\rho_{1,\mathsf{sk}}$ are some appropriate quantum state.

From the gentle measurement lemma (Lemma 3.5), we have

$$\left\| U_{\mathsf{Vrfy},k}(\psi_k \otimes |0\cdots0\rangle\langle 0\cdots0|)U_{\mathsf{Vrfy},k}^\dagger - |1\rangle\langle 1| \otimes \rho_{1,k} \right\|_1 \leq \mathsf{negl}(\lambda).$$

In particular, this implies that

$$\|\psi_k \otimes |0\cdots0\rangle\langle 0\cdots0| \otimes |1\rangle\langle 1| - \psi_k^*\|_1 \leq \mathsf{negl}(\lambda).$$

Therefore, we have

$$\begin{aligned}
&\Pr\left[\top \leftarrow \mathsf{Vrfy}(1^\lambda, k', \psi_k) : \begin{array}{l} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \psi_k \leftarrow \mathsf{StateGen}(k) \\ k' \leftarrow \mathcal{B}\left(\psi_k^{\otimes t(\lambda)}\right) \end{array}\right] \\
&= \Pr\left[\top \leftarrow \mathsf{Vrfy}^*(1^\lambda, k', \psi_k \otimes |0\cdots 0\rangle\langle 0\cdots 0| \otimes |1\rangle\langle 1|) : \begin{array}{l} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \psi_k \leftarrow \mathsf{StateGen}(k) \\ k' \leftarrow \mathcal{A}\left((\psi_k \otimes |0\cdots 0\rangle\langle 0\cdots 0| \otimes |1\rangle\langle 1|)^{\otimes t(\lambda)}\right) \end{array}\right] \\
&\geq \Pr\left[\top \leftarrow \mathsf{Vrfy}^*(1^\lambda, k', \psi_k^*) : \begin{array}{l} k \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \psi_k^* \leftarrow \mathsf{StateGen}^*(k) \\ k' \leftarrow \mathcal{A}(\psi_k^{*\otimes t(\lambda)}) \end{array}\right] - \mathsf{negl}(\lambda) \\
&\geq 1/\lambda^c - \mathsf{negl}(\lambda),
\end{aligned}$$

where in the first equation we have used

$$\Pr\left[\top \leftarrow \mathsf{Vrfy}(1^\lambda, k', \psi)\right] = \Pr\left[\top \leftarrow \mathsf{Vrfy}^*(1^\lambda, k', \psi \otimes |0\cdots 0\rangle\langle 0\cdots 0| \otimes |1\rangle\langle 1|)\right]$$

for any $\lambda \in \mathbb{N}$, k^*, and ψ, and in the second inequality, we have used that $\|\psi_k \otimes |0\cdots 0\rangle\langle 0\cdots 0| \otimes |1\rangle\langle 1| - \psi_k^*\|_1 \leq \mathsf{negl}(\lambda)$. This contradicts that Σ satisfies security, and thus Σ^* satisfies security.

□

Proof of Theorem 4.3. Below, we consider a fixed constant n. Let us introduce some notations.

Notations:

- Let $\Sigma_i := (\mathsf{KeyGen}_i, \mathsf{StateGen}_i, \mathsf{Vrfy}_i)$ be a candidate of OWSG for $i \in [n]$.
- For a candidate of OWSG Σ_i, let $\Sigma_i^* := (\mathsf{KeyGen}_i^*, \mathsf{StateGen}_i^*, \mathsf{Vrfy}_i^*)$ be a candidate of OWSG derived from Lemma 4.4 with the following properties:
  - If Σ_i satisfies efficiency, then Σ_i^* satisfies efficiency.
  - Σ_i^* satisfies perfect correctness.
  - If Σ_i satisfies efficiency, correctness and security, then Σ_i^* satisfies security.

Construction of Robust OWSG Combiner: A robust combiner $\mathcal{M}$ is a classical Turing machine that takes as input 1^n and $\{\Sigma_i\}_{i\in[n]}$, and outputs $\Sigma = (\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Vrfy})$ working in the following way.

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

- For all $i \in [n]$, run $k_i^* \leftarrow \mathsf{KeyGen}_i^*(1^\lambda)$.
- Output $k := \{k_i^*\}_{i\in[n]}$.

$\mathsf{StateGen}(1^\lambda, k)$**:**

- Parse $k = k_1^*||\cdots||k_n^*$.
- For all $i \in [n]$, run $\psi_{k_i^*} \leftarrow \mathsf{StateGen}_i^*(k_i^*)$.
- Output $\psi_k := \bigotimes_{i\in[n]} \psi_{k_i^*}$.

$\mathsf{Vrfy}(1^\lambda, k, \psi_k)$**:**

- Parse $k = k_1||\cdots||k_n$ and $\psi_k = \bigotimes_{i\in[n]} \psi_{k_i}$.
- For all $i \in [n]$, run $\mathsf{Vrfy}_i^*(k_i, \psi_{k_i})$. If $\top \leftarrow \mathsf{Vrfy}_i^*(k_i^*, \psi_{k_i^*})$ for all $i \in [n]$, output $\top$. Otherwise, output $\bot$.

Theorem 4.3 follows from the following Lemmata 4.5 to 4.7.

Lemma 4.5. *If all of $\{\Sigma_i\}_{i\in[n]}$ satisfies efficiency, then Σ satisfies efficiency.*

Lemma 4.6. *Σ satisfies perfect correctness.*

Lemma 4.7. *If all of $\{\Sigma_i\}_{i\in[n]}$ satisfies efficiency and at least one of $\{\Sigma_i\}_{i\in[n]}$ satisfies correctness and security, then Σ satisfies security.*

Lemma 4.5 trivially follows. Lemma 4.6 follows because Σ_i^* satisfies correctness for all $i \in [n]$. The proof of Lemma 4.7 is a standard hybrid argument, and thus we skip the proof.

□

4.1 Universal Construction

Definition 4.8. *We say that a set of uniform QPT algorithms $\Sigma_{\mathsf{Univ}} = (\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Vrfy})$ is a universal construction of OWSG if Σ_{Univ} is an OWSG scheme as long as there exists an OWSG.*

Theorem 4.9. *There exists a universal construction of OWSG.*

For showing Theorem 4.9, the robust OWSGs combiner of Definition 4.1 is not adequate to construct universal construction for OWSGs. Therefore, we reintroduce a definition of robust OWSGs combiner, which we call robust OWSGs combiner for universal construction.

Definition 4.10 (Robust OWSGs Combiner for universal construction). *A $(1, n)$-robust OWSGs combiner for universal construction $\mathsf{Comb}.\Sigma$ consists of three algorithms $(\mathsf{Comb.KeyGen}, \mathsf{Comb.StateGen}, \mathsf{Comb.Vrfy})$, where n is some polynomial. A $(1, n)$-robust OWSG combiner $(\mathsf{Comb.KeyGen}, \mathsf{Comb.StateGen}, \mathsf{Comb.Vrfy})$ has the following syntax:*

$\mathsf{Comb.KeyGen}(1^\lambda, \{\Sigma_i\}_{i\in[n(\lambda)]})$*: It takes as input a security parameter λ and $n(\lambda)$ candidates of OWSGs $\{\Sigma_i\}_{i\in[n(\lambda)]}$ and outputs a classical key k.*

$\mathsf{Comb.StateGen}(1^\lambda, k, \{\Sigma_i\}_{i\in[n(\lambda)]})$*: It takes as input a security parameter 1^λ, k and $\{\Sigma_i\}_{i\in[n(\lambda)]}$, and outputs a quantum state ψ_k.*

$\mathsf{Comb.Vrfy}(1^\lambda, k, \psi_k, \{\Sigma_i\}_{i\in[n(\lambda)]})$*: It takes as input a security parameter 1^λ, k, ψ_k, and $\{\Sigma_i\}_{i\in[n(\lambda)]}$, and outputs $\top$ or $\bot$.*

Efficiency. *The algorithms $(\mathsf{Comb.KeyGen}, \mathsf{Comb.StateGen}, \mathsf{Comb.Vrfy})$ are uniform QPT algorithms.*

Correctness. *For all n candidates $\{\Sigma_i\}_{i\in[n(\lambda)]}$,*

$$\Pr\left[\top \leftarrow \mathsf{Comb.Vrfy}(1^\lambda, k, \psi_k, \{\Sigma_i\}_{i\in[n(\lambda)]}) \;\middle|\; \begin{array}{l} k \leftarrow \mathsf{Comb.KeyGen}(1^\lambda, \{\Sigma_i\}_{i\in[n(\lambda)]}) \\ \psi_k \leftarrow \mathsf{Comb.StateGen}(1^\lambda, k, \{\Sigma_i\}_{i\in[n(\lambda)]}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Security. *Let $\{\Sigma_i\}_{i\in\mathbb{N}}$ be a sequence of candidates of OWSGs promised that Σ_i satisfies efficiency for all $i \in \mathbb{N}$. If there exists $i^* \in \mathbb{N}$ such that Σ_{i^*} satisfies correctness and security and $i^* < n(\lambda)$ for all sufficiently large security parameters $\lambda \in \mathbb{N}$, then for all non-uniform QPT adversaries $\mathcal{A}$ and all polynomials t, we have*

$$\Pr\left[\top \leftarrow \mathsf{Comb.Vrfy}(1^\lambda, k^*, \psi_k, \{\Sigma_i\}_{i\in[n(\lambda)]}) \;\middle|\; \begin{array}{l} k \leftarrow \mathsf{Comb.KeyGen}(1^\lambda, \{\Sigma_i\}_{i\in[n(\lambda)]}) \\ \psi_k \leftarrow \mathsf{Comb.StateGen}(1^\lambda, k, \{\Sigma_i\}_{i\in[n(\lambda)]}) \\ k^* \leftarrow \mathcal{A}(\psi_k^{\otimes t(\lambda)}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Theorem 4.11. *There exists a $(1, n)$-robust OWSG combiner for universal construction for all polynomial n.*

We can show Theorem 4.11 in the same way as Theorem 4.3, and thus we skip the proof. For proving Theorem 4.9, let us introduce the following Proposition 4.12.

Proposition 4.12. *Assume that there exist OWSGs. Then, there exists a set of classical polynomial-time Turing machine* $\mathcal{M}^* := (x^*, y^*, z^*)$ *such that*

- $\Sigma[\mathcal{M}^*] := (\mathsf{KeyGen}[x^*], \mathsf{StateGen}[y^*], \mathsf{Vrfy}[z^*])$ *is a OWSG scheme that satisfies correctness and security.*
- $x^*(1^\lambda)$ *halts within* λ^3 *steps for all sufficiently large* $\lambda \in \mathbb{N}$.
- $y^*(1^\lambda, k)$ *halts within* λ^3 *steps for all sufficiently large* $\lambda \in \mathbb{N}$, *where* $k \leftarrow \mathsf{KeyGen}[x^*](1^\lambda)$.
- $z^*(1^\lambda, k, |\psi_k|)$ *halts within* λ^3 *steps for all sufficiently large* $\lambda \in \mathbb{N}$, *where* $k \leftarrow \mathsf{KeyGen}[x^*](1^\lambda)$ *and* $\psi_k \leftarrow \mathsf{StateGen}[y^*](1^\lambda, k, \psi_k)$.

This can be shown by a standard padding trick. For the reader's convenience, we describe the proof in Appendix A.

Proof of Theorem 4.9. First, let us describe some notations:

Notations.

- For a set of classical Turing machines $\mathcal{M} := x||y||z$, we write $\Sigma[\mathcal{M}] := (\mathsf{KeyGen}[x], \mathsf{StateGen}[y], \mathsf{Vrfy}[z])$ to mean the candidate of OWSG that works as follows:
  - $\mathsf{KeyGen}[x](1^\lambda)$ runs $x(1^\lambda)$, obtains a general quantum circuit $C_\lambda[x]$, runs $C_\lambda[x]$, and outputs its output.
  - $\mathsf{StateGen}[y](1^\lambda, k)$ runs $y(1^\lambda, k)$, obtains a general quantum circuit $C_{\lambda,k}[y]$, runs $C_{\lambda,k}[y]$, and outputs its output.
  - $\mathsf{Vrfy}[z](1^\lambda, k, \psi_k)$ runs $z(1^\lambda, k, |\psi_k|)$, obtains a general quantum circuit $C_{\lambda,k,|\psi_k|}[z]$, runs $C_{\lambda,k,|\psi_k|}[z]$ on input ψ_k, and outputs its output.
- For a set of classical Turing machines $\mathcal{M} := x||y||z$, we write $\widetilde{\Sigma}[\mathcal{M}] := (\widetilde{\mathsf{KeyGen}}[x], \widetilde{\mathsf{StateGen}}[y], \widetilde{\mathsf{Vrfy}}[z])$ to mean the candidate of OWSGs that works as follows:
  - $\widetilde{\mathsf{KeyGen}}[x](1^\lambda)$ runs $x(1^\lambda)$. If x does not halt within λ^3 steps, $\widetilde{\mathsf{KeyGen}}[x]$ outputs $\top$. Otherwise, obtains a general quantum circuit $C_\lambda[x]$, runs $C_\lambda[x]$, and outputs its output.
  - $\widetilde{\mathsf{StateGen}}[y](1^\lambda, k)$ outputs $\top$ if $k = \top$. Otherwise, $\widetilde{\mathsf{StateGen}}[y](1^\lambda, k)$ runs $y(1^\lambda, k)$. If y does not halt within λ^3 steps, $\widetilde{\mathsf{StateGen}}[y]$ outputs $\top$. Otherwise, obtains a general quantum circuit $C_{\lambda,k}[y]$, runs $C_{\lambda,k}[y]$, and outputs its output.
  - $\widetilde{\mathsf{Vrfy}}[z](1^\lambda, k, \psi_k)$ outputs $\top$ if $k = \top$ or $\psi_k = \top$. Otherwise, $\widetilde{\mathsf{Vrfy}}[z]$ runs $z(1^\lambda, k, |\psi_k|)$. If it does not halt within λ^3 steps, $\widetilde{\mathsf{Vrfy}}[z]$ outputs $\top$. Otherwise, obtains a general quantum circuit $C_{\lambda,k,|\psi_k|}$, runs $C_{\lambda,k,|\psi_k|}(\psi_k)$ on input ψ_k, and outputs its output.
- For any $\lambda \in \mathbb{N}$, we write $\{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}$ to mean

$$\{\widetilde{\mathsf{KeyGen}}[x], \widetilde{\mathsf{StateGen}}[y], \widetilde{\mathsf{Vrfy}}[z]\}_{x,y,z\in[\lambda]}.$$

- We consider a polynomial n such that $n(\lambda) = \lambda^3$ for all $\lambda \in \mathbb{N}$ since we combine λ^3-OWSG candidates. We write $\mathsf{Comb}.\Sigma := \mathsf{Comb}.(\mathsf{KeyGen}, \mathsf{StateGen}, \mathsf{Vrfy})$ to mean a $(1, n)$-robust OWSGs combiner for universal construction.

Construction. We give a description of $\Sigma_{\mathsf{Univ}} := (\mathsf{KeyGen}_{\mathsf{Univ}}, \mathsf{StateGen}_{\mathsf{Univ}}, \mathsf{Vrfy}_{\mathsf{Univ}})$.

$\mathsf{KeyGen}_{\mathsf{Univ}}(1^\lambda)$**:**

- Output $k \leftarrow \mathsf{Comb.KeyGen}(1^\lambda, \{(\widetilde{\mathsf{KeyGen}}[x], \widetilde{\mathsf{StateGen}}[y], \widetilde{\mathsf{Vrfy}}[z])\}_{x,y,z\in[\lambda]})$.

$\mathsf{StateGen}_{\mathsf{Univ}}(1^\lambda, k)$**:**

- Output $\psi_k \leftarrow \mathsf{Comb.StateGen}(1^\lambda, k, \{(\widetilde{\mathsf{KeyGen}}[x], \widetilde{\mathsf{StateGen}}[y], \widetilde{\mathsf{Vrfy}}[z])\}_{x,y,z\in[\lambda]})$, and output its output.

$\mathsf{Vrfy}_{\mathsf{Univ}}(1^\lambda, k, \psi_k)$**:**

- Output $\top/\bot \leftarrow \mathsf{Comb.Vrfy}(1^\lambda, k, \psi_k, \{(\widetilde{\mathsf{KeyGen}}[x], \widetilde{\mathsf{StateGen}}[y], \widetilde{\mathsf{Vrfy}}[z])\}_{x,y,z\in[\lambda]})$.

Theorem 4.9 follows from the following Lemmata 4.13 to 4.15.

Lemma 4.13. Σ_{Univ} *satisfies efficiency.*

Lemma 4.14. Σ_{Univ} *satisfies correctness.*

Lemma 4.15. *If there exist OWSGs, then* Σ_{Univ} *satisfies security.*

Proof of Lemma 4.13. Lemma 4.13 follows because $(\widetilde{\mathsf{KeyGen}}[x], \widetilde{\mathsf{StateGen}}[y], \widetilde{\mathsf{Vrfy}}[z])$ is a set of uniform QPT algorithms for any $x, y, z \in [\lambda]$, and $\mathsf{Comb}.\Sigma$ is also a set of uniform QPT algorithms. □

Proof of Lemma 4.14. From the construction, we have

$$\begin{aligned}
&\Pr\left[\top \leftarrow \mathsf{Vrfy}_{\mathsf{Univ}}(1^\lambda, k, \psi_k) : k \leftarrow \mathsf{KeyGen}_{\mathsf{Univ}}(1^\lambda), \psi_k \leftarrow \mathsf{StateGen}_{\mathsf{Univ}}(1^\lambda, k)\right] \\
&= \Pr\left[\top \leftarrow \mathsf{Comb.Vrfy}(1^\lambda, \psi_k, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \;\middle|\; \begin{array}{l} k \leftarrow \mathsf{Comb.KeyGen}(1^\lambda, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \\ \psi_k \leftarrow \mathsf{Comb.StateGen}(1^\lambda, k, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \end{array}\right].
\end{aligned}$$

Because $\widetilde{\Sigma}[\mathcal{M}]$ is a set of uniform QPT algorithms and the robust combiner $\mathsf{Comb}.\Sigma$ satisfies correctness, we have

$$\Pr\left[\top \leftarrow \mathsf{Comb.Vrfy}(1^\lambda, \psi_k, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \;\middle|\; \begin{array}{l} k \leftarrow \mathsf{Comb.KeyGen}(1^\lambda, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \\ \psi_k \leftarrow \mathsf{Comb.StateGen}(1^\lambda, k, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

This implies that Σ_{Univ} satisfies correctness.

□

Proof of Lemma 4.15. Assume that there exists an OWSG. Then, from Proposition 4.12, there exists a set of classical Turing machines (x^*, y^*, z^*) such that $x^*, y^*, z^* \in [n]$ for some $n \in \mathbb{N}$, and $x^*(1^\lambda)$, $y^*(1^\lambda)$, and $z^*(1^\lambda)$ halt within λ^3 steps for all sufficiently large security parameters $\lambda \in \mathbb{N}$, and moreover $(\mathsf{KeyGen}[x^*], \mathsf{StateGen}[y^*], \mathsf{Vrfy}[z^*])$ is an OWSG scheme that satisfies correctness and security. Furthermore, $(\widetilde{\mathsf{KeyGen}}[x^*], \widetilde{\mathsf{StateGen}}[y^*], \widetilde{\mathsf{Vrfy}}[z^*])$ also satisfies correctness and security because for all sufficiently large security parameters, $(\widetilde{\mathsf{KeyGen}}[x^*], \widetilde{\mathsf{StateGen}}[y^*], \widetilde{\mathsf{Vrfy}}[z^*])$ emulates a correct-and-secure OWSG scheme $(\mathsf{KeyGen}[x^*], \mathsf{StateGen}[y^*], \mathsf{Vrfy}[z^*])$. Therefore, for any polynomial t and QPT adversary $\mathcal{A}$, we have

$$\Pr\left[\top \leftarrow \mathsf{Comb.Vrfy}(1^\lambda, k^*, \psi_k, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \;\middle|\; \begin{array}{l} k \leftarrow \mathsf{Comb.KeyGen}(1^\lambda, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \\ \psi_k \leftarrow \mathsf{Comb.StateGen}(1^\lambda, k, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \\ k^* \leftarrow \mathcal{A}(\psi_k^{\otimes t(\lambda)}) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

This is because $\mathsf{Comb}.\Sigma$ satisfies security and $\{(\widetilde{\mathsf{KeyGen}}[x], \widetilde{\mathsf{StateGen}}[y], \widetilde{\mathsf{Vrfy}}[z])\}_{x,y,z\in[\lambda]}$ includes $(\widetilde{\mathsf{KeyGen}}[x^*], \widetilde{\mathsf{StateGen}}[y^*], \widetilde{\mathsf{Vrfy}}[z^*])$ for all sufficiently large $\lambda \in \mathbb{N}$.

Furthermore, from the construction of $(\mathsf{KeyGen}_{\mathsf{Univ}}, \mathsf{StateGen}_{\mathsf{Univ}}, \mathsf{Vrfy}_{\mathsf{Univ}})$, for all polynomial t, QPT adversary $\mathcal{A}$, and security parameters $\lambda \in \mathbb{N}$, we have

$$\Pr\left[\top \leftarrow \mathsf{Comb.Vrfy}(1^\lambda, k^*, \psi_k, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \;\middle|\; \begin{array}{l} k \leftarrow \mathsf{Comb.KeyGen}(1^\lambda, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \\ \psi_k \leftarrow \mathsf{Comb.StateGen}(1^\lambda, k, \{\widetilde{\Sigma}[\mathcal{M}]\}_{x,y,z\in[\lambda]}) \\ k^* \leftarrow \mathcal{A}(\psi_k^{\otimes t(\lambda)}) \end{array}\right]$$
$$= \Pr\left[\top \leftarrow \mathsf{Vrfy}_{\mathsf{Univ}}(1^\lambda, k^*, \psi_k) \;\middle|\; \begin{array}{l} k \leftarrow \mathsf{KeyGen}_{\mathsf{Univ}}(1^\lambda) \\ \psi_k \leftarrow \mathsf{StateGen}_{\mathsf{Univ}}(1^\lambda, k) \\ k^* \leftarrow \mathcal{A}(\psi_k^{\otimes t(\lambda)}) \end{array}\right].$$

Therefore, our universal construction Σ_{Univ} satisfies security.

□

□

5 Robust Combiner for Public-Key Quantum Money Mini-Scheme

Definition 5.1 (Robust Combiner for Public-Key Quantum Money Mini-Scheme). *A robust combiner for public-key quantum money mini-scheme is a deterministic classical polynomial-time Turing machine $\mathcal{M}$ with the following properties:*

- *$\mathcal{M}$ takes as input 1^n with $n \in \mathbb{N}$ and n-candidates for public-key quantum money mini-schemes $\{\Sigma_i := (\mathsf{Mint}_i, \mathsf{Vrfy}_i)\}_{i\in[n]}$ promised that all candidates satisfy efficiency, and outputs a single set of algorithms $\Sigma := (\mathsf{Mint}, \mathsf{Vrfy})$.*
- *If all of $\{\Sigma_i\}_{i\in[n]}$ satisfy efficiency and at least one of $\{\Sigma_i\}_{i\in[n]}$ satisfies both correctness and security, then Σ is a public-key quantum money mini-scheme that satisfies efficiency, correctness, and security.*

Theorem 5.2. *A robust combiner for public-key quantum money mini-scheme exists.*

For proving Theorem 5.2, we introduce the following Lemma 5.3.

Lemma 5.3. *Let $\Sigma = (\mathsf{Mint}, \mathsf{Vrfy})$ be a candidate for public-key quantum money mini-scheme. From Σ, we can construct a public-key quantum money mini-scheme $\Sigma^* = (\mathsf{Mint}^*, \mathsf{Vrfy}^*)$ with the following properties:*

1. *If Σ is a uniform QPT algorithm, then Σ^* is a uniform QPT algorithm.*
2. *Σ^* satisfies correctness.*
3. *If Σ is a uniform QPT algorithm and satisfies both correctness and security, then Σ^* satisfies security.*

The proof is the same as Lemma 4.4. For the reader's convenience, we describe the construction of Σ^* in Appendix B.

Proof of Theorem 5.2. Below, we consider a fixed constant n. Let us introduce some notations.

Notations.

- Let Σ_i be a candidate of public-key quantum money mini-scheme for $i \in [n]$.
- For a candidate of public-key quantum money mini-scheme Σ_i, let $\Sigma_i^* := (\mathsf{Mint}_i^*, \mathsf{Vrfy}_i^*)$ be a candidate of public-key quantum money mini-scheme derived from Lemma 5.3, which satisfies:
  - Σ_i^* is a uniform QPT algorithm if Σ_i is a uniform QPT algorithm.
  - Σ_i^* satisfies correctness.
  - Σ_i^* satisfies security if Σ_i is a uniform QPT algorithm and satisfies both correctness and security.

Construction of Robust Combiner for Public-Key Quantum Money Mini-Scheme: A robust combiner for public-key quantum money mini-scheme is a deterministic classical polynomial-time Turing machine $\mathcal{M}$ that takes as input 1^n and $\{\Sigma_i\}_{i\in[n]}$, and outputs the following set of algorithms $\Sigma = (\mathsf{Mint}, \mathsf{Vrfy})$:

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

- For all $i \in [n]$, run $(s_i^*, \rho_{s_i}^*) \leftarrow \mathsf{Mint}_i^*(1^\lambda)$.
- Output $s := \{s_i^*\}_{i\in[n]}$ and $\rho_s := \bigotimes_{i\in[n]} \rho_{s_i}^*$.

$\mathsf{Vrfy}(1^\lambda, s, \rho)$**:**

- Parse $s = \{s_i\}_{i\in[n]}$. Let ρ be a quantum state on n registers, $\{R[i]\}_{i\in[n]}$, each of which is of $|\rho_{s_i}|$ qubits, and let $\rho[i] := \mathrm{Tr}_{R[1]\cdots R[i-1]R[i+1]\cdots R[n]}(\rho)$.
- For all $i \in [n]$, run $\mathsf{Vrfy}_i^*(1^\lambda, s_i, \rho[i])$. If $\top \leftarrow \mathsf{Vrfy}_i^*(1^\lambda, s_i, \rho[i])$ for all $i \in [n]$, output $\top$. Otherwise, output $\bot$.

Theorem 5.2 follows from the following Lemmata 5.4 to 5.6.

Lemma 5.4. *If all of $\{\Sigma_i\}_{i\in[n]}$ satisfies efficiency, then Σ satisfies efficiency.*

Lemma 5.5. *Σ satisfies correctness.*

Lemma 5.6. *If all of $\{\Sigma_i\}_{i\in[n]}$ satisfies efficiency and at least one of $\{\Sigma_i\}_{i\in[n]}$ satisfies both correctness and security, then Σ satisfies security.*

Lemma 5.4 trivially follows. Lemma 5.5 follows because Σ_i^* satisfies correctness for all $i \in [n]$.

Proof of Lemma 5.6. We can prove Lemma 5.6 via a standard hybrid argument. For a reader's convenience, we describe the proof. Let Σ_x be a candidate of public-key quantum money mini-scheme that satisfies both correctness and security. Then, Σ_x^* satisfies security from Lemma 5.3. Assume that there exists a QPT adversary $\mathcal{A}$ that breaks the security of Σ, and then construct an adversary $\mathcal{B}_x$ that breaks the security of Σ_x^*. We describe $\mathcal{B}_x$:

1. $\mathcal{B}_x$ receives $\rho_{s_x}^*$ from the challenger of Σ_x^*.
2. $\mathcal{B}_x$ runs $(s_i^*, \rho_{s_i}^*) \leftarrow \mathsf{Mint}_i^*(1^\lambda)$ for all $i \in [n]\backslash x$, and sends $\{\rho_{s_i}^*\}_{i\in[n]}$ to $\mathcal{A}$.
3. $\mathcal{B}_x$ receives σ from $\mathcal{A}$. Here, σ is a quantum state on $2n$ registers, $\{R[i]\}_{i\in[2n]}$, where $R[i]$ and $R[i+n]$ are a register acting on $\left|\rho_{s_i}^*\right|$-length qubits. Let $\sigma[i] := \mathrm{Tr}_{R[1]\cdots R[i-1]R[i+1]\cdots R[2n]}(\sigma)$ for $i \in [2n]$.
4. $\mathcal{B}_x$ sends $\sigma[x]$ and $\sigma[x+n]$ to the challenger of Σ_x^*.
5. The challenger runs $\mathsf{Vrfy}_x^*(1^\lambda, s_x^*, \sigma[x])$ and $\mathsf{Vrfy}(1^\lambda, s_x^*, \sigma[x+n])$, and outputs $\top$ if both of them output $\top$.

Clearly, $\mathcal{B}_x$ perfectly simulates the challenge of Σ. Because $\mathcal{A}$ breaks the security of Σ, $\mathcal{A}$ outputs σ such that $\top \leftarrow \mathsf{Vrfy}_i^*(1^\lambda, s_i^*, \sigma[i])$ and $\top \leftarrow \mathsf{Vrfy}_i^*(1^\lambda, s_i^*, \sigma[i+n])$ for all $i \in [n]$ with non-negligible probability. Therefore, the challenger outputs $\top$ with non-negligible probability, which implies that $\mathcal{B}_x$ breaks the security of Σ_x^*. This completes the proof. □

□

5.1 Universal Construction

Definition 5.7. *We say that a set of uniform algorithms $\Sigma_{\mathsf{Univ}} = (\mathsf{Mint}, \mathsf{Vrfy})$ is a universal construction of public-key quantum money mini-scheme if Σ_{Univ} is a public-key quantum money mini-scheme as long as there exists a public-key quantum money mini-scheme.*

Theorem 5.8. *There exists a universal construction of public-key quantum money mini-scheme.*

The proof is almost the same as Theorem 4.9, and thus we skip the proof.

6 Robust Canonical Quantum Bit Commitment Combiner

Definition 6.1 (Robust Canonical Quantum Bit Commitment Combiner). *A robust canonical quantum bit commitment combiner is a deterministic classical polynomial-time Turing machine $\mathcal{M}$ with the following properties:*

- *$\mathcal{M}$ takes as input 1^n and n-deterministic classical polynomial-time Turing machine $\{\mathcal{T}_i\}_{i\in[n]}$ that produces unitary, and outputs a deterministic classical polynomial-time Turing machine $\mathcal{T}$ that produces unitary.*
- *Let $(Q_{i,0}(\lambda), Q_{i,1}(\lambda))$ be the unitary obtained by $\mathcal{T}_i(\lambda)$ and let $(Q_0(\lambda), Q_1(\lambda))$ be the unitary obtained by $\mathcal{T}(\lambda)$. If one of $\{\{Q_{i,0}(\lambda), Q_{i,1}(\lambda)\}_{\lambda\in\mathbb{N}}\}_{i\in[n]}$ satisfies computational binding and computational hiding, then $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ is a quantum bit commitment that satisfies statistical binding and computational hiding.*

In this section, we show the Theorem 6.2.

Theorem 6.2. *There exists a robust canonical quantum bit commitment combiner.*

First, let us introduce the following Proposition 6.3.

Proposition 6.3. *Let $\Sigma = \{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ be a candidate of a canonical quantum bit commitment. From Σ, we can construct a canonical quantum bit commitment $\Sigma^* := \{Q_0^*(\lambda), Q_1^*(\lambda)\}_{\lambda\in\mathbb{N}}$ such that:*

1. *Σ^* satisfies statistical binding.*
2. *If Σ satisfies computational binding and computational hiding, then Σ^* satisfies computational hiding.*

Proposition 6.3 directly follows from the following Lemma 6.4.

Lemma 6.4 (Amplifying Binding). *Let $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ be a candidate of canonical quantum bit commitment. Let $\{Q_0^*(\lambda), Q_1^*(\lambda)\}_{\lambda\in\mathbb{N}}$ be a candidate of canonical quantum bit commitment described as follows:*

- *If $\mathbf{C}$ and $\mathbf{R}$ are the commitment and reveal registers of $(Q_0(\lambda), Q_1(\lambda))$, and $\widetilde{\mathbf{C}}$ and $\widetilde{\mathbf{R}}$ are the commitment and reveal registers of $(\widetilde{Q_0}(\lambda), \widetilde{Q_1}(\lambda))$, which is the flavor conversion of $(Q_0(\lambda), Q_1(\lambda))$ introduced in Lemma 3.12, then the commitment and reveal registers of $(Q_0^*(\lambda), Q_1^*(\lambda))$ are defined as $\mathbf{C}^* := \left(\mathbf{C}^{\otimes\lambda}, \widetilde{\mathbf{C}}^{\otimes\lambda}\right)$, and $\mathbf{R}^* := \left(\mathbf{R}^{\otimes\lambda}, \widetilde{\mathbf{R}}^{\otimes\lambda}\right)$.*
- *For $b\in\{0,1\}$, the unitary $Q_b^*(\lambda)$ is defined as follows:*

$$Q_b^*(\lambda) := (Q_b(\lambda)\otimes\widetilde{Q_b}(\lambda))^{\otimes\lambda}.$$

Then, the following is satisfied:

1. *$\{Q_0^*(\lambda), Q_1^*(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies statistical binding.*
2. *If $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies computational hiding and computational binding, then $\{Q_0^*(\lambda), Q_1^*(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies computational hiding.*

Proof of Lemma 6.4. Below, we fix the security parameter λ, and write (Q_0, Q_1), $(\widetilde{Q_0}, \widetilde{Q_1})$ and (Q_0^*, Q_1^*) to mean $(Q_0(\lambda), Q_1(\lambda))$, $(\widetilde{Q_0}(\lambda), \widetilde{Q_1}(\lambda))$ and $(Q_0^*(\lambda), Q_1^*(\lambda))$, respectively.

Proof of the first item We define

$$\rho_b := \mathrm{Tr}_{\mathbf{R}}(Q_b|0\rangle_{\mathbf{C},\mathbf{R}}) \text{ and } \widetilde{\rho_b} := \mathrm{Tr}_{\widetilde{\mathbf{R}}}(\widetilde{Q_b}|0\rangle_{\widetilde{\mathbf{C}},\widetilde{\mathbf{R}}}) \text{ and } \rho_b^* := \mathrm{Tr}_{\mathbf{R}^*}(Q_b^*|0\rangle_{\mathbf{C}^*,\mathbf{R}^*}).$$

From the construction of Q_0^* and Q_1^*, we have

$$(\rho_b\otimes\widetilde{\rho_b})^{\otimes\lambda} := \mathrm{Tr}_{\mathbf{R}^*}(Q_b^*|0\rangle_{\mathbf{C}^*,\mathbf{R}^*}).$$

Let $0 \leq f \leq 1$ be some value such that

$$F(\rho_0, \rho_1) = f.$$

We have

$$\mathsf{TD}(\rho_0, \rho_1) \leq \sqrt{1 - F(\rho_0, \rho_1)} \leq \sqrt{1 - f}.$$

In particular, this implies that (Q_0, Q_1) satisfies $\sqrt{1-f}$-statistical hiding. From Lemma 3.12, this implies that $(\widetilde{Q_0}, \widetilde{Q_1})$ satisfies $(1-f)^{1/4}$-statistical binding. Furthermore, from Uhlmann's theorem (Theorem 3.6), this implies that

$$F(\widetilde{\rho_0}, \widetilde{\rho_1}) \leq (1-f)^{1/2},$$

which we prove later.

Therefore, we have

$$F(\rho_0^*, \rho_1^*) = F\left((\rho_0 \otimes \widetilde{\rho_0})^{\otimes \lambda}, (\rho_1 \otimes \widetilde{\rho_1})^{\otimes \lambda}\right) \leq F(\rho_0, \rho_1)^\lambda F(\widetilde{\rho_0}, \widetilde{\rho_1})^\lambda \leq f^\lambda (1-f)^{\lambda/2} \leq 2^{-\lambda/2},$$

which implies that (Q_0^*, Q_1^*) satisfies statistical binding.

Now, we show that if $(\widetilde{Q_0}, \widetilde{Q_1})$ satisfies $(1-f)^{1/4}$-statistical binding, then $F(\widetilde{\rho_0}, \widetilde{\rho_1}) \leq (1-f)^{1/2}$. For contradiction, assume that $F(\widetilde{\rho_0}, \widetilde{\rho_1}) > (1-f)^{1/2}$, and then show that $(\widetilde{Q_0}, \widetilde{Q_1})$ does not satisfies $(1-f)^{1/4}$-statistical binding. From Uhlmann's Theorem (Theorem 3.6), there exists some unitary $U_{\widetilde{\mathbf{R}}}$ acting on the register $\widetilde{\mathbf{R}}$ such that

$$F(\widetilde{\rho_0}, \widetilde{\rho_1}) = \left|\langle 0|_{\widetilde{\mathbf{C}}, \widetilde{\mathbf{R}}} \widetilde{Q_0}^\dagger (I_{\widetilde{\mathbf{C}}} \otimes U_{\widetilde{\mathbf{R}}}) \widetilde{Q_1} |0\rangle_{\widetilde{\mathbf{C}}, \widetilde{\mathbf{R}}}\right|^2 > (1-f)^{1/2}.$$

Now, we have

$$\begin{aligned}
&\left\| \left(\langle 0|_{\widetilde{\mathbf{C}}, \widetilde{\mathbf{R}}} \widetilde{Q_0}^\dagger \otimes I_{\widetilde{\mathbf{C}}, \widetilde{\mathbf{Z}}}\right) (I_{\widetilde{\mathbf{C}}, \widetilde{\mathbf{Z}}} \otimes U_{\widetilde{\mathbf{R}}}) \widetilde{Q_1} |0\rangle_{\widetilde{\mathbf{C}}, \widetilde{\mathbf{R}}} |\tau\rangle_{\widetilde{\mathbf{Z}}} \right\|_1 \\
&= \left\| \left(\langle 0|_{\widetilde{\mathbf{C}}, \widetilde{\mathbf{R}}} \widetilde{Q_0}^\dagger (I_{\widetilde{\mathbf{C}}} \otimes U_{\widetilde{\mathbf{R}}}) \widetilde{Q_1} |0\rangle_{\widetilde{\mathbf{C}}, \widetilde{\mathbf{R}}}\right) |\tau\rangle_{\widetilde{\mathbf{Z}}} \right\|_1 = \left|\langle 0|_{\widetilde{\mathbf{C}}, \widetilde{\mathbf{R}}} \widetilde{Q_0}^\dagger (I_{\widetilde{\mathbf{C}}} \otimes U_{\widetilde{\mathbf{R}}}) \widetilde{Q_1} |0\rangle_{\widetilde{\mathbf{C}}, \widetilde{\mathbf{R}}}\right| > (1-f)^{\frac{1}{4}},
\end{aligned}$$

which contradicts that$(\widetilde{Q_0}, \widetilde{Q_1})$ satisfies $(1-f)^{1/4}$-statistical binding.

Proof of the second item. We prove that (Q_0^*, Q_1^*) satisfies computational hiding if (Q_0, Q_1) satisfies computational hiding and computational binding. Because $(\widetilde{Q_0}, \widetilde{Q_1})$ is the flavor conversion of (Q_0, Q_1), $(\widetilde{Q}_0, \widetilde{Q}_1)$ also satisfies computational hiding. Therefore, we can reduce the computational hiding of (Q_0^*, Q_1^*) to those of (Q_0, Q_1) and $(\widetilde{Q}_0, \widetilde{Q}_1)$ by a standard hybrid argument. □

Proof of Theorem 6.2. Below, we consider some fixed constant n. For $i \in [n]$, let $\mathcal{T}_i$ be a deterministic classical Turing machine that takes as input 1^λ, and outputs $(Q_{i,0}(\lambda), Q_{i,1}(\lambda))$. Let $\Sigma_i := \{Q_{i,0}(\lambda), Q_{i,1}(\lambda)\}_{\lambda \in \mathbb{N}}$ be a candidate of canonical quantum bit commitment. Let $\Sigma_i^* := \{Q_{i,0}^*(\lambda), Q_{i,1}^*(\lambda)\}_{\lambda \in \mathbb{N}}$ be a candidate of canonical quantum bit commitment such that:

1. Σ_i^* satisfies statistical binding.
2. Σ_i^* satisfies computational hiding if Σ_i satisfies computational hiding and computational binding.

Note that such a canonical quantum bit commitment is obtained from Proposition 6.3.

A robust canonical quantum bit commitment combiner is a deterministic classical polynomial-time Turing machine $\mathcal{M}$ that takes as input 1^n and $\{\mathcal{T}_i\}_{i \in [n]}$, and outputs a deterministic classical polynomial-time Turing machine $\mathcal{T}$ that works as follows. $\mathcal{T}$ takes as input 1^λ and outputs the following QPT unitary $(\mathsf{Comb}.Q_0(\lambda), \mathsf{Comb}.Q_1(\lambda))$:

- If $\mathbf{C}_i^*$ and $\mathbf{R}_i^*$ are the commitment register and the reveal register of $(Q_{i,0}^*(\lambda), Q_{i,1}^*(\lambda))$, then the commitment and reveal register of $(\mathsf{Comb}.Q_0(\lambda), \mathsf{Comb}.Q_1(\lambda))$ are defined as $\mathbf{C} := \{\mathbf{C}_i^*\}_{i \in [n]}$ and $\mathbf{R} = (\{\mathbf{R}_i^*\}_{i \in [n]}, \{\mathbf{D}_i^*\}_{i \in [n]})$, where $\mathbf{D}_i^*$ is an additional one-qubit register for $i \in [n]$.

- For $b \in \{0,1\}$, the unitary $\mathsf{Comb}.Q_b$ is defined as follows:

$$\mathsf{Comb}.Q_b(\lambda) := \left(\sum_{r\in\{0,1\}^n} \bigotimes_{i\in[n]} (Q^*_{i,r_i}(\lambda) \otimes |r_i\rangle\langle r_i|_{\mathbf{D}^*_i})\right)\left(\bigotimes_{i\in[n]} I_{\mathbf{C}^*_i,\mathbf{R}^*_i} \otimes X^b_{\mathbf{D}^*_1} \bigotimes_{i\in\{2,\cdots,n\}} \mathsf{CNOT}_{\mathbf{D}^*_1,\mathbf{D}^*_i} \bigotimes_{i\in\{2,\cdots,n\}} H_{\mathbf{D}^*_i}\right).$$

  Here, r_i is the i-th bit of r and $\mathsf{CNOT}_{\mathbf{D}^*_1,\mathbf{D}^*_i}$ is a CNOT gate, where $\mathbf{D}^*_1$ is a target register and $\mathbf{D}^*_i$ is a control register. Note that we have

$$\mathsf{Comb}.Q_b(\lambda)|0\rangle_{\mathbf{C},\mathbf{R}} = \frac{1}{2^{(n-1)/2}} \sum_{\{r:\sum_{i\in[n]} r_i=b\}} \bigotimes_{i\in[n]} (Q^*_{i,r_i}(\lambda)|0\rangle_{\mathbf{C}^*_i,\mathbf{R}^*_i} \otimes |r_i\rangle_{\mathbf{D}^*_i}).$$

We have the following Lemmata 6.5 and 6.6, which we prove later. Therefore, Theorem 6.2 holds.

Lemma 6.5. $\{\mathsf{Comb}.Q_0(\lambda), \mathsf{Comb}.Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ *satisfies statistical binding.*

Lemma 6.6. *If one of* $\{\Sigma_i\}_{i\in[n]}$ *satisfies computational hiding and computational binding, then* $\{\mathsf{Comb}.Q_0(\lambda), \mathsf{Comb}.Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ *satisfies computational hiding.*

□

Proof of Lemma 6.5. Below, we fix security parameter λ, and write $\{Q^*_{i,0}, Q^*_{i,1}\}_{i\in[n]}$ and $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ to mean $\{Q^*_{i,0}(\lambda), Q^*_{i,1}(\lambda)\}_{i\in[n]}$ and $(\mathsf{Comb}.Q_0(\lambda), \mathsf{Comb}.Q_1(\lambda))$, respectively. Let us denote $\rho^*_{i,b} := \mathrm{Tr}_{\mathbf{R}_i}(Q^*_{i,b}|0\rangle_{\mathbf{C}_i,\mathbf{R}_i})$. We write $\mathsf{Comb}.\rho_b := \mathrm{Tr}_{\mathbf{R}}(\mathsf{Comb}.Q_b|0\rangle_{\mathbf{C},\mathbf{R}})$ and write R to mean $\sum_{i\in[n]} r_i$. Note that we have

$$\mathsf{Comb}.\rho_b = \frac{1}{2^{n-1}} \sum_{\{r:R=b\}} \bigotimes_{i\in[n]} \rho^*_{i,r_i}.$$

Now, we show that

$$\mathsf{TD}(\mathsf{Comb}.\rho_0, \mathsf{Comb}.\rho_1) \geq 1 - \mathsf{negl}(\lambda).$$

For that, it is sufficient to show that there exists a POVM measurement $\{\mathsf{Comb}.\Pi_0, \mathsf{Comb}.\Pi_1\}$ that distinguishes $\mathsf{Comb}.\rho_0$ from $\mathsf{Comb}.\rho_1$. From Lemma 6.4, all Σ^*_i satisfies statistical binding. This implies that we have $\mathsf{TD}(\rho^*_{i,0}, \rho^*_{i,1}) \geq 1 - \mathsf{negl}(\lambda)$. Moreover, this implies that there exists a two-outcome POVM measurement $\{\Pi^*_{i,0}, \Pi^*_{i,1}\}$ such that

$$\mathrm{Tr}\left(\Pi^*_{i,0}(\rho^*_{i,0} - \rho^*_{i,1})\right) = \mathrm{Tr}\left(\Pi^*_{i,1}(\rho^*_{i,1} - \rho^*_{i,0})\right) \geq 1 - \mathsf{negl}(\lambda).$$

We introduce the two-outcome POVM measurement $\{\mathsf{Comb}.\Pi_0 := \sum_{\{r:R=0\}} \bigotimes_{i\in[n]} \Pi^*_{i,r_i}, \mathsf{Comb}.\Pi_1 := \sum_{\{r:R=1\}} \bigotimes_{i\in[n]} \Pi^*_{i,r_i}\}$.

Then, we have

$$\begin{aligned}
\mathsf{TD}(\mathsf{Comb}.\rho_0, \mathsf{Comb}.\rho_1) &\geq \mathrm{Tr}(\mathsf{Comb}.\Pi_0(\mathsf{Comb}.\rho_0 - \mathsf{Comb}.\rho_1)) \\
&= \frac{1}{2^{n-1}} \mathrm{Tr}\left(\mathsf{Comb}.\Pi_0 \left(\sum_{\{r:R=0\}} \bigotimes_{i\in[n]} \rho^*_{i,r_i} - \sum_{\{r:R=1\}} \bigotimes_{i\in[n]} \rho^*_{i,r_i}\right)\right) \\
&= \frac{1}{2^{n-1}} \mathrm{Tr}\left(\mathsf{Comb}.\Pi_0 \left(\sum_{r} \bigotimes_{i\in[n]} (-1)^{r_i} \rho^*_{i,r_i}\right)\right) \\
&= \frac{1}{2^{n-1}} \mathrm{Tr}\left(\mathsf{Comb}.\Pi_0 \bigotimes_{i\in[n]} \left(\rho^*_{i,0} - \rho^*_{i,1}\right)\right) \\
&= \frac{1}{2^{n-1}} \mathrm{Tr}\left(\left(\sum_{\{r:R=0\}} \bigotimes_{i\in[n]} \Pi^*_{i,r_i}\right)\left(\bigotimes_{i\in[n]} \rho^*_{i,0} - \rho^*_{i,1}\right)\right) \\
&= \frac{1}{2^{n-1}} \mathrm{Tr}\left(\sum_{\{r:R=0\}} \left(\bigotimes_{i\in[n]} \Pi^*_{i,r_i} \left(\rho^*_{i,0} - \rho^*_{i,1}\right)\right)\right) \\
&= \frac{1}{2^{n-1}} \sum_{\{r:R=0\}} \mathrm{Tr}\left(\bigotimes_{i\in[n]} \Pi^*_{i,r_i} \left(\rho^*_{i,0} - \rho^*_{i,1}\right)\right) \\
&= \frac{1}{2^{n-1}} \sum_{\{r:R=0\}} \prod_{i\in[n]} \mathrm{Tr}\left(\Pi^*_{i,r_i} \left(\rho^*_{i,0} - \rho^*_{i,1}\right)\right) \\
&= \prod_{i\in[n]} \mathrm{Tr}\left(\Pi^*_{i,0} \left(\rho^*_{i,0} - \rho^*_{i,1}\right)\right) \geq (1 - \mathsf{negl}(\lambda))^n \geq 1 - n \cdot \mathsf{negl}(\lambda).
\end{aligned}$$

Here, we have used that $\mathrm{Tr}(A+B) = \mathrm{Tr}(A) + \mathrm{Tr}(B)$ in the sixth equation, and we have used $\mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A)\,\mathrm{Tr}(B)$ in the seventh equation, and we have used that

$$\prod_{i\in[n]} \mathrm{Tr}\left(\Pi^*_{i,r_i} \left(\rho^*_{i,0} - \rho^*_{i,1}\right)\right) = \prod_{i\in[n]} \mathrm{Tr}\left(\Pi^*_{i,0} \left(\rho^*_{i,0} - \rho^*_{i,1}\right)\right)$$

for all $r \in \{0,1\}^n$ with $\sum_{i\in[n]} r_i = 0$ in the final equation.

Furthermore, we have $F(\mathsf{Comb}.\rho_0, \mathsf{Comb}.\rho_1) \leq 1 - \mathsf{TD}(\mathsf{Comb}.\rho_0, \mathsf{Comb}.\rho_1)^2 \leq 2n \cdot \mathsf{negl}(\lambda)$. From Uhlmann's theorem (Theorem 3.6), this implies that $\{\mathsf{Comb}.Q_0(\lambda), \mathsf{Comb}.Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies statistical binding. □

Proof of Lemma 6.6. Below, we fix security parameter λ, and write $\{Q^*_{i,0}, Q^*_{i,1}\}_{i\in[n]}$ and $(\mathsf{Comb}.Q_0, \mathsf{Comb}.Q_1)$ to mean $\{Q^*_{i,0}(\lambda), Q^*_{i,1}(\lambda)\}_{i\in[n]}$ and $(\mathsf{Comb}.Q_0(\lambda), \mathsf{Comb}.Q_1(\lambda))$, respectively. Let $\rho^*_{i,b} := \mathrm{Tr}_{\mathbf{R}_i}(Q^*_{i,b}\,|0\rangle_{\mathbf{C}_i,\mathbf{R}_i})$. We show that $\{\mathsf{Comb}.Q_0(\lambda), \mathsf{Comb}.Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies computational hiding as long as one of $\{\Sigma_i\}_{i\in[n]}$ satisfies computational hiding and computational binding. Let Σ_x be the canonical quantum bit commitment that satisfies computational hiding and computational binding. Then, from Lemma 6.4, Σ^*_x satisfies computational hiding. Now, we introduce the following sequence of hybrid experiments against QPT adversary $\mathcal{A}$.

$\mathsf{Hyb}_0(b)$**:**

1. The challenger sends $\mathsf{Comb}.\rho_b$ to $\mathcal{A}$.
2. $\mathcal{A}$ outputs b^*.

$\mathsf{Hyb}_1(b)$**:**

1. The challenger randomly samples $r_i \leftarrow \{0,1\}$ for all $i \in [n]\backslash x$. We write R to mean $\sum_{i\in[n]\backslash x} r_i$.

2. The challenger sends $\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,R+b} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n}$ to $\mathcal{A}$.
3. $\mathcal{A}$ outputs b^*.

$\mathsf{Hyb}_2(b)$**:**

1. The challenger randomly samples $r_i \leftarrow \{0,1\}$ for all $i \in [n]\backslash x$.
2. The challenger sends $\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,0} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n}$.
3. $\mathcal{A}$ outputs b^*.

We have the following Propositions 6.7 to 6.9. Therefore, we have

$$|\Pr[\mathsf{Hyb}_0(0)=1] - \Pr[\mathsf{Hyb}_0(1)=1]| \leq \mathsf{negl}(\lambda),$$

which implies that $\{\mathsf{Comb}.\rho_0(\lambda), \mathsf{Comb}.\rho_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies computational hiding.

Proposition 6.7. $\Pr[\mathsf{Hyb}_0(b)=1] = \Pr[\mathsf{Hyb}_1(b)=1]$ *for* $b \in \{0,1\}$.

Proposition 6.8. *If* Σ^*_x *satisfies computational hiding, then*

$$|\Pr[\mathsf{Hyb}_1(b)=1] - \Pr[\mathsf{Hyb}_2(b)=1]| \leq \mathsf{negl}(\lambda)$$

for each $b \in \{0,1\}$.

Proposition 6.9. $\Pr[\mathsf{Hyb}_2(0)=1] = \Pr[\mathsf{Hyb}_2(1)=1]$.

Propositions 6.7 and 6.9 trivially follows, and thus we omit the proof.

Proof of Proposition 6.8. For simplicity, we write $\sum_r$ to mean that $\sum_{\{r:r_i\in\{0,1\}}$ for $_{i\in[n]\backslash x\}}$, and recall that $R := \sum_{i\in[n]\backslash x} r_i$.

Then, we have

$$\begin{aligned}
&\Pr[\mathsf{Hyb}_1(0)=1] - \Pr[\mathsf{Hyb}_2(0)=1] \\
&= \frac{1}{2^{n-1}}\sum_r \left(\Pr\left[1 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,R} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right]\right) \\
&\quad - \frac{1}{2^{n-1}}\sum_r \left(\Pr\left[1 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,0} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right]\right) \\
&= \frac{1}{2^{n-1}}\sum_r \Bigg(\Pr\left[1 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,R} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right] \\
&\qquad - \Pr\left[1 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,0} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right]\Bigg) \\
&= \frac{1}{2^{n-1}}\sum_{\{r:R=1\}} \Bigg(\Pr\left[1 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,1} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right] \\
&\qquad - \Pr\left[1 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,0} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right]\Bigg) \\
&= \frac{1}{2^{n-1}}\sum_{\{r:R=1\}} \Bigg(\Pr\left[0 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,0} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right] \\
&\qquad - \Pr\left[0 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,1} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right]\Bigg).
\end{aligned}$$

For showing a contradiction, assume that there exists some constant c and a QPT adversary $\mathcal{A}$ such that

$$|\Pr[\mathsf{Hyb}_1(0)=1] - \Pr[\mathsf{Hyb}_2(0)=1]| \geq 1/\lambda^c$$

for all sufficiently large security parameters $\lambda \in \mathbb{N}$ and then construct a QPT algorithm $\mathcal{B}_x$ that breaks the computational hiding of Σ^*_x.

1. $\mathcal{B}_x$ receives $\rho^*_{x,b}$ from the challenger of Σ^*_x, where b is randomly sampled from $\{0,1\}$.
2. $\mathcal{B}_x$ randomly samples $r_i \leftarrow \{0,1\}$ for all $i \in [n]\backslash x$.
3. $\mathcal{B}_x$ sends $\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,b} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n}$ to $\mathcal{A}$.
4. $\mathcal{B}_x$ receives b^* from $\mathcal{A}$.
5. $\mathcal{B}_x$ outputs b^*+1 if $R=1$, and outputs 0 otherwise, where $R = \sum_{i\in[n]\backslash x} r_i$.

We compute $|\Pr[1 \leftarrow \mathcal{B}_x : b=0] - \Pr[1 \leftarrow \mathcal{B}_x : b=1]|$. It holds that

$$
\begin{aligned}
&|\Pr[1 \leftarrow \mathcal{B}_x : b=0] - \Pr[1 \leftarrow \mathcal{B}_x : b=1]| \\
&= \frac{1}{2}|\Pr[0 \leftarrow \mathcal{A} : b=0, R=1] - \Pr[0 \leftarrow \mathcal{A} : b=1, R=1]| \\
&= \frac{1}{2^n}\Bigg| \sum_{\{r:R=1\}} \left(\Pr\left[0 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,0} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right]\right) \\
&- \sum_{\{r:R=1\}} \left(\Pr\left[0 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,1} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right]\right)\Bigg| \\
&= \frac{1}{2^n}\Bigg| \sum_{\{r:R=1\}} \Bigg(\Pr\left[0 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,0} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right] \\
&- \Pr\left[0 \leftarrow \mathcal{A}(\rho^*_{1,r_1} \otimes \cdots \otimes \rho^*_{x-1,r_{x-1}} \otimes \rho^*_{x,1} \otimes \rho^*_{x+1,r_{x+1}} \otimes \cdots \otimes \rho^*_{n,r_n})\right]\Bigg)\Bigg| \\
&= \frac{1}{2}|\Pr[\mathsf{Hyb}_1(0)=1] - \Pr[\mathsf{Hyb}_2(0)=1]|.
\end{aligned}
$$

This implies that if there exists a QPT adversary such that $|\Pr[\mathsf{Hyb}_1(0)=1] - \Pr[\mathsf{Hyb}_2(0)=1]|$ is non-negligible, then $\mathcal{B}_x$ breaks the computational hiding of Σ^*_x. Therefore, we have

$$|\Pr[\mathsf{Hyb}_1(0)=1] - \Pr[\mathsf{Hyb}_2(0)=1]| \leq \mathsf{negl}(\lambda).$$

In a similar way, we can prove that

$$|\Pr[\mathsf{Hyb}_1(1)=1] - \Pr[\mathsf{Hyb}_2(1)=1]| \leq \mathsf{negl}(\lambda).$$

□

□

6.1 Universal Construction

Definition 6.10. *We say that a sequence of uniform QPT unitaries $\Sigma_{\mathsf{Univ}} = \{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ is a universal construction of canonical quantum bit commitment if Σ_{Univ} is canonical quantum bit commitment as long as there exists canonical quantum bit commitment.*

Theorem 6.11. *There exists a universal construction of canonical quantum bit commitment.*

The proof is almost the same as Theorem 4.9, and thus we skip the proof.

7 Robust Combiner for Unclonable Encryption

Definition 7.1 (Robust Combiner for Unclonable Secret-Key Encryption). *A robust combiner for (one-time) unclonable secret-key encryption with $\ell(\lambda)$-bit plaintexts is a deterministic classical polynomial-time Turing machine $\mathcal{M}$ with the following properties:*

- *$\mathcal{M}$ takes as input 1^n with $n \in \mathbb{N}$ and n-candidates (one-time) unclonable secret-key encryption with $\ell(\lambda)$-bit plaintexts $\{\Sigma_i := (\mathsf{KeyGen}_i, \mathsf{Enc}_i, \mathsf{Dec}_i)\}_{i\in[n]}$ promised that all candidates satisfies efficiency, and outputs a set of algorithms $\Sigma := (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$.*
- *If all of $\{\Sigma_i\}_{i\in[n]}$ satisfies efficiency and at least one of $\{\Sigma_i\}_{i\in[n]}$ satisfies correctness, (one-time) IND-CPA security and (one-time) unclonable IND-CPA security, then Σ is (one-time) unclonable secret-key encryption for $\ell(\lambda)$-bit plaintexts that satisfies efficiency, correctness, (one-time) IND-CPA security and (one-time) unclonable IND-CPA security.*

In this section, we prove the following Theorem 7.2.

Theorem 7.2. *There exists a robust combiner for (one-time) unclonable secret-key encryption with $\ell(\lambda)$-bit plaintexts for all polynomial ℓ.*

As a corollary, we obtain the following Corollary 7.3.

Corollary 7.3. *There exists a robust combiner for unclonable public-key encryption with $\ell(\lambda)$-bit plaintexts for all polynomial ℓ.*

Proof of Corollary 7.3. We give a rough sketch of the proof.

Corollary 7.3 follows from the following observations. We can trivially obtain one-time unclonable SKE from unclonable PKE. From Theorem 7.2, we have a robust combiner for one-time unclonable SKE. Furthermore, we can trivially construct PKE with quantum ciphertexts from unclonable PKE. It is known that there exists a robust PKE combiner [HKN+05], and we observe that we can also construct a robust combiner for PKE with quantum ciphertexts in the same way. Moreover, we can construct unclonable PKE from one-time unclonable SKE, and PKE with quantum ciphertexts. This is because we can construct unclonable PKE from one-time SKE and receiver non-committing encryption with quantum ciphertexts [3] (For the detail, see Appendix E), and receiver non-committing encryption with quantum ciphertexts can be constructed from PKE with quantum ciphertexts in the same way as the classical ciphertext case [CHK05, KNTY19].

By combining these observations, we can construct a robust combiner for unclonable PKE as follows. Given candidates of unclonable PKE $\{\Sigma_i\}_{i\in[n]}$, we first use a robust combiner for one-time unclonable SKE, and obtain a new candidate of one-time unclonable SKE Σ_{SKE} regarding each candidate Σ_i as a one-time unclonable SKE scheme. Next, we use a robust combiner for PKE with quantum ciphertexts and obtain a new candidate of PKE with quantum ciphertexts Σ_{PKE} regarding each candidate Σ_i as a (not necessarily unclonable) PKE scheme. Then, we construct a receiver non-committing encryption with quantum ciphertexts Σ_{NCE} from Σ_{PKE}. Finally, we construct unclonable PKE $\Sigma_{\mathsf{unclone}}$ from one-time unclonable SKE Σ_{SKE} and receiver non-committing encryption with quantum ciphertexts Σ_{NCE}. □

For proving Theorem 7.2, we introduce the following Lemma 7.4.

Lemma 7.4. *Let Σ be a candidate for (one-time) unclonable secret-key encryption with $\ell(\lambda)$-bit plaintexts. From Σ, we can construct a (one-time) unclonable secret-key encryption with $\ell(\lambda)$-bit plaintexts $\Sigma^* := (\mathsf{KeyGen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$ such that:*

1. *Σ^* is a uniform QPT algorithm, if Σ is a uniform QPT algorithm.*

[3] [AK21] shows that unclonable PKE can be constructed from one-time unclonable SKE and PKE with classical ciphertexts. Note that it is unclear whether we can construct unclonable PKE from one-time SKE and PKE with "quantum" ciphertexts in the same way as [AK21]. This is because they use the existence of OWFs in their proof although it is unclear whether PKE with quantum ciphertexts implies OWFs. Therefore, we use the technique of [HMNY21] instead. (For the detail, see Appendix E)

2. Σ^* *satisfies perfect correctness.*

3. Σ^* *satisfies (one-time) IND-CPA security and (one-time) unclonable IND-CPA security if* Σ *is a uniform QPT algorithm and satisfies correctness, (one-time) IND-CPA security and (one-time) unclonable IND-CPA security.*

The proof is almost the same as Lemma 4.4. For the reader's convenience, we describe the construction of Σ^* in Appendix D.

Proof of Theorem 7.2. Below, we consider a fixed constant n and a fixed polynomial ℓ. Let us describe some notations:

Notations.

- Let Σ_i be a candidate of (one-time) unclonable secret-key encryption with $\ell(\lambda)$-length for $i \in [n]$.
- For a candidate of (one-time) unclonable secret-key encryption with $\ell(\lambda)$-bit plaintexts Σ_i, let $\Sigma_i^* := (\mathsf{KeyGen}_i^*, \mathsf{Enc}_i^*, \mathsf{Dec}_i^*)$ be a candidate of (one-time) unclonable secret-key encryption with $\ell(\lambda)$-bit plaintexts derived from Lemma 7.4, which satisfies:
  - Σ_i^* is a uniform QPT algorithm, if Σ_i is a uniform QPT algorithm.
  - Σ_i^* satisfies correctness.
  - Σ_i^* satisfies (one-time) IND-CPA security and (one-time) unclonable IND-CPA security if Σ_i is uniform QPT algorithm and satisfies correctness, (one-time) IND-CPA security, and (one-time) unclonable IND-CPA security.

Construction of Robust (One-Time) Unclonable Secret-Key Encryption. A robust combiner for (one-time) unclonable secret-key encryption with $\ell(\lambda)$-bit plaintexts is a deterministic classical polynomial-time Turing machine that takes as input 1^n and $\{\Sigma_i\}_{i\in[n]}$, and outputs the following set of algorithms $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$:

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

- For all $i \in [n]$, run $\mathsf{sk}_i^* \leftarrow \mathsf{KeyGen}_i^*(1^\lambda)$.
- Output $\mathsf{sk} := \{\mathsf{sk}_i^*\}_{i\in[n]}$.

$\mathsf{Enc}(1^\lambda, \mathsf{sk}, m)$**:**

- For all $i \in [n]$, sample $r_i \leftarrow \{0,1\}^{\ell(\lambda)}$ promised that $\sum_{i\in[n]} r_i = m$, where the $\ell(\lambda)$ is the length of plaintext m.
- For all $i \in [n]$, run $\mathsf{CT}_i^* \leftarrow \mathsf{Enc}_i^*(1^\lambda, \mathsf{sk}_i^*, r_i)$ for all $i \in [n]$.
- Output $\mathsf{CT} := \{\mathsf{CT}_i^*\}_{i\in[n]}$.

$\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT})$**:**

- Run $r_i^* \leftarrow \mathsf{Dec}_i^*(1^\lambda, \mathsf{sk}_i^*, \mathsf{CT}_i^*)$ for all $i \in [n]$.
- Output $\sum_{i\in[n]} r_i^*$.

Theorem 7.2 follows from the following Lemmata 7.5 to 7.8.

Lemma 7.5. *If all of* $\{\Sigma_i\}_{i\in[n]}$ *satisfies efficiency,* Σ *satisfies efficiency.*

Lemma 7.6. Σ *satisfies correctness.*

Lemma 7.7. *If all of* $\{\Sigma_i\}_{i\in[n]}$ *satisfies efficiency and one of* $\{\Sigma_i\}_{i\in[n]}$*, satisfies both correctness and (one-time) IND-CPA security, then* Σ *satisfies (one-time) IND-CPA security.*

Lemma 7.8. *If all of* $\{\Sigma_i\}_{i\in[n]}$ *satisfies efficiency and one of* $\{\Sigma_i\}_{i\in[n]}$*, satisfies both correctness and (one-time) unclonable IND-CPA security, then* Σ *satisfies (one-time) unclonable IND-CPA security.*

Lemmata 7.5 and 7.6 trivially follows, and thus we skip the proof. The proof of Lemma 7.7 is the same as that of Lemma 7.8, and thus we skip the proof.

Proof of Lemma 7.8. We prove the Lemma 7.8 via a standard hybrid argument. For the reader's convenience, we describe the proof. For simplicity, we consider the one-time case where Σ_i is a candidate of one-time unclonable secret-key encryption for each $i \in [n]$. We show that Σ satisfies unclonable IND-CPA security as long as all of $\{\Sigma_i\}_{i\in[n]}$ satisfy efficiency and one of $\{\Sigma_i\}_{i\in[n]}$ satisfies one-time unclonable IND-CPA security. Let Σ_x be the candidate for one-time unclonable secret-key encryption that satisfies both correctness and one-time unclonable IND-CPA security. Then, Σ_x^* satisfies unclonable IND-CPA security from Lemma 7.4. Assume that there exists a QPT adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ that breaks the one-time unclonable IND-CPA security of Σ, and then construct a set of QPT adversaries $(\widetilde{\mathcal{A}_x}, \widetilde{\mathcal{B}_x}, \widetilde{\mathcal{C}_x})$ that breaks the one-time unclonable security of Σ_x^*.

1. $\widetilde{\mathcal{A}_x}$ receives (m_0, m_1) from $\mathcal{A}$.
2. $\widetilde{\mathcal{A}_x}$ samples $r_i \leftarrow \{0,1\}^{\ell(\lambda)}$ for all $i \in [n]\backslash x$, and sends $(M_0 := m_0 + \sum_{i\in[n]\backslash x} r_i, M_1 := m_1 + \sum_{i\in[n]\backslash x} r_i)$ to the challenger of Σ_x^*.
3. The challenger of Σ_x^* samples $b \leftarrow \{0,1\}$, and runs $\mathsf{CT}_x[M_b]^* \leftarrow \mathsf{Enc}_x^*(1^\lambda, \mathsf{sk}_x^*, M_b)$.
4. $\widetilde{\mathcal{A}_x}$ receives from $\mathsf{CT}_x[M_b]^*$, runs $\mathsf{sk}_i^* \leftarrow \mathsf{KeyGen}_i^*(1^\lambda)$ for all $i \in [n]\backslash x$, samples r_i for all $i \in [n]\backslash x$, runs $\mathsf{CT}_i[r_i]^* \leftarrow \mathsf{Enc}_i^*(1^\lambda, \mathsf{sk}_i^*, r_i)$, and sends $(\mathsf{CT}_1[r_1]^*, \cdots, \mathsf{CT}_{x-1}[r_{x-1}]^*, \mathsf{CT}_x[M_b]^*, \mathsf{CT}_{x+1}[r_{x+1}]^*, \cdots, \mathsf{CT}_n[r_n]^*)$ to $\mathcal{A}$.
5. When $\mathcal{A}$ outputs $\rho_{\mathcal{B},\mathcal{C}}$, $\widetilde{\mathcal{A}_x}$ sends $\{\mathsf{sk}_i\}_{i\in[n]\backslash x}$ and the $\mathcal{B}$ register (resp. the $\mathcal{C}$ register) to $\widetilde{\mathcal{B}_x}$ (resp. $\widetilde{\mathcal{C}_x}$).
6. $\widetilde{\mathcal{B}_x}$ and $\widetilde{\mathcal{C}_x}$ receive sk_x^* from the challenger of Σ_x^*.
7. $\widetilde{\mathcal{B}_x}$ (resp. $\widetilde{\mathcal{C}_x}$) sends $\{\mathsf{sk}_i^*\}_{i\in[n]}$ and the $\mathcal{B}$ register to $\mathcal{B}$ (resp. the $\mathcal{C}$ register to $\mathcal{C}$).
8. The experiment outputs 1 if $b = b_{\mathcal{B}} = b_{\mathcal{C}}$, where $b_{\mathcal{B}}$ (resp. $b_{\mathcal{C}}$) is the output of $\mathcal{B}$ (resp. $\mathcal{C}$).

From the construction of $(\widetilde{\mathcal{A}_x}, \widetilde{\mathcal{B}_x}, \widetilde{\mathcal{C}_x})$, it perfectly simulates the challenger of Σ. Therefore, if $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ breaks the one-time unclonable IND-CPA security of Σ, then $(\widetilde{\mathcal{A}_x}, \widetilde{\mathcal{B}_x}, \widetilde{\mathcal{C}_x})$ breaks the one-time unclonable IND-CPA security of Σ_x^*. □

□

7.1 Universal Constructions

Definition 7.9. *We say that a set of uniform QPT algorithms $\Sigma_{\mathsf{Univ}} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is a universal construction of (one-time) unclonable SKE (resp. PKE) if Σ_{Univ} is (one-time) unclonable SKE (resp. PKE) as long as there exists (one-time) unclonable SKE (resp. PKE).*

We give a universal construction of unclonable encryption via robust combiners.

Universal Construction via Robust Combiner

Theorem 7.10. *There exists a universal construction of (one-time) unclonable SKE and unclonable PKE.*

The proof is almost the same as Theorem 4.9, and thus we skip the proof.

8 Universal Plaintext Extension for Unclonable Encryption

In this section, we prove the following Theorem 8.1.

Theorem 8.1. *Assume that there exists a decomposable quantum randomized encoding and one-time unclonable SKE* $\Sigma_{\mathsf{unclone}} = \mathsf{Unclone}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *where the size of the quantum circuit* $\mathsf{Unclone}.\mathsf{Dec}(1^\lambda, \cdot, \cdot)$ *is* $\ell(\lambda)$*. Then, for all polynomial* n*, there exists a polynomial* p *which depends on the polynomial* n *and* ℓ *and a set of uniform QPT algorithms* $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *which depends on the polynomial* p *such that* Σ *is a one-time unclonable secret-key encryption for* $n(\lambda)$*-bit plaintexts.*

Remark 8.2. Our construction is universal construction for one-time unclonable SKE in the sense that our construction does not depend on the single-bit scheme $\Sigma_{\mathsf{unclone}}$ that is assumed to exist except for the size of the decryption circuit of $\Sigma_{\mathsf{unclone}}$.

As corollaries, we obtain Corollaries 8.3 and 8.4.

Corollary 8.3. *For all polynomial* n*, there exists a set of uniform QPT algorithms* $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that* Σ *is unclonable secret-key encryption for* $n(\lambda)$*-bit plaintexts if there exists unclonable secret-key encryption for single-bit plaintexts.*

Corollary 8.4. *For all polynomial* n*, there exists a set of uniform QPT algorithms* $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that* Σ *is unclonable public-key encryption for* $n(\lambda)$*-bit plaintexts if there exists unclonable public-key encryption for single-bit plaintexts.*

Proof of Corollary 8.4. We give a rough sketch of the proof of Corollary 8.4. Note that, in the same way, we can prove Corollary 8.3.

We can construct PKE with quantum ciphertexts and one-time unclonable SKE with single-bit plaintexts from unclonable PKE for single-bit plaintexts. We can construct decomposable quantum randomized encoding from PKE with quantum ciphertexts. Furthermore, from Theorem 8.1, we can construct one-time unclonable SKE with $n(\lambda)$-bit plaintexts from decomposable quantum randomized encoding and one-time unclonable SKE with single-bit plaintexts.

On the other hand, we can construct receiver non-committing encryption with quantum ciphertexts from PKE with quantum ciphertexts. By combining the receiver non-committing encryption with quantum ciphertexts and one-time unclonable SKE with $n(\lambda)$-bit plaintexts, we obtain unclonable PKE with $n(\lambda)$-bit plaintexts (For the detail, see Appendix E). □

Proof of Theorem 8.1. First, let us describe notations and observations.

Notations and observations.

- Let $C_{\lambda,p}[m]$ be a quantum circuit of size $p(\lambda)$ with λ-qubit quantum inputs and λ-bit classical inputs such that it outputs m for any inputs, where p is a polynomial which we specify later.
- Let $\Sigma_{\mathsf{RE}} := \mathsf{RE}.(\mathsf{Enc}, \mathsf{Dec})$ be a decomposable quantum randomized encoding. Given quantum circuit C and n_1-length quantum input and n_2-length classical input $\mathbf{q}$ and x, the encoding $\widehat{C}(\mathbf{q}, x)$ can be separated as follows:

$$\widehat{C}(\mathbf{q}, x, r, e) = (\widehat{C}_{\mathsf{off}}, \widehat{C}_1, \cdots, \widehat{C}_{n_1+n_2})(\mathbf{q}, x, r, e),$$

  where r is uniformly ransom string and e is some quantum state. From decomposability, $\widehat{C}_{\mathsf{off}}$ acts only on r and e, and $\widehat{C_i}$ acts only on $\mathbf{q_i}, r$ and e for $i \in [n_1]$, and $\widehat{C_i}$ acts only on x_i and r for $i \in \{n_1+1, \cdots, n_1+n_2\}$. For any quantum circuit C, we write $\mathsf{lab}[i, x_i] = \widehat{C_i}(x_i, r_i)$ and $\mathsf{lab}[i, \mathbf{q}_i] = \widehat{C_i}(\mathbf{q}_i, r, e)$.

Construction. We give a construction of one-time unclonable secret-key encryption $\Sigma := (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with $n(\lambda)$-bit plaintexts by using decomposable quantum randomized encoding. In the construction, we only use decomposable quantum randomized encoding. The construction is secure as long as the underlying decomposable quantum randomized encoding is secure and there exists one-time unclonable secret-key encryption for single-bit plaintexts.

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

- Sample $x \leftarrow \{0,1\}^\lambda$.
- Sample $R[i] \leftarrow \{0,1\}^{\ell(\lambda)}$ for all $i \in [\lambda]$.
- Output $\mathsf{sk} := \left(x, \{R[i]\}_{i\in[\lambda]}\right)$.

$\mathsf{Enc}(1^\lambda, \mathsf{sk}, m)$**:**

- Parse $\mathsf{sk} = \left(x, \{R[i]\}_{i\in[\lambda]}\right)$.
- Prepare the quantum circuit $C_{\lambda,p}[m]$ that outputs m for any inputs.
- Compute $\widehat{C_{\lambda,p}}[m]_{\mathsf{off}}$.
- Compute $\{\mathsf{lab}[i,0]\}_{i\in[\lambda]}$, and $\{\mathsf{lab}[i,b]\}_{i\in\{\lambda+1,\cdots,2\lambda\},b\in\{0,1\}}$.
- Sample $S[i] \leftarrow \{0,1\}^{\ell(\lambda)}$ for all $i \in [\lambda]$.
- Compute $\mathsf{Lab.CT}[i+\lambda, x[i]] := R[i]+\mathsf{lab}[i+\lambda, x[i]]$ and $\mathsf{Lab.CT}[i+\lambda, 1-x[i]] := S[i]+\mathsf{lab}[i+\lambda, 1-x[i]]$ for all $i \in [\lambda]$.
- Output

$$\mathsf{CT} := \left(\widehat{C_{\lambda,p}}[m]_{\mathsf{off}},\ \{\mathsf{lab}[i,0]\}_{i\in[\lambda]}, \{\mathsf{Lab.CT}[i,b]\}_{i\in\{\lambda+1,\cdots,2\lambda\},b\in\{0,1\}}\right).$$

$\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT})$**:**

- Parse $\mathsf{sk} = \left(x, \{R[i]\}_{i\in[\lambda]}\right)$ and

$$\mathsf{CT} = \left(\widehat{C_{\lambda,p}}[m]_{\mathsf{off}}, \{\mathsf{lab}[i,0]\}_{i\in[\lambda]}, \{\mathsf{Lab.CT}[i,b]\}_{i\in\{\lambda+1,\cdots,2\lambda\},b\in\{0,1\}}\right).$$

- Compute $\mathsf{lab}[i+\lambda, x[i]] := \mathsf{Lab.CT}[i+\lambda, x[i]] + R[i]$ for all $i \in [\lambda]$.
- Compute

$$\mathsf{RE.Dec}\left(\widehat{C_{\lambda,p}}[m]_{\mathsf{off}}, \{\mathsf{lab}[i,0]\}_{i\in[\lambda]}, \{\mathsf{lab}[i,x[i]]\}_{i\in\{\lambda+1,\cdots,2\lambda\}}\right)$$

  and outputs its output.

Lemma 8.5. *Σ satisfies efficiency if Σ_{RE} is decomposable quantum randomized encoding.*

Lemma 8.6. *Σ satisfies correctness if Σ_{RE} is decomposable quantum randomized encoding.*

Lemma 8.7. *If Σ_{RE} is decomposable quantum randomized encoding and there exists one-time unclonable secret-key encryption with single-bit plaintexts, Σ satisfies one-time IND-CPA security for some polynomial p.*

Lemma 8.8. *If Σ_{RE} is decomposable quantum randomized encoding and there exists one-time unclonable secret-key encryption with single-bit plaintexts, Σ satisfies one-time unclonable IND-CPA security for some polynomial p.*

Lemma 8.5 straightforwardly follows. We can see that Lemma 8.6 holds as follows. First, if $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{sk}, m)$, $\mathsf{Dec}(\mathsf{sk}, \mathsf{CT})$ outputs the output of $C_{\lambda,p}[m](0^\lambda, x)$. From the definition of $C_{\lambda,p}[m]$, $C_{\lambda,p}[m](0^\lambda, x)$ outputs m for all x.

The proof of Lemma 8.7 is the same as Lemma 8.8, and thus we skip the proof.

Proof of Lemma 8.8. By a standard argument, we can show the following Proposition 8.9.

Proposition 8.9. *If there exists one-time unclonable secret-key encryption for single-bit plaintexts, then there exists a one-time unclonable secret-key encryption for single-bit plaintexts scheme* $\Sigma_{\mathsf{unclone}} = \mathsf{Unclone}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *such that the following properties are satisfied:*

1. $\Sigma_{\mathsf{unclone}}$ *satisfies perfect correctness.*
2. *For all security parameters* $\lambda \in \mathbb{N}$ *and* $b \in \{0,1\}$*, we have* $|\mathsf{sk}_\lambda| = |\mathsf{CT}_{\lambda,b}| = \lambda$*, where* $\mathsf{sk}_\lambda \leftarrow \mathsf{Unclone.KeyGen}(1^\lambda)$ *and* $\mathsf{CT}_{\lambda,b} \leftarrow \mathsf{Unclone.Enc}(1^\lambda, \mathsf{sk}_\lambda, b)$.
3. *For all security parameters* λ*,* $\mathsf{Unclone.KeyGen}(1^\lambda)$ *uniformly randomly samples* sk_λ.

We give the proof of Proposition 8.9 in Appendix F. We define $D_\lambda[m_0, m_1]$ as a quantum circuit that takes as input λ-qubit quantum inputs ρ and λ-bit classical bits x, runs the quantum circuit $b \leftarrow \mathsf{Unclone.Dec}(1^\lambda, x, \rho)$, and outputs m_b. Now, we define p as a polynomial large enough to run the circuit $D_\lambda[m_0, m_1]$.

We describe the sequence of hybrids against the adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$.

Hyb_0**:** This is the original one-time unclonable IND-CPA security experiment.

1. The challenger samples $b \leftarrow \{0,1\}$.
2. The challenger samples $x \leftarrow \{0,1\}^\lambda$ and $R[i] \leftarrow \{0,1\}^{\ell(\lambda)}$ for all $i \in [\lambda]$.
3. $\mathcal{A}$ sends (m_0, m_1) to the challenger.
4. The challenger computes $\widehat{C_{\lambda,p}}[m_b]_{\mathsf{off}}$, $\{\mathsf{lab}[i,0]\}_{i\in[\lambda]}$, and $\{\mathsf{lab}[i,\beta]\}_{i\in\{\lambda+1,\cdots,2\lambda\},\beta\in\{0,1\}}$.
5. The challenger samples $S[i] \leftarrow \{0,1\}^{\ell(\lambda)}$ for all $i \in [\lambda]$, and computes
$$\mathsf{Lab.CT}[i+\lambda, x[i]] \coloneqq R[i] + \mathsf{lab}[i+\lambda, x[i]]$$
$$\mathsf{Lab.CT}[i+\lambda, 1-x[i]] \coloneqq S[i] + \mathsf{lab}[i+\lambda, 1-x[i]]$$
for all $i \in [\lambda]$.
6. The challenger sends
$$\mathsf{CT} \coloneqq \left(\widehat{C_{\lambda,p}}[m]_{\mathsf{off}}, \{\mathsf{lab}[i,0]\}_{i\in[\lambda]}, \{\mathsf{Lab.CT}[i,\beta]\}_{i\in\{\lambda+1,\cdots,2\lambda\},\beta\in\{0,1\}}\right).$$
to $\mathcal{A}$.
7. $\mathcal{A}$ produces $\rho_{\mathcal{B},\mathcal{C}}$ and sends the corresponding registers to $\mathcal{B}$ and $\mathcal{C}$.
8. $\mathcal{B}$ and $\mathcal{C}$ receives $\left(x, \{R[i]\}_{i\in[\lambda]}\right)$, and outputs $b_\mathcal{B}$ and $b_\mathcal{C}$.
9. The experiment outputs 1 if $b_\mathcal{B} = b_\mathcal{C} = b$, and otherwise 0.

Hyb_1**:**

1. The challenger samples $b \leftarrow \{0,1\}$.
2. The challenger samples $x \leftarrow \{0,1\}^\lambda$ and $R[i] \leftarrow \{0,1\}^{\ell(\lambda)}$ for all $i \in [\lambda]$.
3. The adversary $\mathcal{A}$ sends (m_0, m_1) to the challenger.
4. The challenger computes $\mathsf{unclone.CT}_b \leftarrow \mathsf{Unclone.Enc}(1^\lambda, x, b)$, where $\mathsf{unclone.CT}_b$ is the λ-length quantum states.
5. The challenger computes $\widehat{D_\lambda}[m_0, m_1]_{\mathsf{off}}$, $\{\mathsf{lab}[i, \mathsf{unclone.CT}_b[i]]\}_{i\in[\lambda]}$, and $\{\mathsf{lab}[i,\beta]\}_{i\in\{\lambda+1,\cdots,2\lambda\},\beta\in\{0,1\}}$.
6. The challenger samples $S[i] \leftarrow \{0,1\}^{\ell(\lambda)}$ for all $i \in [\lambda]$, and computes
$$\mathsf{Lab.CT}[i+\lambda, x[i]] \coloneqq R[i] + \mathsf{lab}[i+\lambda, x[i]]$$
$$\mathsf{Lab.CT}[i+\lambda, 1-x[i]] \coloneqq S[i] + \mathsf{lab}[i+\lambda, 1-x[i]]$$
for all $i \in [\lambda]$.

7. The challenger sends

$$\mathsf{CT} \coloneqq \left(\widehat{D_\lambda}[m_0, m_1]_{\mathsf{off}}, \{\mathsf{lab}[i, \mathsf{unclone}.\mathsf{CT}_b[i]]\}_{i\in[\lambda]}, \{\mathsf{Lab}.\mathsf{CT}[i,\beta]\}_{i\in\{\lambda+1,\cdots,2\lambda\},\beta\in\{0,1\}}\right)$$

to $\mathcal{A}$.
8. $\mathcal{A}$ produces $\rho_{\mathcal{B},\mathcal{C}}$ and sends the corresponding registers to $\mathcal{B}$ and $\mathcal{C}$.
9. $\mathcal{B}$ and $\mathcal{C}$ receives $\left(x, \{R[i]\}_{i\in[\lambda]}\right)$, and outputs $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$.
10. The experiment outputs 1 if $b_{\mathcal{B}} = b_{\mathcal{C}} = b$, and otherwise 0.

Lemma 8.8 follows from the following Propositions 8.10 and 8.11.

Proposition 8.10. *If Σ_{RE} is decomposable quantum randomized encoding, then*

$$|\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]| \leq \mathsf{negl}(\lambda).$$

Proposition 8.11. *If there exists a one-time unclonable secret-key encryption $\Sigma_{\mathsf{Unclone}}$ with single-bit plaintexts, then*

$$|\Pr[\mathsf{Hyb}_1 = 1]| \leq \frac{1}{2} + \mathsf{negl}(\lambda).$$

□

Proof of Proposition 8.10. Assume that there exists a QPT adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ such that

$$|\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]|$$

is non-negligible. Then, construct a QPT adversary $\widetilde{\mathcal{A}}$ that breaks the security of Σ_{RE} as follows.

1. $\widetilde{\mathcal{A}}$ samples $b \leftarrow \{0,1\}$.
2. $\widetilde{\mathcal{A}}$ samples $x \leftarrow \{0,1\}^\lambda$ and $R[i] \leftarrow \{0,1\}^{\ell(\lambda)}$ for all $i \in [\lambda]$.
3. $\widetilde{\mathcal{A}}$ receives (m_0, m_1) from the $\mathcal{A}$.
4. $\widetilde{\mathcal{A}}$ computes $\mathsf{unclone}.\mathsf{CT}_b \leftarrow \mathsf{Unclone}.\mathsf{Enc}(1^\lambda, x, b)$.
5. $\widetilde{\mathcal{A}}$ sends $\left(\{C_{\lambda,p}[m_b], 0^\lambda, x\}, \{D_\lambda[m_0, m_1], \mathsf{unclone}.\mathsf{CT}_b, x\}\right)$ to the challenger of Σ_{RE} in Proposition 3.22.
6. The challenger samples $b^* \leftarrow \{0,1\}$, and does the following.
   - If $b^* = 0$, then the challenger computes

$$\left(\widehat{C}_{\mathsf{off}}, \{\mathsf{lab}[i]\}_{i\in[2\lambda]}\right) \leftarrow \mathsf{RE}.\mathsf{Enc}\left(1^\lambda, C_{\lambda,p}[m_b], (0^\lambda, x)\right),$$

   and sends $\left(\widehat{C}_{\mathsf{off}}, \{\mathsf{lab}[i]\}_{i\in[2\lambda]}\right)$ to $\widetilde{\mathcal{A}}$.
   - If $b^* = 1$, then the challenger computes

$$\left(\widehat{C}_{\mathsf{off}}, \{\mathsf{lab}[i]\}_{i\in[2\lambda]}\right) \leftarrow \mathsf{RE}.\mathsf{Enc}\left(1^\lambda, D_{\lambda,p}[m_0, m_1], (\mathsf{unclone}.\mathsf{CT}_b, x)\right),$$

   and sends $\left(\widehat{C}_{\mathsf{off}}, \{\mathsf{lab}[i]\}_{i\in[2\lambda]}\right)$ to $\widetilde{\mathcal{A}}$.

7. $\widetilde{\mathcal{A}}$ samples $S[i] \leftarrow \{0,1\}^{\ell(\lambda)}$ for all $i \in [\lambda]$, computes

$$\begin{aligned}\mathsf{Lab.CT}[i+\lambda, x[i]] &:= R[i] + \mathsf{lab}[i+\lambda]\\ \mathsf{Lab.CT}[i+\lambda, 1-x[i]] &:= S[i]\end{aligned}$$

for all $i \in [\lambda]$, and runs $\mathcal{A}$ on

$$\mathsf{CT} := \left(\widehat{C}_{\mathsf{off}}, \{\mathsf{lab}[i]\}_{i\in[\lambda]}, \{\mathsf{Lab.CT}[i,\beta]\}_{i\in\{\lambda+1,\cdots,2\lambda\},\beta\in\{0,1\}}\right),$$

and generates $\rho_{\mathcal{B},\mathcal{C}}$.

8. $\widetilde{\mathcal{A}}$ sends the corresponding register to $\mathcal{B}$ and $\mathcal{C}$, respectively.
9. $\widetilde{\mathcal{A}}$ sends x and $\{R[i]\}_{i\in[\lambda]}$ to $\mathcal{B}$ and $\mathcal{C}$.
10. $\mathcal{B}$ and $\mathcal{C}$ outputs $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$, respectively.
11. $\widetilde{\mathcal{A}}$ outputs 1 if $b = b_{\mathcal{B}} = b_{\mathcal{C}}$, and outputs 0 otherwise.

From the construction of $\widetilde{\mathcal{A}}$, if $b^* = 0$, $\widetilde{\mathcal{A}}$ perfectly simulates the challenger of Hyb_0. Otherwise, $\widetilde{\mathcal{A}}$ perfectly simulates the challenger of Hyb_1. Furthermore, we have

$$C_{\lambda,p}[m_b](0^\lambda, x) = D_\lambda[m_0,m_1](\mathsf{unclone.CT}_b, x) = m_b,$$

and the size of $C_{\lambda,p}$ is equal to $D_\lambda[m_0,m_1]$ for an appropriate polynomial p. Therefore, if there exists a QPT adversary $(\mathcal{A},\mathcal{B},\mathcal{C})$ such that

$$|\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]|$$

is non-negligible, then it contradicts that Σ_{RE} satisfies security from Proposition 3.22. □

Proof of Proposition 8.11. Assume that there exists a QPT adversary $(\mathcal{A},\mathcal{B},\mathcal{C})$ such that $\Pr[\mathsf{Hyb}_1 = 1]$ is non-negligible. Then, construct a QPT adversary $(\widetilde{\mathcal{A}},\widetilde{\mathcal{B}},\widetilde{\mathcal{C}})$ that breaks the unclonable IND-CPA security of $\Sigma_{\mathsf{unclone}}$ as follows.

1. The challenger of $\Sigma_{\mathsf{unclone}}$ samples $b \leftarrow \{0,1\}$.
2. $\widetilde{\mathcal{A}}$ samples $R[i,\beta] \leftarrow \{0,1\}^{\ell(\lambda)}$ for all $i \in [\lambda]$ and $\beta \in \{0,1\}$.
3. $\widetilde{\mathcal{A}}$ receives (m_0,m_1) from the $\mathcal{A}$.
4. $\widetilde{\mathcal{A}}$ sends $(0,1)$ to the challenger, and receives $\mathsf{unclone.CT}_b$, where $\mathsf{unclone.CT}_b \leftarrow \mathsf{Unclone.Enc}(1^\lambda, x, b)$ and $x \leftarrow \{0,1\}^\lambda$.
5. $\widetilde{\mathcal{A}}$ computes $\widehat{D_\lambda}[m_0,m_1]_{\mathsf{off}}$, $\{\mathsf{lab}[i, \mathsf{unclone.CT}_b[i]]\}_{i\in[\lambda]}$, and $\{\mathsf{lab}[i,\beta]\}_{i\in\{\lambda+1,\cdots,2\lambda\},\beta\in\{0,1\}}$.
6. $\widetilde{\mathcal{A}}$ computes $\mathsf{Lab.CT}[i+\lambda,\beta] := R[i,\beta] + \mathsf{lab}[i+\lambda,\beta]$ for all $i \in [\lambda]$ and $\beta \in \{0,1\}$.
7. $\widetilde{\mathcal{A}}$ runs $\mathcal{A}$ on

$$\left(\widehat{D_\lambda}[m_0,m_1]_{\mathsf{off}}, \{\mathsf{lab}[i, \mathsf{unclone.CT}_b[i]]\}_{i\in[\lambda]}, \{\mathsf{Lab.CT}[i,\beta]\}_{i\in\{\lambda+1,\cdots,2\lambda\},\beta\in\{0,1\}}\right),$$

obtains $\rho_{\mathcal{B},\mathcal{C}}$, and sends the $\mathcal{B}$ register and $\{R[i,\beta]\}_{i\in[\lambda],\beta\in\{0,1\}}$ to $\mathcal{B}$ and the $\mathcal{C}$ register and $\{R[i,\beta]\}_{i\in[\lambda],\beta\in\{0,1\}}$ to $\mathcal{C}$.
8. $\widetilde{\mathcal{B}}$ (resp. $\widetilde{\mathcal{C}}$) receives the secret-key x from the challenger of $\Sigma_{\mathsf{unclone}}$ and sends $(x, \{R[i,x[i]]\}_{i\in[\lambda]})$ and the $\mathcal{B}$ register (resp. $\mathcal{C}$ register) to $\mathcal{B}$ (resp. $\mathcal{C}$).

9. The experiment outputs 1 if $b = b_{\mathcal{B}} = b_{\mathcal{C}}$ where $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$ are the outputs of $\mathcal{B}$ and $\mathcal{C}$, respectively.

From the construction of $(\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}}, \widetilde{\mathcal{C}})$, it perfectly simulates the challenger of Hyb_1. Therefore, if there exists some QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ such that $\Pr[\mathsf{Hyb}_1 = 1]$ is non-negligible, it contradicts that $\Sigma_{\mathsf{unclone}}$ satisfies unclonable IND-CPA security. □

□

Acknowledgements. TH is supported by JSPS research fellowship and by JSPS KAKENHI No. JP22J21864.

References

[Aar18] Scott Aaronson. Shadow tomography of quantum states. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 325–338. ACM Press, June 2018. (Cited on page 1.)

[AAS20] Scott Aaronson, Yosi Atia, and Leonard Susskind. On the hardness of detecting macroscopic superpositions. *Electron. Colloquium Comput. Complex.*, TR20-146, 2020. (Cited on page 44.)

[ABJ+19] Prabhanjan Ananth, Saikrishna Badrinarayanan, Aayush Jain, Nathan Manohar, and Amit Sahai. From FE combiners to secure MPC and back. In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part I*, volume 11891 of *LNCS*, pages 199–228. Springer, Heidelberg, December 2019. (Cited on page 1.)

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In *STOC*, pages 41–60. ACM, 2012. (Cited on page 1, 2, 13.)

[AGQY22] Prabhanjan Ananth, Aditya Gulati, Luowen Qian, and Henry Yuen. Pseudorandom (function-like) quantum state generators: New definitions and applications. In Eike Kiltz and Vinod Vaikuntanathan, editors, *Theory of Cryptography*, pages 237–265, Cham, 2022. Springer Nature Switzerland. (Cited on page 1, 3.)

[AJN+16] Prabhanjan Ananth, Aayush Jain, Moni Naor, Amit Sahai, and Eylon Yogev. Universal constructions and robust combiners for indistinguishability obfuscation and witness encryption. In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part II*, volume 9815 of *LNCS*, pages 491–520. Springer, Heidelberg, August 2016. (Cited on page 1.)

[AJS17] Prabhanjan Ananth, Aayush Jain, and Amit Sahai. Robust transforming combiners from indistinguishability obfuscation to functional encryption. In *EUROCRYPT (1)*, pages 91–121. Springer, 2017. (Cited on page 1.)

[AK21] Prabhanjan Ananth and Fatih Kaleoglu. Unclonable encryption, revisited. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography*, pages 299–329, Cham, 2021. Springer International Publishing. (Cited on page 4, 6, 15, 31, 46.)

[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, *Advances in Cryptology – CRYPTO 2022*, pages 212–241, Cham, 2022. Springer Nature Switzerland. (Cited on page 3, 4, 10.)

[AQY22] Prabhanjan Ananth, Luowen Qian, and Henry Yuen. Cryptography from pseudorandom quantum states. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 208–236, Cham, 2022. Springer Nature Switzerland. (Cited on page 1, 2, 3.)

[BB84] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, page 175, India, 1984. (Cited on page 3.)

[BCKM21] James Bartusek, Andrea Coladangelo, Dakshita Khurana, and Fermi Ma. On the round complexity of secure quantum computation. In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part I*, volume 12825 of *LNCS*, pages 406–435, Virtual Event, August 2021. Springer, Heidelberg. (Cited on page 3.)

[BCQ23] Zvika Brakerski, Ran Canetti, and Luowen Qian. On the computational hardness needed for quantum cryptography. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 24:1–24:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. (Cited on page 1, 2, 3.)

[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 4, 14.)

[BY22] Zvika Brakerski and Henry Yuen. Quantum garbled circuits. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 804–817, New York, NY, USA, 2022. Association for Computing Machinery. (Cited on page 3, 10, 16, 17.)

[CHK05] Ran Canetti, Shai Halevi, and Jonathan Katz. Adaptively-secure, non-interactive public-key encryption. In Joe Kilian, editor, *TCC 2005*, volume 3378 of *LNCS*, pages 150–168. Springer, Heidelberg, February 2005. (Cited on page 31.)

[CLS01] Claude Crépeau, Frédéric Légaré, and Louis Salvail. How to convert the flavor of a quantum bit commitment. In Birgit Pfitzmann, editor, *EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 60–77. Springer, Heidelberg, May 2001. (Cited on page 3.)

[CX22] Shujiao Cao and Rui Xue. On constructing one-way quantum state generators, and more. Cryptology ePrint Archive, Paper 2022/1323, 2022. https://eprint.iacr.org/2022/1323. (Cited on page 2.)

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976. (Cited on page 1.)

[DMS00] Paul Dumais, Dominic Mayers, and Louis Salvail. Perfectly concealing quantum bit commitment from any quantum one-way permutation. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 300–315. Springer, Heidelberg, May 2000. (Cited on page 3.)

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31:469–472, 1985. (Cited on page 1.)

[FGH+12] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter W. Shor. Quantum money from knots. In Shafi Goldwasser, editor, *ITCS 2012*, pages 276–289. ACM, January 2012. (Cited on page 1, 2.)

[GLSV21] Alex B. Grilo, Huijia Lin, Fang Song, and Vinod Vaikuntanathan. Oblivious transfer is in MiniQCrypt. In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 531–561. Springer, Heidelberg, October 2021. (Cited on page 3.)

[GTK16] Shafi Goldwasser and Yael Tauman Kalai. Cryptographic assumptions: A position paper. In Eyal Kushilevitz and Tal Malkin, editors, *Theory of Cryptography*, pages 505–522, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg. (Cited on page 1.)

[Her05] Amir Herzberg. On tolerant cryptographic constructions. In Alfred Menezes, editor, *Topics in Cryptology – CT-RSA 2005*, pages 172–190, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. (Cited on page 1.)

[HKM+23] Taiga Hiroka, Fuyuki Kitagawa, Tomoyuki Morimae, Ryo Nishimaki, Tapas Pal, and Takashi Yamakawa. Certified everlasting secure collusion-resistant functional encryption, and more. Cryptology ePrint Archive, Paper 2023/236, 2023. https://eprint.iacr.org/2023/236. (Cited on page 46.)

[HKN+05] Danny Harnik, Joe Kilian, Moni Naor, Omer Reingold, and Alon Rosen. On robust combiners for oblivious transfer and other primitives. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 96–113. Springer, Heidelberg, May 2005. (Cited on page 1, 4, 5, 6, 9, 17, 31.)

[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, *Advances in Cryptology – ASIACRYPT 2021*, pages 606–636, Cham, 2021. Springer International Publishing. (Cited on page 4, 6, 31, 46.)

[HMY23] Minki Hhan, Tomoyuki Morimae, and Takashi Yamakawa. From the hardness of detecting superpositions to cryptography: Quantum public key encryption and commitments. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 639–667, Cham, 2023. Springer Nature Switzerland. (Cited on page 3, 7, 8, 9, 14, 44.)

[JLS18] Zhengfeng Ji, Yi-Kai Liu, and Fang Song. Pseudorandom quantum states. In Hovav Shacham and Alexandra Boldyreva, editors, *CRYPTO 2018, Part III*, volume 10993 of *LNCS*, pages 126–152. Springer, Heidelberg, August 2018. (Cited on page 1, 3.)

[JMS20] Aayush Jain, Nathan Manohar, and Amit Sahai. Combiners for functional encryption, unconditionally. In Anne Canteaut and Yuval Ishai, editors, *EUROCRYPT 2020, Part I*, volume 12105 of *LNCS*, pages 141–168. Springer, Heidelberg, May 2020. (Cited on page 1.)

[Kan18] Daniel M. Kane. Quantum money from modular forms. *arXiv:1809.05925*, 2018. (Cited on page 1, 2.)

[KNTY19] Fuyuki Kitagawa, Ryo Nishimaki, Keisuke Tanaka, and Takashi Yamakawa. Adaptively secure and succinct functional encryption: Improving security and efficiency, simultaneously. In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 521–551. Springer, Heidelberg, August 2019. (Cited on page 31, 46.)

[KQST23] William Kretschmer, Luowen Qian, Makrand Sinha, and Avishay Tal. Quantum cryptography in algorithmica. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 1589–1602, New York, NY, USA, 2023. Association for Computing Machinery. (Cited on page 2.)

[Kre21] William Kretschmer. Quantum Pseudorandomness and Classical Complexity. In Min-Hsiu Hsieh, editor, *16th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2021)*, volume 197 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. (Cited on page 1.)

[KSS22] Daniel M. Kane, Shahed Sharif, and Alice Silverberg. Quantum money from quaternion algebras, 2022. (Cited on page 1, 2.)

[KT23] Dakshita Khurana and Kabir Tomer. Commitments from quantum one-wayness, 2023. (Cited on page 3.)

[LC97] Hoi-Kwong Lo and H. F. Chau. Is quantum bit commitment really possible? *Phys. Rev. Lett.*, 78:3410–3413, 1997. (Cited on page 1, 3.)

[Lev85] Leonid A. Levin. One-way functions and pseudorandom generators. In *17th ACM STOC*, pages 363–365. ACM Press, May 1985. (Cited on page 1.)

[LMZ23] Jiahui Liu, Hart Montgomery, and Mark Zhandry. Another round of breaking and making quantum money:. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, pages 611–638, Cham, 2023. Springer Nature Switzerland. (Cited on page 1, 2.)

[May97] Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Phys. Rev. Lett.*, 78:3414–3417, 1997. (Cited on page 1, 3.)

[MY22a] Tomoyuki Morimae and Takashi Yamakawa. One-wayness in quantum cryptography. Cryptology ePrint Archive, Paper 2022/1336, 2022. https://eprint.iacr.org/2022/1336. (Cited on page 1, 2, 3, 12.)

[MY22b] Tomoyuki Morimae and Takashi Yamakawa. Quantum commitments and signatures without one-way functions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022*, pages 269–295, Cham, 2022. Springer Nature Switzerland. (Cited on page 1, 2, 3.)

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005. (Cited on page 1.)

[WW23] Brent Waters and Daniel Wichs. Universal amplification of kdm security: From 1-key circular to multi-key kdm. Cryptology ePrint Archive, Paper 2023/1058, 2023. https://eprint.iacr.org/2023/1058. (Cited on page 10.)

[Yan22] Jun Yan. General properties of quantum bit commitments (extended abstract). In Shweta Agrawal and Dongdai Lin, editors, *Advances in Cryptology – ASIACRYPT 2022*, pages 628–657, Cham, 2022. Springer Nature Switzerland. (Cited on page 2, 3, 7, 13.)

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 408–438. Springer, Heidelberg, May 2019. (Cited on page 1, 2.)

[Zha23a] Mark Zhandry. Quantum minimalism (talk). https://www.youtube.com/watch?v=7cqnrASfjco&ab_channel=SimonsInstitute, 2023. (Cited on page 2.)

[Zha23b] Mark Zhandry. Quantum money from abelian group actions. *IACR Cryptol. ePrint Arch.*, 2023:1097, 2023. (Cited on page 1, 2.)

A Proof of Proposition 4.12

Assume that there exists an OWSG. Then, there exists a set of classical Turing machines $\mathcal{M} := (x, y, z)$ such that $\Sigma[\mathcal{M}] := (\mathsf{KeyGen}[x], \mathsf{StateGen}[y], \mathsf{Vrfy}[z])$ satisfies correctness and security because OWSG is a set of uniform QPT algorithms. Let c_x, and c_y, and c_z be a constant such that x, y, and z halts within λ^{c_x}, λ^{c_y}, and λ^{c_z} steps for all sufficiently large $\lambda \in \mathbb{N}$, respectively. For simplicity, let us assume that $c_x \geq c_y \geq c_z$. Note that the same argument goes through in the other cases.

For the set of uniform QPT algorithms $\Sigma[\mathcal{M}] = (\mathsf{KeyGen}[x], \mathsf{StateGen}[y], \mathsf{Vrfy}[z])$, $\Sigma[\mathcal{M}^*] := (\mathsf{KeyGen}[x^*], \mathsf{StateGen}[y^*], \mathsf{Vrfy}[z^*])$ is the set of uniform algorithms working as follows:

$\mathsf{KeyGen}[x^*](1^\lambda)$**:**

- It runs a classical Turing machine x on 1^κ and obtain a general quantum circuit $C[x]_\kappa$, where the $\kappa \in \mathbb{N}$ is the largest integer such that $\kappa + \kappa^{c_x} \leq \lambda$.
- Output k, which is the output of $C[x]_\kappa$.

$\mathsf{StateGen}[y^*](1^\lambda, k)$**:**

- It runs a classical Turing machine y on $(1^\kappa, k)$ and obtain a general quantum circuit $C[y]_{\kappa,k}$, where the $\kappa \in \mathbb{N}$ is the largest integer such that $\kappa + \kappa^{c_x} \leq \lambda$.
- Output ψ_k, which is the output of $C[y]_{\kappa,k}$.

$\mathsf{Vrfy}[z^*](1^\lambda, k, \psi_k)$**:**

- It runs a classical Turing machine z on $(1^\kappa, k, |\psi_k|)$ and obtain a general quantum circuit $C[z]_{\kappa,k,|\psi_k|}$, where the $\kappa \in \mathbb{N}$ is the largest integer such that $\kappa + \kappa^{c_x} \leq \lambda$.
- Output $\top$ if $1 \leftarrow C[z]_{\kappa,k,|\psi_k|}(\psi_k)$, and output $\bot$ if $0 \leftarrow C[z]_{\kappa,k,|\psi_k|}(\psi_k)$.

We can see that x^*, y^*, and z^* halts within λ^3 steps for all sufficiently large $\lambda \in \mathbb{N}$. Given 1^λ, x^* first computes κ within $O(\lambda^2)$ steps. Furthermore, $x(1^\kappa)$ halts within $\kappa^{c_x} \leq \lambda$ steps. Overall, x^* halts within $O(\lambda^2)$ steps. For the same reason, y^* and z^* also halts within $O(\lambda^2)$ steps. Therefore, for all sufficiently large security parameters $\lambda \in \mathbb{N}$, $x^*(1^\lambda)$, $y^*(1^\lambda)$, and $z^*(1^\lambda)$ halt within λ^3 steps. Apparently, $\Sigma[\mathcal{M}^*]$ satisfies correctness if $\Sigma[\mathcal{M}]$ satisfies correctness.

Furthermore, by a standard hybrid argument, we can show that the construction satisfies security as follows. Suppose that $\Sigma[\mathcal{M}^*]$ does not satisfy security and show that $\Sigma[\mathcal{M}]$ does not satisfy security. Since we assume that $\Sigma[\mathcal{M}^*]$ does not satisfy security, there exists a polynomial t, a constant C and a QPT adversary $\mathcal{A}$ such that

$$\Pr\left[\top \leftarrow \mathsf{Vrfy}[z^*](1^{\lambda+\lambda^{c_x}}, k^*, \psi_k) : \begin{array}{l} k \leftarrow \mathsf{KeyGen}[x^*](1^{\lambda+\lambda^{c_x}}) \\ \psi_k \leftarrow \mathsf{StateGen}[y^*](1^{\lambda+\lambda^{c_x}}) \\ k^* \leftarrow \mathcal{A}(\psi_k^{\otimes t(\lambda+\lambda^{c_x})}) \end{array}\right] \geq 1/(\lambda+\lambda^{c_x})^C$$

for infinitely many security parameters λ. Let t' be a polynomial such that $t'(\lambda) \geq t(\lambda+\lambda^{c_x})$ for all $\lambda \in \mathbb{N}$. Now, we construct a QPT adversary $\mathcal{B}$ that breaks $\Sigma[\mathcal{M}]$ as follows.

1. $\mathcal{B}$ first receives $\psi_k^{\otimes t'(\lambda)}$, where $k \leftarrow \mathsf{KeyGen}[x](1^\lambda)$ and $\psi_k \leftarrow \mathsf{StateGen}[y](1^\lambda, k)$.
2. $\mathcal{B}$ runs $k^* \leftarrow \mathcal{A}(\psi_k^{\otimes t(\lambda+\lambda^{c_x})})$.
3. $\mathcal{B}$ outputs k^*.

From the construction of $(\mathsf{KeyGen}[x^*], \mathsf{StateGen}[y^*], \mathsf{Vrfy}[z^*])$, $(\mathsf{KeyGen}[x^*](1^{\lambda+\lambda^{c_x}}), \mathsf{StateGen}[y^*](1^{\lambda+\lambda^{c_x}}, k), \mathsf{Vrfy}[z^*](1^{\lambda+\lambda^{c_x}}, k, \psi_k))$ works in the same way as $(\mathsf{KeyGen}[x](1^\lambda), \mathsf{StateGen}[y](1^\lambda, k), \mathsf{Vrfy}[z](1^\lambda, k, \psi_k))$. Therefore, there exists some constant D such that

$$\begin{aligned}
&\Pr\left[\top \leftarrow \mathsf{Vrfy}[z](1^\lambda, k^*, \psi_k) : \begin{array}{l} k \leftarrow \mathsf{KeyGen}[x](1^\lambda) \\ \psi_k \leftarrow \mathsf{StateGen}[y](1^\lambda, k) \\ k^* \leftarrow \mathcal{B}(\psi_k^{\otimes t'(\lambda)}) \end{array}\right] \\
&= \Pr\left[\top \leftarrow \mathsf{Vrfy}[z^*](1^{\lambda+\lambda^{c_x}}, k^*, \psi_k) : \begin{array}{l} k \leftarrow \mathsf{KeyGen}[x^*](1^{\lambda+\lambda^{c_x}}) \\ \psi_k \leftarrow \mathsf{StateGen}[y^*](1^{\lambda+\lambda^{c_x}}, k) \\ k^* \leftarrow \mathcal{A}(\psi_k^{\otimes t(\lambda+\lambda^{c_x})}) \end{array}\right] \geq 1/(\lambda+\lambda^{c_x})^C \geq 1/\lambda^D
\end{aligned}$$

for infinitely many λ. This contradicts that $\Sigma[\mathcal{M}]$ satisfies security. Therefore, $\Sigma[\mathcal{M}^*]$ satisfies security.

Therefore, $\Sigma[\mathcal{M}^*] = (\mathsf{KeyGen}[x^*], \mathsf{StateGen}[y^*], \mathsf{Vrfy}[z^*])$ is a OWSG scheme, where x^*, y^*, and z^* halts within λ^3 steps.

B Proof of Lemma 5.3

Without loss of generality, $\mathsf{Vrfy}(1^\lambda, s, \rho)$ can be considered as the algorithm working in the following way.

For input $(1^\lambda, s, \rho)$, run a classical Turing machine $\mathcal{M}$ on $(1^\lambda, s, |\rho|)$, obtain $U_{\mathsf{Vrfy},k}$, append auxiliary state $|0\cdots0\rangle\langle0\cdots0|$ to ρ, apply a unitary $U_{\mathsf{Vrfy},s}$ on $\psi \otimes |0\cdots0\rangle\langle0\cdots0|$, obtain ρ_ψ, and measure the first qubit of ρ_ψ with the computational basis and output $\top$ if the measurement result is 1 and $\bot$ otherwise.

We describe the $\Sigma^* := (\mathsf{Mint}^*, \mathsf{Vrfy}^*)$.

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

- Run $(s, \rho_s) \leftarrow \mathsf{Mint}(1^\lambda)$.
- Apply $U_{\mathsf{Vrfy},s}$ on $\rho_s \otimes |0\cdots0\rangle\langle0\cdots0|$, obtain $\rho_k := U_{\mathsf{Vrfy},s}(\rho_s \otimes |0\cdots0\rangle\langle0\cdots0|)U_{\mathsf{Vrfy},s}^\dagger$, measures the first bit of ρ_s in the computational basis, and obtains the measurement result b, and post-measurement quantum state $\rho_{b,s}$.
  - If the measurement result is 1, then output $s^* := s$ and $\rho_s^* := U_{\mathsf{Vrfy},s}^\dagger(|1\rangle\langle1| \otimes \rho_{1,s})U_{\mathsf{Vrfy},s}$.
  - If the measurement result is 0, then output $s^* := \bot$ and $\rho_s^* := U_{\mathsf{Vrfy},s}^\dagger(|0\rangle\langle0| \otimes \rho_{0,s})U_{\mathsf{Vrfy},s}$.

$\mathsf{Vrfy}^*(1^\lambda, s^*, \rho)$**:**

- If $s^* = \bot$, then output $\top$.
- If $s^* := s \neq \bot$, then measure the first qubit of $U_{\mathsf{Vrfy},s}\rho U_{\mathsf{Vrfy},s}^\dagger$ in the computational basis. Output $\top$ if the measurement result is 1, and output $\bot$ otherwise.

C Proof of Lemma 3.12

Proof of Lemma 3.12. We prove that if the commitment $\{Q_0(\lambda), Q_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies c-X hiding, then $\{\widetilde{Q}_0(\lambda), \widetilde{Q}_1(\lambda)\}_{\lambda\in\mathbb{N}}$ satisfies $\sqrt{c}$-X binding, where $X \in \{\text{computational, statistical}\}$. Because the same argument goes through, we consider the case where $X =$ statistical. Below, we fix a security parameter, and write (Q_0, Q_1) and $(\widetilde{Q_0}, \widetilde{Q_1})$ to mean $(Q_0(\lambda), Q_1(\lambda))$ and $(\widetilde{Q_0}(\lambda), \widetilde{Q_1}(\lambda))$, respectively.

First, let us introduce the following Theorem C.1.

Theorem C.1 (Equivalence between swapping and distinguishing [AAS20, HMY23]). *Let $|x_i\rangle$, $|y_i\rangle$ be orthogonal n-qubit states and $|\tau_i\rangle$ be an m-qubit state. Let U be a polynomial-time computable unitary over $(n+m)$-qubit states and define Γ as*

$$\Gamma := \left\|(\langle y|\otimes I^{\otimes m})U|x\rangle|\tau\rangle + (\langle x|\otimes I^{\otimes m})U|y\rangle|\tau\rangle\right\|_1.$$

Then, there exists a non-uniform QPT distinguisher $\mathcal{A}$ with advice $|\tau'\rangle = |\tau\rangle\otimes\frac{|x\rangle|0\rangle+|y\rangle|1\rangle}{\sqrt{2}}$ that distinguishes $|\psi\rangle = \frac{|x\rangle+|y\rangle}{\sqrt{2}}$ and $|\phi\rangle = \frac{|x\rangle-|y\rangle}{\sqrt{2}}$ with advantage $\frac{\Gamma^2}{4}$. Moreover, if U does not act on some qubits, then $\mathcal{A}$ also does not act on those qubits.

Let us assume that $\{\widetilde{Q_0}(\lambda), \widetilde{Q_1}(\lambda)\}_{\lambda\in\mathbb{N}}$ is not $\sqrt{c}$-statistical biding, and let d be some constant that satisfies $d \geq \sqrt{c}$. Then, there exists a unitary $U_{\widetilde{\mathbf{RZ}}}$ over $\widetilde{\mathbf{R}} = \mathbf{C}$ and an ancillary register $\mathbf{Z}$ and a state $|\tau\rangle_{\mathbf{Z}}$ such that

$$\left\|((\langle 0|\widetilde{Q_1}^\dagger)_{\widetilde{\mathbf{CR}}}\otimes I_{\mathbf{Z}})(I_{\widetilde{\mathbf{C}}}\otimes U_{\widetilde{\mathbf{RZ}}})((\widetilde{Q_0}|0\rangle)_{\widetilde{\mathbf{CR}}}|\tau\rangle_{\mathbf{Z}})\right\|_1 \geq d.$$

We observe that U does not act on $\mathbf{D}$, and thus it cannot cause any interference between states that take 0 and 1 in $\mathbf{D}$. Therefore, we have

$$\begin{aligned}&((\langle 0|\widetilde{Q}_1^\dagger)_{\widetilde{\mathbf{CR}}}\otimes I_{\mathbf{Z}})(I_{\widetilde{\mathbf{C}}}\otimes U_{\widetilde{\mathbf{RZ}}})(\widetilde{Q}_0|0\rangle_{\widetilde{\mathbf{CR}}}|\tau\rangle_{\mathbf{Z}})\\ &= \frac{1}{2}\begin{pmatrix}((\langle 0|Q_1^\dagger)_{\mathbf{CR}}\langle 0|_{\mathbf{D}}\otimes I_{\mathbf{Z}})(I_{\mathbf{R},\mathbf{D}}\otimes U_{\mathbf{C},\mathbf{Z}})(Q_0|0\rangle_{\mathbf{CR}}|0\rangle_{\mathbf{D}}|\tau\rangle_{\mathbf{Z}})\\ -((\langle 0|Q_1^\dagger)_{\mathbf{CR}}\langle 0|_{\mathbf{D}}\otimes I_{\mathbf{Z}})(I_{\mathbf{R},\mathbf{D}}\otimes U_{\mathbf{C},\mathbf{Z}})(Q_0|0\rangle_{\mathbf{CR}}|0\rangle_{\mathbf{D}}|\tau\rangle_{\mathbf{Z}})\end{pmatrix}.\end{aligned}$$

Similarly, we have

$$\begin{aligned}&((\langle 0|\widetilde{Q}_0^\dagger)_{\widetilde{\mathbf{CR}}}\otimes I_{\mathbf{Z}})(I_{\widetilde{\mathbf{C}}}\otimes U_{\widetilde{\mathbf{RZ}}})(\widetilde{Q}_1|0\rangle_{\widetilde{\mathbf{CR}}}|\tau\rangle_{\mathbf{Z}})\\ &= \frac{1}{2}\begin{pmatrix}((\langle 0|Q_1^\dagger)_{\mathbf{CR}}\langle 0|_{\mathbf{D}}\otimes I_{\mathbf{Z}})(I_{\mathbf{R},\mathbf{D}}\otimes U_{\mathbf{C},\mathbf{Z}})(Q_0|0\rangle_{\mathbf{CR}}|0\rangle_{\mathbf{D}}|\tau\rangle_{\mathbf{Z}})\\ -((\langle 0|Q_1^\dagger)_{\mathbf{CR}}\langle 0|_{\mathbf{D}}\otimes I_{\mathbf{Z}})(I_{\mathbf{R},\mathbf{D}}\otimes U_{\mathbf{C},\mathbf{Z}})(Q_0|0\rangle_{\mathbf{CR}}|0\rangle_{\mathbf{D}}|\tau\rangle_{\mathbf{Z}})\end{pmatrix}.\end{aligned}$$

In particular, we have

$$((\langle 0|\widetilde{Q}_1^\dagger)_{\widetilde{\mathbf{CR}}}\otimes I_{\mathbf{Z}})(I_{\widetilde{\mathbf{C}}}\otimes U_{\widetilde{\mathbf{RZ}}})(\widetilde{Q}_0|0\rangle_{\widetilde{\mathbf{CR}}}|\tau\rangle_{\mathbf{Z}}) = ((\langle 0|\widetilde{Q}_0^\dagger)_{\widetilde{\mathbf{CR}}}\otimes I_{\mathbf{Z}})(I_{\widetilde{\mathbf{C}}}\otimes U_{\widetilde{\mathbf{RZ}}})(\widetilde{Q}_1|0\rangle_{\widetilde{\mathbf{CR}}}|\tau\rangle_{\mathbf{Z}}).$$

This implies that

$$\left\|\ ((\langle 0|\widetilde{Q}_1^\dagger)_{\widetilde{\mathbf{CR}}}\otimes I_{\mathbf{Z}})(I_{\widetilde{\mathbf{C}}}\otimes U_{\widetilde{\mathbf{RZ}}})(\widetilde{Q}_0|0\rangle_{\widetilde{\mathbf{CR}}}|\tau\rangle_{\mathbf{Z}}) + ((\langle 0|\widetilde{Q}_0^\dagger)_{\widetilde{\mathbf{CR}}}\otimes I_{\mathbf{Z}})(I_{\widetilde{\mathbf{C}}}\otimes U_{\widetilde{\mathbf{RZ}}})(\widetilde{Q}_1|0\rangle_{\widetilde{\mathbf{CR}}}|\tau\rangle_{\mathbf{Z}})\ \right\|_1 \geq 2d.$$

If we set $|x\rangle := \widetilde{Q_0}\,|0\rangle_{\widetilde{\mathbf{CR}}}$ and $|y\rangle := \widetilde{Q_1}\,|0\rangle_{\widetilde{\mathbf{CR}}}$, then $|x\rangle$ and $|y\rangle$ are orthogonal. Then, by Theorem C.1, there exists a non-uniform distinguisher $\mathcal{A}$ with a polynomial-size advice $|\tau'\rangle$ that does not act on $\widetilde{\mathbf{C}} = (\mathbf{R}, \mathbf{D})$ and distinguishes

$$|\psi\rangle = \frac{|x\rangle + |y\rangle}{\sqrt{2}} = (Q_0\,|0\rangle_{\mathbf{CR}})\,|0\rangle_{\mathbf{D}}$$

and

$$|\phi\rangle = \frac{|x\rangle - |y\rangle}{\sqrt{2}} = (Q_1\,|0\rangle_{\mathbf{CR}})\,|1\rangle_{\mathbf{D}}$$

with $d^2 \geq c$. This contradicts that (Q_0, Q_1) satisfies c-statistical hiding, and thus $(\widetilde{Q_0}, \widetilde{Q_1})$ satisfies $\sqrt{c}$-statistical binding. □

D Proof of Lemma 7.4

We give the proof of Lemma 7.4.

Proof of Lemma 7.4. For a candidate of one-time unclonable secret-key encryption $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with $n(\lambda)$-plaintext space, we can assume that $\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT})$ works as follows without loss of generality:

For input $(1^\lambda, \mathsf{sk}, \mathsf{CT})$, run a classical Turing machine $\mathcal{M}$ on $(1^\lambda, \mathsf{sk}, |\mathsf{CT}|)$, obtain a unitary $U_{\mathsf{Dec},\mathsf{sk}}$, append auxiliary state $|0\cdots0\rangle\langle0\cdots0|$ to CT, apply a unitary $U_{\mathsf{Dec},\mathsf{sk}}$ on $\mathsf{CT} \otimes |0\cdots0\rangle\langle0\cdots0|$, obtain ρ_{CT}, and measure the first $n(\lambda)$ qubit of ρ_{CT} with the computational basis and output its output.

Construction of one-time unclonable secret key encryption: We give a construction $\Sigma^* = (\mathsf{KeyGen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$.

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

- Run $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda)$.
- Output $\mathsf{sk}^* := \mathsf{sk}$.

$\mathsf{Enc}^*(1^\lambda, \mathsf{sk}^*, m)$**:**

- Parse $\mathsf{sk}^* = \mathsf{sk}$.
- Run $\mathsf{CT} \leftarrow \mathsf{Enc}(1^\lambda, \mathsf{sk}, m)$.
- Measure the first $n(\lambda)$-bit of $U_{\mathsf{Dec},\mathsf{sk}}\,(\mathsf{CT} \otimes |0\cdots0\rangle\langle0\cdots0|)\,U_{\mathsf{Dec},\mathsf{sk}}^\dagger$ in the computational basis, and obtains m^* and post-measurement quantum state $\rho_{m^*,\mathsf{sk}}$.
  - If $m = m^*$, then output $\mathsf{CT}^* := U_{\mathsf{Dec},\mathsf{sk}}^\dagger\,(m \otimes \rho_{m,\mathsf{sk}})\,U_{\mathsf{Dec},\mathsf{sk}} \otimes |1\rangle\langle1|$.
  - If $m \neq m^*$, output $\mathsf{CT}^* := m \otimes |0\rangle\langle0|$.

$\mathsf{Dec}^*(1^\lambda, \mathsf{sk}^*, \mathsf{CT}^*)$**:**

- Parse $\mathsf{CT}^* = \rho \otimes |b\rangle\langle b|$ and $\mathsf{sk}^* = \mathsf{sk}$.
- Measure the final bit of CT^* with $\{|1\rangle\langle1|, |0\rangle\langle0|\}$.
  - If the result is 1, then measure the first $n(\lambda)$-bit of $U_{\mathsf{Dec},\mathsf{sk}}\rho U_{\mathsf{Dec},\mathsf{sk}}^\dagger$ in the computational basis, and outputs its output.
  - If the result is 0, then measure the first $n(\lambda)$-qubit of CT in the computational basis and outputs its output.

□

E Unclonable PKE from One-Time Unclonable SKE and PKE with Quantum Ciphertexts

It was shown that unclonable PKE can be constructed from one-time unclonable SKE and PKE with "classical" ciphertexts [AK21]. However, it is unclear whether we can construct unclonable PKE from one-time unclonable SKE and PKE with "quantum" ciphertexts based on their technique. This is because their security proof relies on the existence of OWFs, but it is an open problem whether PKE with quantum ciphertexts implies OWFs or not. Therefore, for the reader's convenience, we construct unclonable PKE from one-time unclonable SKE and PKE with quantum ciphertexts.

Our construction is based on the technique of [HMNY21]. First, let us introduce receiver non-committing encryption with quantum ciphertexts. Note that in the same way as [KNTY19, HKM+23], we can construct receiver non-committing encryption with quantum ciphertexts from PKE with quantum ciphertexts.

Definition E.1 (Receiver Non-Committing Encryption with Quantum Ciphertexts.). *An receiver non-committing encryption is a set of algorithms* $\Sigma := (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$ *such that:*

$\mathsf{Setup}(1^\lambda)$*: It takes* 1^λ*, and outputs a classical key pair* $(\mathsf{pk}, \mathsf{MSK})$.

$\mathsf{KeyGen}(1^\lambda, \mathsf{MSK})$: *It takes* 1^λ *and* MSK*, and outputs a classical key* sk.

$\mathsf{Enc}(1^\lambda, \mathsf{pk}, m)$*: It takes* 1^λ*,* pk *and* m*, and outputs a quantum ciphertext* CT.

$\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT})$*: It takes* 1^λ*,* sk *and* CT*, and outputs* m.

$\mathsf{Fake}(1^\lambda, \mathsf{pk})$*: It takes* 1^λ *and* pk*, and outputs a fake quantum ciphertext* $\widetilde{\mathsf{CT}}$ *and an auxiliary state* aux.

$\mathsf{Reveal}(1^\lambda, \mathsf{pk}, \mathsf{MSK}, \mathsf{aux}, m)$*: It takes* 1^λ*,* pk*,* MSK*,* aux*, and* m*, and outputs a secret key* $\widetilde{\mathsf{sk}}$.

Efficiency. *The algorithms* $(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$ *are uniform QPT algorithms.*

Correctness.

$$\Pr\left[m \leftarrow \mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT}) : (\mathsf{pk}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda), \mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda, \mathsf{MSK}), \mathsf{CT} \leftarrow \mathsf{Enc}(1^\lambda, \mathsf{pk}, m)\right] \geq 1 - \mathsf{negl}(\lambda).$$

Security. *Given a receiver non-committing encryption* Σ*, we consider a security experiment* $\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma,\mathcal{A}}(\lambda, b)$ *against* $\mathcal{A}$.

1. *The challenger runs* $(\mathsf{pk}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$ *and sends* pk *to* $\mathcal{A}$.
2. $\mathcal{A}$ *sends* m *to the challenger.*
3. *The challenger does the following:*
   - *If* $b = 0$*, the challenger generates* $\mathsf{CT} \leftarrow \mathsf{Enc}(1^\lambda, \mathsf{pk}, m)$ *and* $\mathsf{sk} \leftarrow \mathsf{KeyGen}(1^\lambda, \mathsf{MSK})$*, and sends* $(\mathsf{CT}, \mathsf{sk})$ *to* $\mathcal{A}$.
   - *If* $b = 1$*, the challenger generates* $(\widetilde{\mathsf{CT}}, \mathsf{aux}) \leftarrow \mathsf{Fake}(1^\lambda, \mathsf{pk})$ *and* $\widetilde{\mathsf{sk}} \leftarrow \mathsf{Reveal}(1^\lambda, \mathsf{pk}, \mathsf{MSK}, \mathsf{aux}, m)$*, and sends* $(\widetilde{\mathsf{CT}}, \widetilde{\mathsf{sk}})$ *to* $\mathcal{A}$.
4. $\mathcal{A}$ *outputs* $b' \in \{0, 1\}$*, and the experiment outputs* 1 *if* $b' = b$.

We say that Σ *is RNC secure if for all sufficiently large security parameters* $\lambda \in \mathbb{N}$*, for any QPT adversary* $\mathcal{A}$*, it holds that*

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

Construction We construct unclonable PKE $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ from one-time unclonable SKE $\Sigma_{\mathsf{SKE}} = \mathsf{SKE}.(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ and receiver non-committing encryption with quantum ciphertexts $\Sigma_{\mathsf{NCE}} = \mathsf{NCE}.(\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Fake}, \mathsf{Reveal})$:

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

- Run $(\mathsf{nce.pk}, \mathsf{nce.MSK}) \leftarrow \mathsf{NCE.Setup}(1^\lambda)$ and $\mathsf{nce.sk} \leftarrow \mathsf{NCE.KeyGen}(1^\lambda, \mathsf{nce.MSK})$.
- Output $\mathsf{pk} := \mathsf{nce.pk}$ and $\mathsf{sk} := \mathsf{nce.sk}$.

$\mathsf{Enc}(1^\lambda, \mathsf{pk}, m)$**:**

- Parse $\mathsf{pk} = \mathsf{nce.pk}$.
- Run $\mathsf{ske.sk} \leftarrow \mathsf{SKE.KeyGen}(1^\lambda)$ and $\mathsf{ske.CT} \leftarrow \mathsf{SKE.Enc}(1^\lambda, \mathsf{ske.sk}, m)$.
- Run $\mathsf{nce.CT} \leftarrow \mathsf{NCE.Enc}(1^\lambda, \mathsf{nce.pk}, \mathsf{ske}, \mathsf{sk})$.
- Output $\mathsf{CT} := (\mathsf{nce.CT}, \mathsf{ske.CT})$.

$\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT})$**:**

- Parse $\mathsf{sk} = \mathsf{nce.sk}$ and $\mathsf{CT} = (\mathsf{nce.CT}, \mathsf{ske.CT})$.
- Run $\mathsf{ske.sk} \leftarrow \mathsf{NCE.Dec}(1^\lambda, \mathsf{nce.sk}, \mathsf{nce.CT})$.
- Run $\mathsf{SKE.Dec}(1^\lambda, \mathsf{ske.sk}, \mathsf{ske.CT})$ and outputs its output.

Obviously, Σ satisfies efficiency, correctness, and IND-CPA security.

Lemma E.2. *Σ satisfies unclonable IND-CPA security.*

Proof. We describe the sequence of hybrids against QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$.

Hyb_0**:** This is the original security experiment of Σ.

1. The challenger samples $b \leftarrow \{0, 1\}$.
2. $\mathcal{A}$ receives $\mathsf{nce.pk}$, where $(\mathsf{nce.pk}, \mathsf{nce.MSK}) \leftarrow \mathsf{NCE.Setup}(1^\lambda)$.
3. $\mathcal{A}$ sends (m_0, m_1) to the challenger.
4. $\mathcal{A}$ receives $(\mathsf{nce.CT}, \mathsf{ske.CT}_b)$ from the challenger, where $\mathsf{ske.sk} \leftarrow \mathsf{SKE.KeyGen}(1^\lambda)$, $\mathsf{nce.CT} \leftarrow \mathsf{NCE.Enc}(1^\lambda, \mathsf{nce.pk}, \mathsf{ske.sk})$ and $\mathsf{ske.CT}_b \leftarrow \mathsf{SKE.Enc}(1^\lambda, \mathsf{ske.sk}, m_b)$.
5. $\mathcal{A}$ generates $\rho_{\mathcal{B},\mathcal{C}}$ and sends the $\mathcal{B}$ and $\mathcal{C}$ register to $\mathcal{B}$ and $\mathcal{C}$, respectively.
6. $\mathcal{B}$ and $\mathcal{C}$ receives $\mathsf{nce.sk}$ and outputs $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$, respectively, where $\mathsf{nce.sk} \leftarrow \mathsf{NCE.KeyGen}(1^\lambda, \mathsf{nce.MSK})$.
7. The experiment outputs 1 if $b = b_{\mathcal{B}} = b_{\mathcal{C}}$.

Hyb_1**:** This is the same as Hyb_0 except that $(\widetilde{\mathsf{nce.CT}}, \widetilde{\mathsf{nce.sk}})$ is used instead of $(\mathsf{nce.CT}, \mathsf{nce.sk})$, where $(\widetilde{\mathsf{nce.CT}}, \mathsf{aux}) \leftarrow \mathsf{Fake}(1^\lambda, \mathsf{nce.pk})$ and $\widetilde{\mathsf{nce.sk}} \leftarrow \mathsf{Reveal}(1^\lambda, \mathsf{nce.pk}, \mathsf{nce.MSK}, \mathsf{aux}, \mathsf{ske.sk})$.

We have the following Propositions E.3 and E.4.

Proposition E.3. *If Σ_{NCE} is RNC secure, then*

$$|\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]| \leq \mathsf{negl}(\lambda).$$

Proposition E.4. *If Σ_{SKE} is one-time unclonable IND-CPA secure, then*

$$\Pr[\mathsf{Hyb}_1 = 1] \leq 1/2 + \mathsf{negl}(\lambda).$$

□

Proof of Proposition E.3. This can be shown by a standard hybrid argument. Assume that there a QPT adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ such that

$$|\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]|$$

is non-negligible. Then, construct a QPT adversary $\widetilde{\mathcal{A}}$ that breaks the RNC security of Σ_{NCE} as follows.

1. $\widetilde{\mathcal{A}}$ samples $b \leftarrow \{0,1\}$.
2. $\widetilde{\mathcal{A}}$ receives $\mathsf{nce.pk}$ from the challenger of $\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma_{\mathsf{NCE}},\widetilde{\mathcal{A}}}(\lambda, b^*)$, where $(\mathsf{nce.pk}, \mathsf{nce.MSK}) \leftarrow \mathsf{NCE.Setup}(1^\lambda)$.
3. $\widetilde{\mathcal{A}}$ sends $\mathsf{nce.pk}$ to $\mathcal{A}$, and receives (m_0, m_1) from $\mathcal{A}$.
4. $\widetilde{\mathcal{A}}$ samples $\mathsf{ske.sk} \leftarrow \mathsf{SKE.KeyGen}(1^\lambda)$, computes $\mathsf{ske.CT}_b \leftarrow \mathsf{SKE.Enc}(1^\lambda, \mathsf{ske.sk}, m_b)$, and sends $\mathsf{ske.sk}$ to the challenger of $\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma_{\mathsf{NCE}},\widetilde{\mathcal{A}}}(\lambda, b^*)$.
5. The challenger of $\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma_{\mathsf{NCE}},\widetilde{\mathcal{A}}}(\lambda, b^*)$ works as follows:
   - If $b^* = 0$, then runs $\mathsf{nce.CT}^* \leftarrow \mathsf{NCE.Enc}(1^\lambda, \mathsf{nce.pk}, \mathsf{ske.sk})$ and $\mathsf{nce.sk}^* \leftarrow \mathsf{NCE.KeyGen}(1^\lambda, \mathsf{nce.MSK})$, and sends $(\mathsf{nce.CT}^*, \mathsf{nce.sk}^*)$ to $\widetilde{\mathcal{A}}$.
   - If $b^* = 1$, then runs $(\mathsf{nce.CT}^*, \mathsf{aux}) \leftarrow \mathsf{Fake}(1^\lambda, \mathsf{nce.pk})$ and $\mathsf{nce.sk}^* \leftarrow \mathsf{Reveal}(1^\lambda, \mathsf{nce.pk}, \mathsf{nce.MSK}, \mathsf{aux}, \mathsf{ske.sk})$, and sends $(\mathsf{nce.CT}^*, \mathsf{nce.sk}^*)$ to $\widetilde{\mathcal{A}}$.
6. $\widetilde{\mathcal{A}}$ runs $\mathcal{A}$ on $(\mathsf{ske.CT}_b, \mathsf{nce.CT}^*)$, and obtains $\rho_{\mathcal{B},\mathcal{C}}$.
7. $\widetilde{\mathcal{A}}$ sends $\mathsf{nce.sk}^*$ and the $\mathcal{B}$ register (resp. the $\mathcal{C}$ register) to $\mathcal{B}$ (resp. $\mathcal{C}$).
8. $\mathcal{B}$ and $\mathcal{C}$ outputs $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$, respectively.
9. $\widetilde{\mathcal{A}}$ outputs 1 if $b = b_{\mathcal{B}} = b_{\mathcal{C}}$, and 0 otherwise.

From the construction of $\widetilde{\mathcal{A}}$,

- If $b^* = 0$, $\widetilde{\mathcal{A}}$ perfectly simulates the challenger of Hyb_0 and thus it outputs the output of Hyb_0.
- If $b^* = 1$, $\widetilde{\mathcal{A}}$ perfectly simulates the challenger of Hyb_1 and thus it outputs the output of Hyb_1.

Therefore, we have

$$\left|\Pr\left[\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma_{\mathsf{NCE}},\widetilde{\mathcal{A}}}(\lambda, 0) = 1\right] - \Pr\left[\mathsf{Exp}^{\mathsf{rec\text{-}nc}}_{\Sigma_{\mathsf{NCE}},\widetilde{\mathcal{A}}}(\lambda, 1) = 1\right]\right| = |\Pr[\mathsf{Hyb}_0 = 1] - \Pr[\mathsf{Hyb}_1 = 1]|,$$

which contradicts that Σ_{NCE} satisfies RNC security. □

Proof of Proposition E.4. This can be shown by a standard hybrid argument. Assume that there there exists a constant C and QPT adversaries $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ such that

$$\Pr[\mathsf{Hyb}_1 = 1] \geq 1/2 + 1/\lambda^C$$

for infinitely many security parameters $\lambda \in \mathbb{N}$. Then, construct a set of QPT adversaries $(\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}}, \widetilde{\mathcal{C}})$ that breaks the unclonable IND-CPA security of Σ_{SKE} as follows.

1. The challenge of Σ_{SKE} samples $b \leftarrow \{0,1\}$.
2. $\widetilde{\mathcal{A}}$ samples $(\mathsf{nce.pk}, \mathsf{nce.MSK}) \leftarrow \mathsf{NCE.Setup}(1^\lambda)$ and sends $\mathsf{nce.pk}$ to $\mathcal{A}$.
3. $\widetilde{\mathcal{A}}$ receives (m_0, m_1) from $\mathcal{A}$, and sends (m_0, m_1) to the challenger.

4. $\widetilde{\mathcal{A}}$ receives $\mathsf{ske.CT}_b$, where $\mathsf{ske.CT}_b \leftarrow \mathsf{SKE.Enc}(1^\lambda, \mathsf{ske.sk}, m_b)$ and $\mathsf{ske.sk} \leftarrow \mathsf{SKE.KeyGen}(1^\lambda)$.
5. $\widetilde{\mathcal{A}}$ runs $(\widetilde{\mathsf{nce.CT}}, \mathsf{aux}) \leftarrow \mathsf{Fake}(1^\lambda, \mathsf{nce.pk})$, and runs $\mathcal{A}$ on $(\widetilde{\mathsf{nce.CT}}, \mathsf{ske.CT}_b)$, and obtains $\rho_{\mathcal{B},\mathcal{C}}$.
6. $\widetilde{\mathcal{A}}$ sends aux, $\mathsf{nce.MSK}$ and the $\mathcal{B}$ (resp. $\mathcal{C}$) register to $\widetilde{\mathcal{B}}$ (resp. $\widetilde{\mathcal{C}}$).
7. $\widetilde{\mathcal{B}}$ (resp. $\widetilde{\mathcal{C}}$) receives $\mathsf{ske.sk}$ and runs $\widetilde{\mathsf{nce.sk}} \leftarrow \mathsf{Reveal}(1^\lambda, \mathsf{nce.pk}, \mathsf{nce.MSK}, \mathsf{aux}, \mathsf{ske.sk})$, and sends $\widetilde{\mathsf{nce.sk}}$ and the $\mathcal{B}$ (resp. $\mathcal{C}$) register to $\mathcal{B}$ (resp. $\mathcal{C}$).
8. $\mathcal{B}$ and $\mathcal{C}$ outputs $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$, respectively.
9. $\widetilde{\mathcal{B}}$ and $\widetilde{\mathcal{C}}$ outputs $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$ as the guess for b, respectively.

From the construction of $(\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}}, \widetilde{\mathcal{C}})$, it perfectly simulates the challenger of Hyb_1. Therefore, we have $b = b_{\mathcal{B}} = b_{\mathcal{C}}$ with non-negligible probability, which implies that $(\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}}, \widetilde{\mathcal{C}})$ break one-time unclonable IND-CPA security of Σ_{SKE}. This is a contradiction. Therefore, we have

$$\Pr[\mathsf{Hyb}_1 = 1] \leq 1/2 + \mathsf{negl}(\lambda).$$

□

F Proof of Proposition 8.9

We give the proof of Proposition 8.9.

Proof of Proposition 8.9. In the same way as proof of Lemma 7.4, we can show that if there exists a one-time unclonable secret-key encryption for single-bit plaintexts, then there exists a scheme $\Sigma^* = (\mathsf{KeyGen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$ that satisfies perfect correctness.

Now, we construct one-time unclonable secret key encryption $\overline{\Sigma} := (\overline{\mathsf{KeyGen}}, \overline{\mathsf{Enc}}, \overline{\mathsf{Dec}})$ with uniformly random secret-key and perfect correctness from one-time unclonable secret key encryption $(\mathsf{KeyGen}^*, \mathsf{Enc}^*, \mathsf{Dec}^*)$ with perfect correctness.

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

- Sample $s \leftarrow \{0,1\}^{s(\lambda)}$, where $s(\lambda)$ is the length of the secret-key sk that $\mathsf{KeyGen}^*(1^\lambda)$ generates
- Output $\overline{\mathsf{sk}} := s$.

$\overline{\mathsf{Enc}}(1^\lambda, \overline{\mathsf{sk}}, m)$**:**

- Parse $\overline{\mathsf{sk}} := s$.
- Run $\mathsf{sk} \leftarrow \mathsf{KeyGen}^*(1^\lambda)$.
- Run $\mathsf{CT} \leftarrow \mathsf{Enc}^*(1^\lambda, \mathsf{sk}, m)$.
- Output $\overline{\mathsf{CT}} := (\mathsf{CT}, \mathsf{sk} + s)$.

$\overline{\mathsf{Dec}}(1^\lambda, \overline{\mathsf{sk}}, \overline{\mathsf{CT}})$**:**

- Parse $\overline{\mathsf{sk}} = s$ and $\overline{\mathsf{CT}} = (\mathsf{CT}, \mathsf{sk}^*)$.
- Compute $\mathsf{sk} = \mathsf{sk}^* + s$.
- Run $\mathsf{Dec}^*(1^\lambda, \mathsf{sk}, \mathsf{CT})$ and output its output.

From the construction, the secret key of Σ^* is uniformly random. Efficiency and perfect correctness of $\overline{\Sigma}$ straightforwardly follow that of Σ^*. We can show that $\overline{\Sigma}$ satisfies unclonable IND-CPA security by a standard hybrid argument. For clarity, we describe the proof of security.

Assume that there exists a QPT adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ that breaks the unclonable IND-CPA security of $\overline{\Sigma}$. Then, construct a QPT adversary $(\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}}, \widetilde{\mathcal{C}})$ that breaks the unclonable IND-CPA security of Σ^*.

1. The challenger of Σ^* samples $b \leftarrow \{0,1\}$.
2. $\widetilde{\mathcal{A}}$ samples $s \leftarrow \{0,1\}^{s(\lambda)}$.
3. $\widetilde{\mathcal{A}}$ receives (m_0, m_1) from $\mathcal{A}$.
4. $\widetilde{\mathcal{A}}$ sends (m_0, m_1) to the challenger of Σ^*.
5. $\widetilde{\mathcal{A}}$ receives CT_b, where $\mathsf{sk} \leftarrow \mathsf{KeyGen}^*(1^\lambda)$ and $\mathsf{CT}_b \leftarrow \mathsf{Enc}^*(1^\lambda, \mathsf{sk}, m_b)$.
6. $\widetilde{\mathcal{A}}$ runs $\mathcal{A}$ on (CT_b, s), obtain $\rho_{\mathcal{B},\mathcal{C}}$, and sends s and the $\mathcal{B}$ register (resp. $\mathcal{C}$ register) to $\widetilde{\mathcal{B}}$ (resp. $\widetilde{\mathcal{C}}$).
7. $\widetilde{\mathcal{B}}$ (resp. $\widetilde{\mathcal{C}}$) receives sk from the challenger of Σ^*, and sends $\mathsf{sk} + s$ and the $\mathcal{B}$ register (resp. $\mathcal{C}$ register) to $\mathcal{B}$ (resp.$\mathcal{C}$).
8. The experiment outputs 1 if $b = b_{\mathcal{B}} = b_{\mathcal{C}}$, where $b_{\mathcal{B}}$ and $b_{\mathcal{C}}$ are the output of $\mathcal{B}$ and $\mathcal{C}$,respectively.

From the construction of $(\widetilde{\mathcal{A}}, \widetilde{\mathcal{B}}, \widetilde{\mathcal{C}})$, it perfectly simulates the challenger of Σ^*. Therefore, if $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ breaks the unclonable IND-CPA security of $\overline{\Sigma}$, it contradicts that Σ^* satisfies unclonable IND-CPA security.

In the construction $\overline{\Sigma}$, the size of $\overline{\mathsf{sk}_\lambda}$ and $\overline{\mathsf{CT}_{\lambda,b}}$ are not necessarily equal to λ, where $\overline{\mathsf{sk}_\lambda} \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$ and $\overline{\mathsf{CT}_{\lambda,b}} \leftarrow \overline{\mathsf{Enc}}(1^\lambda, \overline{\mathsf{sk}_\lambda}, b)$. By wisely choosing a security parameter and a standard padding argument, from $\overline{\Sigma}$, we can construct $\Sigma = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ such that $|\mathsf{sk}_\lambda| = |\mathsf{CT}_{\lambda,b}| = \lambda$ for all $\lambda \in \mathbb{N}$ and b where $\mathsf{sk}_\lambda \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $\mathsf{CT}_{\lambda,b} \leftarrow \mathsf{Enc}(1^\lambda, \mathsf{sk}_\lambda, b)$.

For clarity, we describe the construction of Σ. To describe our construction, let c be a constant such that $\left|\overline{\mathsf{sk}_\lambda}\right| \leq \left|\overline{\mathsf{CT}_{\lambda,b}}\right| \leq \lambda^c$ for all security parameters $\lambda \in \mathbb{N}$ and $b \in \{0,1\}$, where $\overline{\mathsf{sk}_\lambda} \leftarrow \overline{\mathsf{KeyGen}}(1^\lambda)$ and $\overline{\mathsf{CT}_{\lambda,b}} \leftarrow \overline{\mathsf{Enc}}(1^\lambda, \overline{\mathsf{sk}_\lambda}, b)$.

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

- Sample $x \leftarrow \{0,1\}^\lambda$.
- Output $\mathsf{sk} := x$.

$\mathsf{Enc}(1^\lambda, \mathsf{sk}, b)$**:**

- Parse $\mathsf{sk} = x$.
- Let λ' be the largest integer such that $\lambda'^c \leq \lambda$.
- Let $\overline{x}$ be the first $\left|\overline{\mathsf{sk}_{\lambda'}}\right|$-bits of x, where $\overline{\mathsf{sk}_{\lambda'}} \leftarrow \overline{\mathsf{KeyGen}}(1^{\lambda'})$.
- Run $\overline{\mathsf{CT}} \leftarrow \overline{\mathsf{Enc}}(1^{\lambda'}, \overline{x}, b)$. Note that since $\lambda'^c \leq \lambda$, the size of $\overline{\mathsf{CT}}$ is smaller than λ.
- Output $\mathsf{CT} = (\overline{\mathsf{CT}}, 0^{\lambda - |\overline{\mathsf{CT}}|})$.

$\mathsf{Dec}(1^\lambda, \mathsf{sk}, \mathsf{CT})$**:**

- Parse $\mathsf{sk} = x$ and $\mathsf{CT} = (\overline{\mathsf{CT}}, 0^{\lambda - |\overline{\mathsf{CT}}|})$.
- Let λ' be the largest integer such that $\lambda'^c \leq \lambda$.
- Let $\overline{x}$ be the first $\left|\overline{\mathsf{sk}_{\lambda'}}\right|$-bits of x, where $\overline{\mathsf{sk}_{\lambda'}} \leftarrow \overline{\mathsf{KeyGen}}(1^{\lambda'})$.
- Compute $\overline{\mathsf{Dec}}(1^{\lambda'}, \overline{x}, \overline{\mathsf{CT}})$, and outputs its output.

Efficiency and perfect correctness straightforwardly follow. From the construction, it is obvious that sk_λ is uniformly randomly sampled and $|\mathsf{sk}_\lambda| = |\mathsf{CT}_{\lambda,b}| = \lambda$ for all $\lambda \in \mathbb{N}$ and $b \in \{0,1\}$, where $\mathsf{sk}_\lambda \leftarrow \mathsf{KeyGen}(1^\lambda)$ and $\mathsf{Enc}(1^\lambda, \mathsf{sk}_\lambda, b)$. Furthermore, we can prove its security by a standard hybrid argument. □