Uncloneable Encryption, Revisited

Prabhanjan Ananth
UCSB
prabhanjan@cs.ucsb.edu

Fatih Kaleoglu
UCSB
kaleoglu@ucsb.edu

Abstract

Uncloneable encryption, introduced by Broadbent and Lord (TQC'20), is an encryption scheme with the following attractive feature: an adversary cannot create multiple ciphertexts which encrypt to the same message as the original ciphertext.

We revisit this notion and show the following:

1. **Reusability**: The constructions proposed by Broadbent and Lord have the disadvantage that they either guarantee one-time security (that is, the encryption key can only be used once to encrypt the message) in the plain model or they guaranteed security in the random oracle model. We construct uncloneable encryption schemes, where the encryption key can be re-used to encrypt multiple messages. We present two constructions from minimal cryptographic assumptions: (i) a private-key uncloneable encryption scheme assuming post-quantum one-way functions and, (ii) a public-key uncloneable encryption scheme assuming a post-quantum public-key encryption scheme.
2. **Lower Bound and Generalized Construction**: We also revisit the information-theoretic one-time secure construction of Broadbent and Lord. The success probability of the adversary in their construction was guaranteed to be 0.85^n, where n is the length of the message. It was interesting to understand whether the ideal success probability of (negligibly close to) 0.5^n was unattainable. We demonstrate a simple attack that breaks the scheme with probability 0.71^n. We also generalize their construction to be based on a broader class of monogamy of entanglement games (while their construction was based on BB84 game).
3. **Implication to Copy-Protection**: We also show that uncloneable encryption, satisfying a stronger property, called uncloneable-indistinguishability (defined by Broadbent and Lord), implies copy-protection for a simple class of unlearnable functions. While we currently don't have encryption schemes satisfying this stronger property, this implication demonstrates a new path to construct copy-protection.

1 Introduction

Quantum mechanics has led to the discovery of many fascinating cryptographic primitives [Wie83, Aar09, BGS13, Zha19, AGKZ20, BI20, GZ20, ALP21, ALL+20] that are simply not feasible using classical computing. A couple of popular primitives include quantum money [Wie83] and quantum copy-protection [Aar09]. We study one such primitive in this work.

Inspired by the work of Gottesman [Got02] on tamper detection, Broadbent and Lord introduced the beautiful notion of uncloneable encryption [BL20]. This notion is an encryption scheme

that has the following attractive feature: given any encryption of a classical message $m \in \{0,1\}^*$, modeled as a quantum state, the adversary should be unable to generate multiple ciphertexts that encrypt to the same message. Formally speaking, the uncloneability property is modeled as a game between the challenger and the adversary. The adversary consists of three algorithms, denoted by Alice, Bob and Charlie. The challenger samples a message m uniformly at random and then sends the encryption of m to Alice, who then outputs a bipartite state. Bob gets a part of this state and Charlie gets a different part of the state. Then the reveal phase is executed: Bob and Charlie each independently receive the decryption key. Bob and Charlie – who no longer can communicate with each other – now are expected to guess the message m simultaneously. If they do, we declare that the adversary succeeds in this game. An encryption scheme satisfies uncloneability property if any adversary succeeeds in this game with probability at most negligible in the length of m. Note that the no-cloning principle [WZ82] of quantum mechanics is baked into this definition since if it were possible to copy the ciphertext, Alice can send this ciphertext to both Bob and Charlie who can then decrypt this using the decryption key (obtained during the reveal phase) to obtain the message m.

Broadbent and Lord proposed two novel constructions of uncloneable encryption. The drawback of their information-theoretic scheme is that it only guaranteed one-time security. This means that the encryption key can only be used to encrypt one message, after which the key can no longer be used to encrypt messages without compromising on security. On the other hand, their second scheme does provides reusable security, albeit only in the stronger random oracle model. Another (related) drawback is that their schemes were inherently private-key schemes, meaning that only the entity possessing the private encryption key could compute the ciphertext.

1.1 Our Work

Reusability. We revisit the notion of uncloneable encryption of [BL20] and present two constructions. Both of our constructions guarantee reusable security; we can use the same key to encrypt multiple messages. The first construction is a private-key scheme (the encryption key is private) while the second construction is a public-key scheme (the encryption key is available to everyone).

Theorem 1 (Informal)**.** *Assuming post-quantum one-way functions*[1]*, there exists a private-key uncloneable encryption scheme.*

Theorem 2 (Informal)**.** *Assuming the existence of post-quantum public-key encryption schemes*[2]*, there exists a public-key uncloneable encryption scheme.*

Our constructions only guarantee computational security, unlike the previous scheme of Broadbent and Lord. However, our assumptions are the best one can hope for: (a) a private-key *uncloneable* encryption scheme implies a post-quantum private encryption scheme (and thus, post-quantum one-way functions) and, (b) a public-key *uncloneable* encryption scheme implies a public-key encryption scheme. There are candidates from lattices for both post-quantum one-way functions and post-quantum public-key encryption schemes; for example, see [Reg09].

[1] A function f is one-way and post-quantum secure if given $f(x)$, where $x \in \{0,1\}^\lambda$ is sampled uniformly at random, a quantum polynomial-time (QPT) adversary can recover a pre-image of $f(x)$ with probability only negligible in λ.

[2] An encryption scheme is said to be a post-quantum public-key encryption scheme if any quantum polynomial-time (QPT) adversary can distinguish encryptions of two equal-length messages m_0, m_1 with only negligible probability.

Lower Bound and Generalized Construction. The success probability of the adversary in the information-theoretic construction of [BL20] was shown to be $\left(\frac{1}{2}+\frac{1}{2\sqrt{2}}\right)^n \approx 0.85^n$, where n is the length of the messages. Ideally, we would like the success probability of the adversary to be negligibly close to 0.5^n. A natural question to ask is if we can present a different analysis of their construction that gives the optimal bound. We show, in the theorem below, that this is not the case.

Theorem 3 (Informal)**.** *In the conjugate encryption scheme [BL20], a cloning adversary can succeed with probability at least* 0.71^n.

The adversary that achieves this bound is simple: Alice clones the ciphertext with high fidelity using a generic cloning channel [BCMDM00]. After learning the key, Bob and Charlie both try to honestly decrypt their state, and the output of the decryption matches the original message with significant probability for both of them.

This adversarial construction inherently relies on the fact that the ciphertext (in qubits) is not larger than the message (in bits). For uncloneable encryption schemes with large ciphertext size, it is infeasible to achieve a nontrivial bound using this technique.

The construction of [BL20] is based on the BB84 monogamy of entanglement game [TFKW13], whose adversarial success probability is $\left(\frac{1}{2}+\frac{1}{2\sqrt{2}}\right)$. In the hope of improving the bound, we present a simple generalization of their construction by showing a transformation from a broader class of monogamy games to uncloneable encryption; whereas, [BL20] only showed the transformation for the BB84 monogamy game.

Implication to Copy-Protection. We show how to use uncloneable encryption to build quantum copy-protection [Aar09]. Roughly speaking, using a quantum copy-protection scheme, we can copy-protect our programs in such a way that an adversarial entity cannot create multiple versions of this copy-protected program. Recently, this notion has been revisited by many recent works [ALP21, CMP20, ALL+20, KNY20, BJL+21].

However, despite the recent progress, to date, we don't know of any provably secure constructions of copy-protection. We show how to use uncloneable encryption to construct copy-protection for a specific class of point functions. This class consists of functions of the form $f_{a,b}(\cdot)$, where b is a concatenation of the verification key and a signature on 0, that take as input x and output b if and only if $x = a$. This would not immediately yield a provably construction of copy-protection since we need the underlying uncloneable encryption to satisfy a stronger property called *uncloneable indistinguishability* property (see Definition 9) that are not currently satisfied by existing constructions of uncloneable encryption. Nonetheless, this gives a new pathway to demonstrating provably secure constructions of quantum copy-protection.

Theorem 4 (Informal)**.** *Assuming the existence of uncloneable encryption scheme satisfying uncloneable-indistinguishability property and post-quantum one-way functions, there exists a quantum copy-protection scheme, satisfying computational correctness, for a special class of point functions.*

The resulting copy-protection guarantees a weaker correctness property called computational correctness property; informally, this says that any quantum polynomial-time adversary cannot come

up with an input such that the copy-protected circuit is incorrect on this input. We note that such a correctness notion has been studied previously in the context of obfuscation [BLMZ19] (under the name computational functionality preservation). In addition to uncloneable encryption, we use a post-quantum digital signature scheme that can be based on post-quantum one-way functions.

Our construction is inspired a construction of secure software leasing by [BJL+21]. Conceptually, we follow the same approach suggested in their paper, except we replace the tool of quantum authentication codes [BCG+] with uncloneable encryption.

Concurrent Works. The work of Majenz, Schaffner and Tahmasbi [MST21] study various limitations on uncloneable encryption schemes. Specifically, they analyze lower bounds for the success probability of the adversary in any uncloneable encryption scheme. In contrast, our lower bound targets specifically the conjugate encryption scheme of [BL20] and this allowed to present concrete lower bounds.

Hiroka et al. [HMNY21] showed how to make the key reusable in a different primitive called quantum encryption with certified deletion [BI20] using the same conceptual idea but different tools. We note that uncloneable encryption implies quantum encryption with certified deletion if the certificate of deletion is allowed to be quantum. However, Hiroka et al.'s result achieves classical certification of deletion.

1.2 Technical Overview

We present a high level overview of our techniques.

Naive Attempt: A Hybrid Approach. A naive attempt to construct an uncloneable encryption scheme with reusable security is to start with two encryption schemes.

- The first scheme is a (one-time) uncloneable encryption scheme, as considered in the work of [BL20]. We denote this scheme by otUE.
- The second scheme is a post-quantum encryption scheme guaranteeing reusable security but without any uncloneability guarantees[3]. We denote this scheme by $\mathcal{E}$.

At a high level, we hope that we can combine the above two schemes to get the best of both worlds: reusability and uncloneability.

In more detail, using otUE and $\mathcal{E}$, we construct a reusable uncloneable encryption scheme, denoted by rUE, as follows. Sample a decryption key $k_{\mathcal{E}}$ according to the scheme $\mathcal{E}$ and set the decryption key of rUE to be $k_{\mathcal{E}}$. The encryption procedure of rUE is defined as follows. To encrypt a message m, first sample a key k_{otUE} according to the scheme otUE. Output the rUE encryption of m to be $(\mathsf{CT}_{\mathsf{otUE}}, \mathsf{CT}_{\mathcal{E}})$, where $\mathsf{CT}_{\mathsf{otUE}}$ is an encryption of m under the key k_{otUE} and, $\mathsf{CT}_{\mathcal{E}}$ is an encryption of the message k_{otUE} under the key $k_{\mathcal{E}}$. To decrypt, first decrypt $\mathsf{CT}_{\mathcal{E}}$ using $k_{\mathcal{E}}$ to obtain the message k_{otUE}. Using this, then decrypt $\mathsf{CT}_{\mathsf{otUE}}$ to get the message m.

How do we argue uncloneability? Ideally, we would like to reduce the uncloneability property of rUE to the uncloneability property of the underlying one-time scheme otUE. However, we cannot

[3] As an example, we could use Regev's public-key encryption scheme [Reg09].

immediately perform this reduction. The reason being that k_{otUE} is still encrypted under the scheme $\mathcal{E}$ and thus, we need to get rid of this key before invoking the uncloneability property of otUE. To get rid of this key, we need to invoke the semantic security of $\mathcal{E}$. Unfortunately, we cannot invoke the semantic security of $\mathcal{E}$ since the decryption key of $\mathcal{E}$ will be revealed to the adversary and semantic security is trivially violated if the adversary gets the decryption key.

More concretely, Alice upon receiving $(\mathsf{CT}_{\mathsf{otUE}}, \mathsf{CT}_{\mathcal{E}})$ could first break $\mathsf{CT}_{\mathcal{E}}$ to recover k_{otUE} and then decrypt $\mathsf{CT}_{\mathsf{otUE}}$ using k_{otUE} to recover m. Thus, before performing the reduction to rUE, we need to first invoke the security property of $\mathcal{E}$. Here is where we are stuck: as part of the security experiment of the uncloneability property, we need to reveal the decryption key of rUE, which is nothing but $k_{\mathcal{E}}$, to Bob and Charlie after Alice produces the bipartite state. But if we reveal $k_{\mathcal{E}}$, then the security of $\mathcal{E}$ is no longer guaranteed.

Embedding Messages into Keys. To overcome the above issue, we require $\mathcal{E}$ to satisfy an additional property. Intuitively, this property guarantees the existence of an algorithm that produces a fake decryption key that has embedded inside it a message m such that this fake decryption key along with an encryption of 0 should be indistinguishable from an honestly generated decryption key along with an encryption of m.

> **Fake-Key Property**: there is a polynomial-time algorithm $FakeGen$ that given an encryption of 0, denoted by CT_0, and a message m, outputs a fake key fk such that the distributions $\{(\mathsf{CT}_m, k_{\mathsf{PKE}})\}$ and $\{(\mathsf{CT}_0, fk)\}$ are computationally indistinguishable, where CT_m is an encryption of m and k_{PKE} is the decryption key of PKE.

One consequence of the above property is that the decryption of CT_0 using the fake decryption key fk yields the message m.

Using the above fake-key property, we can now fix the issue in the above hybrid approach. Instead of invoking semantic security of $\mathcal{E}$, we instead invoke the fake-key property of PKE. The idea is to remove k_{otUE} completely in the generation $\mathsf{CT}_{\mathcal{E}}$ and only use it during the reveal phase, when the decryption key is revealed to both Bob and Charlie. That is, $\mathsf{CT}_{\mathcal{E}}$ is computed to be an encryption of 0 and instead of revealing the honestly generated key $k_{\mathcal{E}}$ to Bob and Charlie, we instead reveal a fake key that has embedded inside it the message k_{otUE}. After this change, we will now be ready to invoke the uncloneability property of the underlying one-time scheme.

Instantiation: Private-Key Scheme. We used a reusable encryption scheme $\mathcal{E}$ satisfying the fake-key property to construct an uncloneable encryption satisfying reusable security. But does a scheme satisfying fake-key property even exist?

We present two constructions: a private-key and a public-key encryption scheme satisfying fake-key property. We first start with a private-key encryption scheme. We remark that a slight modification of the classical private-key encryption scheme using pseudorandom functions [Gol07] already satisfies this property[4]. The encryption of a message m using the decryption key $k_{\mathcal{E}} = (k, otp)$ is $\mathsf{CT} = (r, PRF_k(r) \oplus m \oplus otp)$, where $r \in \{0,1\}^\lambda$ is chosen uniformly at random, λ is

[4]For the informed reader, this scheme can be viewed as a special case of a primitive called somewhere equivocal encryption [HJO+16], considered in a completely different context.

a security parameter and PRF is a pseudorandom function. To decrypt a ciphertext (r, θ), first compute $PRF_k(r)$ and then compute $\theta \oplus PRF_k(r) \oplus otp$.

The fake key generation algorithm on input a ciphertext $\mathsf{CT} = (r, \theta)$ and a message m, generates a fake key fk as follows: it first samples a key k' uniformly at random and then sets otp' to be $\theta \oplus PRF_{k'}(r) \oplus m$. It sets fk to be (k', otp'). Note that fk is set up in such a way that decrypting CT using fk yields the message m.

Instantiation: Public-Key Scheme. We can present a construction of a public-key scheme using functional encryption [BSW11, O'N10], a fundamental notion in cryptography. A functional encryption (FE) scheme is an encryption scheme where the authority holding the decryption key (also referred to as master secret key) is given the ability to issue functional keys, of the form sk_f for a function f, such that decrypting an encryption of x using sk_f yields the output $f(x)$.

A first attempt to achieve fake-key property using FE is to design the fake key to be a functional key associated with a function, that has the message m, hardwired inside it. This function is a constant function that always ignores the input and outputs m. There are two issues with this approach: firstly, the fake key is a functional key whereas the real key is the master secret key of the functional encryption scheme. An adversary might be able to tell apart the fake key versus the real key and thus, break the security. Secondly, a public-key functional encryption does not guarantee function-hiding property – the function description could be evident from the description of the functional key. This means that the adversary can read off the message m from the description of the functional key.

The first issue can be solved by making sure that even the real key is a functional key associated with the identity function. The second issue involves a little more work: instead of having m in the clear in the description of the function, we instead hardwire encryption of m in the function description. The decryption key for this ciphertext is encrypted inside the ciphertext of the FE scheme. Thus, we have two modes: (a) in the first mode, we encrypt m using FE and the real key is a functional key associated with the identity function (this function has a dummy ciphertext hardwired inside it) and, (b) in the second mode, we encrypt $\widehat{k}$ using FE and the fake key is a functional key associated with a function, which has a ciphertext c encrypting message m hardwired inside it, that decrypts c using $\widehat{k}$ and outputs the result. This trick is not new and is inspired by the Trojan technique [ABSV15] introduced in a completely different context.

In the technical sections, instead of presenting a public-key encryption satisfying fake-key property using FE, we present a direct construction of public-key uncloneable encryption scheme using FE.

Implication to Copy-Protection. Next, we will show how to construct copy-protection for a specific class of point functions from uncloneable encryption. A point function $f_{a,b}(\cdot)$ is represented as follows: it takes as input x and outputs b if $x = a$, otherwise it outputs 0. Our approach is inspired by a recent work by Broadbent et al. [BJL+21] who show how to construct a weaker version of copy-protection (called secure software leasing [ALP21]) from quantum authentication codes.

A first attempt to construct copy-protection, using uncloneable encryption, is as follows: to copy-protect $f_{a,b}(\cdot)$, output an uncloneable encryption[5] of b under the key a; that is, a is interpreted

[5]It suffices to use a one-time uncloneable encryption scheme [BL20] here.

as the decryption key of the uncloneable encryption scheme. We treat the ciphertext as the copy-protected version of $f_{a,b}(\cdot)$. To evaluate this copy-protected state on input x, run the decryption of this ciphertext with the key x. Output the result of the decryption algorithm.

If the input is $x = a$ then, by the correctness of uncloneable encryption, we get the output b. However, if the input is not a, then we need the guarantee that the output is 0 with high probability. Unfortunately, the properties of uncloneable encryption fall short here. Uncloneable encryption does not have any guarantees if the ciphertext is decrypted using an invalid key. It could very well be the case that on input $a' \neq a$, the output of the copy-protection algorithm is b, thus violating the correctness guarantees.

We use digital signatures to enforce the correctness property of the copy-protection scheme. We restrict our attention to a sub-class of point functions, where we interpret b to be the concatentation of a verification key vk and a signature σ on 0. We subsequently modify the evaluation algorithm of the copy-protection scheme to output (vk, σ') if and only if the decryption algorithm of uncloneable encryption yields (vk, σ') and moreover, σ' is a valid signature on 0. This still does not guarantee the fact that the copy-protection scheme satisfies correctness. The reason being that on an input $a' \neq a$, the output could still be a valid signature on 0. Fortunately, this satisfies a weaker but still useful notion of correctness called computational correctness. This property states that an efficient adversary should not be able to find an input such that the evaluation algorithm outputs the incorrect value on this input. The reason why computational correctness holds is because it would be infeasible for the adversary to find an input such that the program outputs a valid signature on 0; if it did then it violates the unforgeability property of the underlying signature scheme.

We need to show that given the copy-protected program, say ρ, an adversary cannot output two copies, say ρ_1 and ρ_2[6], such that both of them evaluate $f_{a,b}(\cdot)$ with non-negligible probability. We prove this by contradiction. To show this, we first observe that we can get rid of the signature in the uncloneable encryption ciphertext, by invoking the uncloneable-indistinguishability property of the uncloneable encryption scheme. This is where we crucially use the stronger indistinguishability property; this property allows us to change from one message to another message of our choice whereas in the (weaker) uncloneability security property, the challenger is the one choosing the message to be encrypted.

Now, we argue that there has to be a copy, say ρ_1 and evaluation algorithm E_1 (note that the adversary can choose the evaluation algorithms of its choice), such that when E_1 evaluates ρ_1 on the input k, where k is the UE key, then we get a valid signature σ on 0 with non-negligible probability. Using ρ_1 we can then construct a forger that violates the unforgeability property of the digital signature scheme.

1.3 Structure of this Paper

In Section 2 we give preliminary background and definitions. In Section 3, we introduce natural definitions for many-time secure uncloneable encryption in both private-key and public-key settings, as well as discuss the previous constructions given in [BL20]. We give a construction for the private-key setting in Section 4 and for the public-key setting in Section 5. In Section 6 we

[6]Technically, this is incorrect since the two copies could be entangled and as written here, ρ_1 and ρ_2 are unentangled. But this is done just for ease of presentation, our argument can be suitably adapted to the general case.

present a generalized uncloneable encryption construciton using monogamy games, and a lower bound for conjugate encryption. Section 7 shows that an uncloneable encryption scheme satisfying uncloneable-indistinguishable security (Definition 9) implies copy-protection.

2 Preliminaries

2.1 Notation

We denote the security parameter by λ. We denote by $\mathsf{negl}(.)$ an arbitrary negligible function and by poly(.) an arbitrary function upper-bounded by a polynomial. We abbreviate probabilistic (resp., quantum) polynomial time by PPT (resp., QPT).

We denote by $\mathcal{M}$, $\mathcal{K}$, and $\mathcal{CT}$ (or $\mathcal{H}_{\mathcal{CT}}$) the message space, the key space, and the ciphertext space, respectively. The message and the key are classical, whereas the ciphertext can be classical or quantum, depending on the context. We use 0 to denote a string of zeroes depending on the context.

2.2 Quantum Computing

Valid quantum states on a register X are represented by the set of density operators on the Hilbert space $\mathcal{H}_X$, denoted by $\mathcal{D}(\mathcal{H}_X)$. A density operator $\rho : \mathcal{H}_X \to \mathcal{H}_X$ is defined a linear, positive semi-definite operator with unit trace, i.e. $\mathsf{Tr}(\rho) = 1$, where Tr is the trace operator. Density operators represent mixed quantum states, and a pure state $|\psi\rangle \in \mathcal{H}_X$ is represented by $|\psi\rangle\langle\psi| \in \mathcal{D}(\mathcal{H}_X)$.

Valid quantum operations from register X to register Y are represented by linear, completely positive trace-preserving (CPTP) maps $\phi : \mathcal{D}(\mathcal{H}_X) \to \mathcal{D}(\mathcal{H}_Y)$, also known as quantum channels. Valid quantum measurements on register X with outcomes $x \in \mathcal{X}$ are represented by a positive operator-valued measure (POVM) on $\mathcal{D}(\mathcal{H}_X)$, which is denoted by $F = (F_x)_{x\in X}$, where F_x are positive semi-definite operators satisfying $\sum_x F_x = \mathbf{id}_X$, with $\mathbf{id}_X$ being the identity operator on $\mathcal{H}_X$. The probability of measuring outcome x on state ρ equals $\mathsf{Tr}(F_x\rho)$.

An EPR pair over n qubits is a fully entangled bipartite $2n$-qubit state, defined as

$$|\mathsf{EPR}_n\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |xx\rangle,$$

where $(|x\rangle)_{x\in\{0,1\}^n}$ is the standard basis.

Indistinguishability. We define two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ to be computationally indistinguishable, denoted by $\mathcal{D}_0 \approx_c \mathcal{D}_1$, if any QPT distinguisher cannot distinguish the distributions $\mathcal{D}_0$ and $\mathcal{D}_1$.

Distance Measures. There are two common distance measures considered in the literature: trace distance and fidelity. The fidelity of two quantum states $\rho, \sigma \in \mathcal{D}(\mathcal{H}_A)$ is a measure of similarity

between ρ and σ which is defined as

$$F(\rho,\sigma) = \left(\mathsf{Tr}\left(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}\right)\right)^2.$$

If $\sigma = |\psi\rangle\langle\psi|$ is a pure state, the fidelity simplifies to $F(\sigma,\rho) = \langle\psi|\rho|\psi\rangle$. We use the following useful fact: fidelity of two states does not increase under quantum operations. We state this fact from [Nie96] as a lemma below:

Lemma 1 (Monotonicity of Fidelity)**.** *Let $\rho,\sigma \in \mathcal{D}(\mathcal{H}_X)$ and $\varphi : \mathcal{D}(\mathcal{H}_X) \to \mathcal{D}(\mathcal{H}_Y)$ be a CPTP map. Then,*

$$F(\varphi(\rho),\varphi(\sigma)) \geq F(\rho,\sigma).$$

The trace distance of two states ρ and σ, denoted by $T(\rho,\sigma)$ is defined as follows:

$$T(\rho,\sigma) = \frac{1}{2}||\rho-\sigma||_{tr} = \frac{1}{2}\mathsf{Tr}\left(\sqrt{(\rho-\sigma)^\dagger(\rho-\sigma)}\right).$$

Almost As Good As New Lemma. We use the Almost As Good As New Lemma[7] [Aar04], restated here verbatim from [Aar16].

Lemma 2 (Almost As Good As New)**.** *Let ρ be a mixed state acting on $\mathbb{C}^d$. Let U be a unitary and $(\Pi_0,\Pi_1 = 1-\Pi_0)$ be projectors all acting on $\mathbb{C}^d\otimes\mathbb{C}^d$. We interpret (U,Π_0,Π_1) as a measurement performed by appending an acillary system of dimension d' in the state $|0\rangle\langle 0|$, applying U and then performing the projective measurement $\{\Pi_0,\Pi_1\}$ on the larger system. Assuming that the outcome corresponding to Π_0 has probability $1-\varepsilon$, i.e., $\mathsf{Tr}[\Pi_0(U\rho\otimes|0\rangle\langle 0|U^\dagger)] = 1-\varepsilon$, we have*

$$T(\rho,\widetilde{\rho}) \leq \frac{\sqrt{\varepsilon}}{2},$$

where $\widetilde{\rho}$ is state after performing the measurement and then undoing the unitary U and tracing out the ancillary system:

$$\widetilde{\rho} = \mathsf{Tr}_{d'}\left(U^\dagger\left(\Pi_0 U\left(\rho\otimes|0\rangle\langle 0|\right)U^\dagger\Pi_0 + \Pi_1 U\left(\rho\otimes|0\rangle\langle 0|\right)U^\dagger\Pi_1\right)U\right)$$

Corollary 1. *Let $\mathcal{Q}$ be a QPT algorithm which takes as input a state $\rho\in\mathcal{D}(\mathcal{H}_A)$ and outputs a classical string $x\in X$. Then, $\mathcal{Q}$ can be reimplemented as $\widetilde{\mathcal{Q}}$ which satisfies the following properties:*

- *On input $\rho\in\mathcal{D}(\mathcal{H}_A)$, $\widetilde{\mathcal{Q}}$ outputs $\rho'\otimes|x'\rangle\langle x'|\in\mathcal{D}(\mathcal{H}_A)\otimes\mathcal{D}(\mathcal{H}_X)$ such that*

$$\Pr\left[x = x_0 : x\leftarrow\mathcal{Q}(\rho)\right] = \Pr\left[x' = x_0 : \rho'\otimes|x'\rangle\langle x'|\leftarrow\widetilde{\mathcal{Q}}(\rho)\right]$$

  for any $x_0\in X$.

[7] This is also known as the Gentle Measurement Lemma in the quantum information theory literature [Win99].

- *For any state $\rho_0 \in \mathcal{D}(\mathcal{H}_A)$ and a string $x_0 \in X$ satisfying*

$$\Pr\left[x = x_0 : |x\rangle\langle x| \leftarrow Q(\rho_0)\right] \geq 1 - \epsilon,$$

it holds that

$$\Pr\left[x' = x_0 \wedge T(\rho', \rho_0) \leq O(\sqrt{\epsilon}) : \rho' \otimes |x'\rangle\langle x'| \leftarrow \widetilde{Q}(\rho_0)\right].$$

In other words, $\widetilde{Q}$ has the same functionality as Q, and it also outputs a residual state ρ' which is close to ρ in trace distance provided that Q outputs the same string $x \in X$ probability close to 1 on input ρ.

Proof (sketch). By the deferred measurement principle, we can transform Q into the following form without changing its functionality:

- It appends to ρ an ancillary system initialized at $|0\rangle\langle 0|_B$.
- It applies a unitary U to the bipartite state $\rho \otimes |0\rangle\langle 0|_B$ to obtain ρ_{AB}.
- It performs a POVM on ρ_{AB} to measure $|x\rangle\langle x| \in \mathcal{D}(\mathcal{H}_X)$ and output x. Let the residual state be $\widetilde{\rho_{AB}}$ after this measurement.

$\widetilde{Q}$ performs the steps above, and then recovers ρ by applying $U^\dagger$ to $\widetilde{\rho_{AB}}$ and tracing out the ancillary system. The analysis is essentially the same as that in Lemma 2, and we refer the reader to [Aar16] for details. □

2.3 Post-Quantum Digital Signatures

Post-quantum signature schemes with perfect correctness, defined below, can be constructed from post-quantum secure one-way functions:

Definition 1 (Post-Quantum Signature Scheme)**.** *A post-quantum signature scheme over a message space $\mathcal{M}$ is a tuple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$*:*

- **Key Generation:** $\mathsf{Gen}(1^\lambda)$ *takes as input a security parameter and outputs a pair of keys (vk, sk).*
- **Signing:** $\mathsf{Sign}(sk, m)$ *takes as input the secret (signing) key sk and a message $m \in \mathcal{M}$. It outputs a signature σ.*
- **Signature Verification:** $\mathsf{Ver}(vk, m, \sigma')$ *takes as input the verification key vk, a message $m \in \mathcal{M}$ and a candidate signature σ'. It outputs a bit $b \in \{0, 1\}$.*

which satisfy correctness and unforgeability properties defined below:

- **Correctness:** *For all messages $m \in \mathcal{M}$, we have*

$$\Pr[b = 1 : (vk, sk) \leftarrow \mathsf{Gen}(1^\lambda),\ \sigma \leftarrow \mathsf{Sign}(sk, m),\ b \leftarrow \mathsf{Ver}(vk, m, \sigma)] = 1.$$

- **Post-Quantum (One-Time) Existential Unforgeability:** *For any QPT adversary $\mathcal{A}$ and any message $m \in \mathcal{M}$, we have:*

$$\Pr[1 \leftarrow \mathsf{Ver}(vk, m, \sigma') : (vk, sk) \leftarrow \mathsf{Gen}(1^\lambda),\ |\sigma'\rangle\langle\sigma'| \leftarrow \mathcal{A}(vk)] \leq \mathsf{negl}(\lambda).$$

Post-quantum digital signatures can be based on post-quantum one-way functions [Rom90].

2.4 Functional Encryption

A functional encryption scheme allows a user to decrypt an encryption of a message x using a functional key associated with C to obtain the value $C(x)$. The security guarantee states that the user cannot learn anything beyond $C(x)$. Depending on the number of functional keys issued in the security experiment, we can consider different versions of functional encryption. Of interest to us is the notion of single-key functional encryption where the adversary can only query for a single functional key during the security experiment.

A public-key functional encryption scheme FE associated with a class of boolean circuits $\mathcal{C}$ is defined by the following algorithms.

- **Setup,** $\mathsf{Setup}(1^\lambda, 1^s)$: On input security parameter λ, maximum size of the circuits s for which functional keys are issued, output the master secret key MSK and the master public key mpk.
- **Key Generation,** $\mathsf{KeyGen}(\mathsf{MSK}, C)$: On input master secret key MSK and a circuit $C \in \mathcal{C}$ of size s, output the functional key SK_C.
- **Encryption,** $\mathsf{Enc}(\mathsf{mpk}, x)$: On input master public key mpk, input x, output the ciphertext CT.
- **Decryption,** $\mathsf{Dec}(\mathsf{SK}_C, \mathsf{CT})$: On input functional key SK_C, ciphertext CT, output the value y.

Remark 1. *A private-key functional encryption scheme is defined similarly, except that* $\mathsf{Setup}(1^\lambda, 1^s)$ *outputs only the master secret key* MSK *and the encryption algorithm* Enc *takes as input the master secret key* MSK *and the message* x.

A functional encryption scheme satisfies the following properties.

Correctness. Consider an input x and a circuit $C \in \mathcal{C}$ of size s. We require the following to hold for every $Q \geq 1$:

$$\Pr\left[C(x) \leftarrow \mathsf{Dec}(\mathsf{SK}_C, \mathsf{CT}) \ : \ \begin{array}{c}(\mathsf{mpk},\mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda, 1^s);\\ \mathsf{SK}_C \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, C);\\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x)\end{array}\right] \geq 1 - \mathsf{negl}(\lambda),$$

for some negligible function negl.

Single-Key Security. We only consider functional encryption schemes satisfying single-key security property. To define the security of a single-key functional encryption scheme FE, we define two experiments Expt_0 and Expt_1. Experiment Expt_0, also referred to as *real* experiment, is parameterized by a PPT stateful adversary $\mathcal{A}$ and a challenger Ch. Experiment Expt_1, also referred to as the *simulated* experiment, is parameterized by a PPT adversary $\mathcal{A}$ and a PPT stateful simulator Sim.

$\underline{\mathsf{Expt}_0^{\mathsf{FE},\mathcal{A},\mathsf{Ch}}(1^\lambda)}$:

- $\mathcal{A}$ outputs the maximum circuit size s.

- Ch executes $\mathsf{FE.Setup}(1^\lambda, 1^s)$ to obtain the master public key-master secret key pair $(\mathsf{mpk}, \mathsf{MSK})$. It sends mpk to $\mathcal{A}$.
- **Challenge Message Query**: After receiving mpk, $\mathcal{A}$ outputs the challenge message x. The challenger computes the challenge ciphertext $\mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{mpk}, x)$. Ch sends CT to $\mathcal{A}$.
- **Circuit Query**: $\mathcal{A}$ upon receiving the ciphertext CT as input, outputs a circuit C of size s. The challenger then sends SK_C to $\mathcal{A}$, where $\mathsf{SK}_C \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, C)$.
- Finally, $\mathcal{A}$ outputs the bit b.

$\underline{\mathsf{Expt}_1^{\mathsf{FE},\mathcal{A},\mathsf{Sim}}(1^\lambda)}$:

- $\mathcal{A}$ outputs the maximum circuit size s.
- Sim, on input $(1^\lambda, 1^s)$, outputs the master public key mpk.
- **Challenge Message Query**: $\mathcal{A}$ upon receiving a public key mpk, outputs a message x. Sim, upon receiving $1^{|x|}$ (i.e., only the length of the input) as input, outputs the challenge ciphertext CT.
- **Circuit Query**: $\mathcal{A}$ upon receiving the ciphertext CT as input, outputs a circuit C of size s. Sim on input $(C, C(x))$, outputs a functional key SK_C.
- Finally, $\mathcal{A}$ outputs a bit b.

A single-key public-key functional encryption scheme is secure if the output distributions of the above two experiments are computationally indistinguishable. More formally,

Definition 2. *A single-key public-key functional encryption scheme* FE *is* **secure** *if for every large enough security parameter* $\lambda \in \mathbb{N}$*, every PPT adversary* $\mathcal{A}$*, there exists a PPT simulator* Sim *such that the following holds:*

$$\left|\Pr\left[0 \leftarrow \mathsf{Expt}_0^{\mathsf{FE},\mathcal{A},\mathsf{Ch}}(1^\lambda)\right] - \Pr\left[0 \leftarrow \mathsf{Expt}_1^{\mathsf{FE},\mathcal{A},\mathsf{Sim}}(1^\lambda)\right]\right| \leq \mathsf{negl}(\lambda),$$

for some negligible function negl*.*

Instantiations. A single-key public-key functional encryption scheme can be built from any public-key encryption scheme [SS10, GVW12]. If the underlying public-key encryption scheme is post-quantum secure then so is the resulting functional encryption scheme.

2.5 Quantum Copy-Protection

Below we present the definition of a copy-protection scheme, adapted from [BJL+21] and originally due to [Aar09].

Definition 3 (Copy-Protection Scheme)**.** *Let* $\mathcal{F} = \mathcal{F}(\lambda) = \{f : X \to Y\}$ *be a class of efficiently computable functions functions. A copy protection scheme for* $\mathcal{F}$ *is a pair of quantum algorithms* $(\mathsf{CopyProtect}, \mathsf{Eval})$ *such that for some output space* $\mathcal{D}(\mathcal{H}_Z)$*:*

- ***Copy Protected State Generation:*** $\mathsf{CopyProtect}(1^\lambda, d_f)$ *takes as input the security paramter* 1^λ *and a classical description* d_f *of a function* $f \in \mathcal{F}$ *(that efficiently computes* f*). It outputs a mixed state* $\rho_f \in \mathcal{D}(\mathcal{H}_Z)$*.*

- ***Evaluation:*** $\mathsf{Eval}(1^\lambda, \rho, x)$ *takes as input the security parameter* 1^λ*, a mixed state* $\rho \in \mathcal{D}(\mathcal{H}_Z)$*, and an input value* $x \in X$*. It outputs a bipartite state* $\rho' \otimes |y\rangle\langle y| \in \mathcal{D}(\mathcal{H}_Z) \otimes \mathcal{D}(\mathcal{H}_Y)$*.*

Correctness: Informally speaking, if an honestly generated copy-protected state ρ_f for a function $f \in \mathcal{F}$ is honestly evaluated using Eval on any input $x \in X$, the output should be $f(x)$. We defer the formal definition of correctness to Section 2.6, where we define a weaker notion of computational correctness specificially for point functions, which is the context we use copy-protection in throughout Section 7.

Security. Security in the context of copy-protection means that given a copy-protected program ρ_f of a function $f \in \mathcal{F}$, no QPT adversary can produce two programs that can both be used to compute f. This is captured in the following definition adapted by the "malicious-malicious security" definition given in [BJL+21]:

Definition 4 (Copy-Protection Security)**.** *A copy-protection scheme* $(\mathsf{CopyProtect}, \mathsf{Eval})$ *for a class* $\mathcal{F}$ *of functions* $f : X \to Y$ *and a distribution* $\mathcal{D}$ *over* $\mathcal{F}$ *is* $\delta(\lambda)$*-**secure** with respect to a family of distributions* $\left\{\mathcal{D}_X^f\right\}_{f\in\mathcal{F}}$ *over* $X \times X$ *if any QPT adversary* $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ *cannot succeed in the following **pirating experiment** with probability greater than* $\delta(\lambda) + \mathsf{negl}(\lambda)$*:*

- *The challenger samples a function* $f \leftarrow \mathcal{D}$ *and sends* $\rho_f \leftarrow \mathcal{D}$ *to* $\mathcal{A}$*.*

- $\mathcal{A}$ *applies a CPTP map to split* ρ_f *into a bipartite state* ρ_{BC}*, and sends the* B *(resp.,* C*) register to* $\mathcal{B}$ *(resp.,* $\mathcal{C}$*). No communication is allowed between* $\mathcal{B}$ *and* $\mathcal{C}$ *after this step.*

- *The challenger samples* $x \leftarrow \mathcal{D}_X^f$ *and sends* x *to both* $\mathcal{B}$ *and* $\mathcal{C}$*.*

- $\mathcal{B}$ *(resp.,* $\mathcal{C}$*) outputs*[8] $y_B \in Y$ *(resp.,* $y_C \in Y$*). The adversary wins if* $y_B = y_C = f(x)$*.*

Note that this definition is referred to as malicious-malicious security because the adversary is free to choose the registers B, C *as well as the evaluation algorithms used by* $\mathcal{B}$ *and* $\mathcal{C}$*.*

2.6 Copy-Protection of Point Functions

Point Functions: Let a and b be binary strings. The point function $f_{a,b} : \{0,1\}^{|a|} \to \{0,1\}^{|b|}$ is defined as

$$f_{a,b}(x) = \begin{cases} b, & x = a \\ 0, & x \neq a \end{cases}.$$

Ordinarily, one would define the correctness property of a copy-protection scheme as follows: an honest evaluation of $f(x)$ using an honestly generated copy-protected state ρ_f for f succeeds with

[8] Since $\mathcal{B}$ and $\mathcal{C}$ cannot communicate, the order in which they use their share of the copy-protected program is insignificant.

small error for all x. For point functions, we define a weaker notion of correctness, which states that it is computationally hard to find an input x which fails honest evaluation. In a bit more detail, an adversary is given a copy-protected program for the point function $f_{a,b}$. Firstly, if he uses this program to honestly evaluate $f_{a,b}$ on input a, then he will obtain output b and the program will not be destroyed. Secondly, if he does not have auxiliary information and he only uses $\mathsf{Eval}()$ to query $f_{a,b}$, then he will not come across an input that evaluates incorrectly except with small probability. We formalize this second condition as a correctness experiment.

Definition 5 (Computational Correctness)**.** *A copy-protection scheme* $(\mathsf{CopyProtect}, \mathsf{Eval})$ *for a class of point functions* $\mathcal{F} = \{f_{a,b} : (a,b) \in X \times Y\}$*, where* $X = \{0,1\}^{\mathrm{poly}(\lambda)}$ *and* $Y = \{0,1\}^{\mathrm{poly}(\lambda)}$*, satisfies computational* $(\varepsilon(\lambda), \delta(\lambda))$*-correctness with respect to a probability distribution* $\mathcal{D}$ *over* $\mathcal{F}$ *if:*

- *For any* $f_{a,b} \in \mathcal{F}$*, we have:*

$$\Pr[\rho' \otimes |b\rangle\langle b| \leftarrow \mathsf{Eval}(1^\lambda, \rho, a) \wedge T(\rho, \rho') \leq \varepsilon(\lambda) : \rho \leftarrow \mathsf{CopyProtect}(1^\lambda, (a,b))] = 1,$$

  where $T(\cdot,\cdot)$ *denotes trace distance.*

- *No QPT adversary* $\mathcal{A}$ *can succeed in the following correctness experiment with probability greater than* $\delta(\lambda)$*:*
  - *The challenger samples* $f_{a,b} \leftarrow \mathcal{D}$ *and computes* $\rho_f^{(0)} \leftarrow \mathsf{CopyProtect}(1^\lambda, (a,b))$*.*
  - *For* $i = 0, 1, \ldots, \mathrm{poly}(\lambda)$*;* $\mathcal{A}$ *sends an adaptive query* $x_i \in X$ *to the challenger, who computes* $\rho_f^{(i+1)} \otimes |y_i\rangle\langle y_i| \leftarrow \mathsf{Eval}(1^\lambda, \rho_f^{(i)}, x_i)$ *and sends* $y_i \in Y$ *back to* $\mathcal{A}$*.*
  - $\mathcal{A}$ *wins if there exists an index* $i \in \{0, 1, \ldots, \mathrm{poly}(\lambda)\}$ *such that* $x_i \neq a$ *and* $y_i \neq 0$*.*

Remark 2. *To give more context on this definition, imaigne a scenario where a software firm (Alice) provides a copy-protected program* ρ_f *to a client (Bob). Computational correctness guarantees that if the client follows the instructions provided by* $\mathcal{A}$*, that is, if he only uses* ρ_f *as an input to the algorithm* $\mathsf{Eval}()$*, then he will get the correct output with overwhelming probability. This is true even if* ρ_f *changes greatly after Bob evaluates the function* f*. However,* ρ_f *has no reusability guarantee once Bob uses third party programs that modify* ρ_f*. Our definition is closely related to the notion of "computational functionality preservation" defined in [BLMZ19] in the context of classical virtual-black-box obfuscation, which states given an obfuscated program, a PPT adversary cannot find an input which evaluates incorrectly. Note that the issue of the program being destroyed is specific to the quantum setting.*

Remark 3. *Computational correctness is stronger than distributional correctness defined in [BJL+21], which states that honest evaluation yields the correct output with probability close to 1, when the input is sampled from some distribution over the input space, as long as the distribution is efficiently samplable (in particular the uniform distribution). The reason is simple: a QPT adversary can sample the query input from this distribution.*

Definition 6 (Copy-Protection Security for Point Functions)**.** *A copy-protection scheme* $(\mathsf{CopyProtect}, \mathsf{Eval})$ *for a class of point functions* $\mathcal{F} = \{f_{a,b} : a \in X, b \in Y\}$ *is called* ***secure*** *if it is* $\frac{1}{2}$*-secure with respect to*

$\left\{\mathcal{D}_X^f\right\}_{f\in\mathcal{F}}$, *where*

$$\Pr[x = a : x \leftarrow \mathcal{D}_X^{f_{a,b}}] = \frac{1}{2},$$
$$\Pr[x = a'] = \frac{1}{2|X| - 2}$$

for all $f_{a,b} \in \mathcal{F}$ *and* $a' \neq a$*. That is,* $\mathcal{D}_X^{f_{a,b}}$ *samples* a *with probability* $1/2$ *and every other* $a' \neq a$ *with equal probability.*

Note that an adversary can trivially succeed in the pirating experiment for point functions with probability $1/2$ by always outputting 0 in both registers.

3 Private-Key and Public-Key Uncloneable Encryption: Definition

We present the definitions of public-key and private-key uncloneable encryptions, satisfying reusable security. Before we present these definitions, we first recall the definition of one-time uncloneable encryption.

3.1 One-Time Uncloneable Encryption

The following definitions were introduced by [BL20] in the context of quantum encryption of classical messages (QECMs). A one-time uncloneable encryption scheme otUE is a tuple of QPT algorithms $(\mathsf{otUE.Setup}, \mathsf{otUE.Enc}, \mathsf{otUE.Dec})$:

- **Setup,** $\mathsf{otUE.Setup}(1^\lambda)$: on input the security parameter λ, it outputs a key $k \in \mathcal{K}$.
- **Encryption,** $\mathsf{otUE.Enc}(k, m)$: on input a the key k and a message $m \in \mathcal{M}$, it outputs a ciphertext $\mathsf{CT} \in \mathcal{H}_{\mathcal{CT}}$.
- **Decryption,** $\mathsf{otUE.Dec}(k, \mathsf{CT})$: on input a key $k \in \mathcal{K}$ and a ciphertext $\mathsf{CT} \in \mathcal{H}_{\mathcal{CT}}$, it outputs a message $m' \in \mathcal{M}$.

Correctness: otUE should satisfy statistical correctness, i.e. for any key $k \in \mathcal{K}$ and any message $m \in \mathcal{M}$ we have

$$\Pr\left[m' = m : |\mathsf{CT}\rangle\langle\mathsf{CT}| \leftarrow \mathsf{otUE.Enc}(k, m),\ |m'\rangle\langle m'| \leftarrow \mathsf{otUE.Dec}(k, \mathsf{CT})\right] \geq 1 - \mathsf{negl}(\lambda).$$

Security: We require a one-time uncloneable encryption scheme to satisfy two security properties: firstly, it is a one-time pad and secondly, it needs to satisfy uncloneable security. We give the formal definitions below:

Definition 7 ((One-Time) Indistinguishability Security)**.** *We say that a QECM is indistinguishable if for any messages* $m_1, m_2 \in \mathcal{M}$ *of equal length, the following holds:*

$$\{\mathsf{Enc}(k, m_1)\} \approx_c \{\mathsf{Enc}(k, m_2)\},$$

where $k \leftarrow \mathsf{Setup}(1^\lambda)$.

Definition 8 (Uncloneable Security)**.** *We say that a QECM with message length n is t-uncloneable secure if a QPT cloning adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ cannot succeed with probability more than $2^{-n+t} + \mathsf{negl}(\lambda)$ in the cloning experiment defined below:*

Cloning Experiment: *The cloning experiment consists of two phases:*

- *In phase 1, the challenger samples a key $k \leftarrow \mathsf{Setup}(1^\lambda)$ and a message $m \in \mathcal{M}$ uniformly at random. He then computes ρ_{CT} and sends it to $\mathcal{A}$, who applies to ρ_{CT} a CPTP map $\phi : \mathcal{D}(\mathcal{H}_A) \to \mathcal{D}(\mathcal{H}_B) \otimes \mathcal{D}(\mathcal{H}_C)$ to obtain the bipartite state ρ_{BC}. She sends the B (resp., C) register of this state to $\mathcal{B}$ (resp., $\mathcal{C}$).*
- *In phase 2, $\mathcal{B}$ and $\mathcal{C}$ are not allowed to communicate. The key k is revealed to both of them. Then, $\mathcal{B}$ (resp., $\mathcal{C}$) applies a POVM B^k (resp., POVM C^k) to their register to measure and output a message m_B (resp., m_C).*
- *The adversary wins iff $m_B = m_C = m$.*

Below is a stronger notion of security which implies both Definition 7 and Definition 8, which is called uncloneable-indistinguishable security[9].

Definition 9 (Uncloneable-Indistinguishable Security)**.** *We say that a QECM with message length n is uncloneable-indistinguishable secure if a QPT cloning-distinguishing adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ cannot succeed with probability more than $1/2 + \mathsf{negl}(\lambda)$ in the cloning-distinguishing experiment defined below:*

Cloning-Distinguishing Experiment: *The cloning experiment consists of two phases:*

- *In phase 1, $\mathcal{A}$ chooses two messages $m_0, m_1 \in \mathcal{M}$ and sends them to the challenger. The challenger samples a key $k \leftarrow \mathsf{Setup}(1^\lambda)$ and a bit b uniformly at random. The challenger then computes $\rho \leftarrow \mathsf{Enc}(k, m_b)$ and sends ρ to $\mathcal{A}$.*
- *In phase 2, $\mathcal{A}$ has a ciphertext ρ to which she applies a CPTP map $\phi : \mathcal{D}(\mathcal{H}_A) \to \mathcal{D}(\mathcal{H}_B) \otimes \mathcal{D}(\mathcal{H}_C)$ to split it into two registers (B, C). She then sends the B and C registers to $\mathcal{B}$ and $\mathcal{C}$, respectively.*
- *In phase 3, the key k is revealed to both $\mathcal{B}$ and $\mathcal{C}$. Then, $\mathcal{B}$ (resp., $\mathcal{C}$) applies a POVM B^k (resp., POVM C^k) to their register to measure and output a bit b_B (resp., b_C).*
- *The adversary wins iff $b_B = b_C = b$.*

[9]We slightly deviate from [BL20] in defining uncloneable-indistinguishable security. We have the adversary choose two messages whereas they require one of the messages to be a uniformly random message. We anticipate that the two definitions may be equivalent.

Instantiations. The work of Broadbent and Lord [BL20] presented two constructions of one-time uncloneable encryption, that is, constructions satisfyint Definition 7 and Definition 8. Their first construction, "conjugate encryption", which encrypts messages of constant length n, is information-theoretic and $n \log_2(1 + 1/\sqrt{2})$-uncloneable secure. This scheme upper-bounds the success probability of a cloning adversary by $1/2 + 1/2\sqrt{2} \approx 0.85$ in the single-bit message ($n = 1$) case.

The second construction, "$\mathcal{F}-$conjugate encryption", is based on computational assumptions. It uses post-quantum pseudo-random functions but is only shown to be secure in the random oracle model. Nonetheless, it satisfies multi-message security and $\log_2(9)$-uncloneable security for long messages. Their analysis for this scheme does not provide an uncloneability bound for the single-bit message case.

There is no known construction of an uncloneable-indistinguishable secure scheme that we know of. For instance, no qubit-wise encryption scheme, including conjugate-encryption and generalized conjugate encryption (Section 5.2) can satisfy that definition for messages of length $n \geq 2$, since the cloning-distinguishing adversary can send one half of the ciphertext to $\mathcal{B}$ and the other half to C.

Conjugate Encryption Upper and Lower Bounds [BL20] shows that in their conjugate encryption scheme a cloning adversary can succeed with probability at most $(1/2 + 1/2\sqrt{2})^n$, which is based on BB84 monogamy-of-entanglement (MOE) game analyzed in [TFKW13]. Their proof technique can be generalized to a class of MOE games, which we call *real-orthogonal monogamy games*, to potentially obtain better security in the event that a monogamy game with a better value exists in this class.

Arbitrary pure single-qubit states on the xz plane of the Bloch Sphere can be cloned with fidelity $f := (1/2 + 1/2\sqrt{2}) \approx 0.85$ [BCMDM00]. Since every ciphertext lies on the xz plane in conjugate encryption, a cloning adversary (for each qubit) clone the ciphertext with fidelity f. In phase 2, both $\mathcal{B}$ and C will decrypt their register, hence each having fidelity f to the message $|m\rangle\langle m|$. By union bound, this implies that they both output m with probability at least $(2f-1)^n \approx 0.7^n$. In the single-bit message case, this means that the scheme of [BL20] can be violated by an adversary with probability 0.7. For details of these upper-lower bounds, see Section 5.2 and Section 6.1.

3.2 Private-Key Uncloneable Encryption

To present the definition of a private-key uncloneable encryption scheme, we first recall the semantic security definition of a private-key encryption scheme.

Definition 10 (Semantic Security)**.** *A private-key encryption scheme* (Setup, Enc, Dec) *is said to satisfy semantic security if it satisfies the following property: for sufficiently large $\lambda \in \mathbb{N}$, for every $(m_1^{(0)}, \ldots, m_q^{(0)})$, $(m_1^{(1)}, \ldots, m_q^{(1)})$ such that $|m_i^{(0)}| = |m_i^{(1)}|$ for every $i \in [q]$ and $q = \mathrm{poly}(\lambda)$,*

$$\left\{\mathsf{Enc}\left(k, m_1^{(0)}\right), \ldots, \mathsf{Enc}\left(k, m_q^{(0)}\right)\right\} \approx_c \left\{\mathsf{Enc}\left(k, m_1^{(1)}\right), \ldots, \mathsf{Enc}\left(k, m_q^{(1)}\right)\right\},$$

where $k \leftarrow \mathsf{Setup}(1^\lambda)$.

We define a private-key uncloneable encryption scheme below.

Definition 11. *A private-key uncloneable encryption scheme, consists of a tuple of algorithms* (Setup, Enc, Dec)*, and satisfies the properties of (reusable) semantic security (see above) and uncloneable security (Definition 8).*

3.3 Public-Key Uncloneable Encryption.

To present the definition of a public-key uncloneable encryption scheme, we first recall the semantic security definition of a public-key encryption scheme below.

Definition 12 (Semantic Security)**.** *A public-key encryption scheme* (Setup, Enc, Dec) *is said to satisfy semantic security property if the following holds: for sufficiently large* $\lambda \in \mathbb{N}$*, for every* m_0, m_1 *of equal length,*

$$\{\mathsf{Enc}(\mathsf{PK}, m_0)\} \approx_c \{\mathsf{Enc}(\mathsf{PK}, m_1)\},$$

the distinguisher also receives as input PK*, where* PK *is such that* $(\mathsf{PK}, \mathsf{SK}) \leftarrow \mathsf{Setup}(1^\lambda)$.

We now present the definition of a public-key uncloneable encryption scheme.

Definition 13. *A public-key uncloneable encryption scheme, consists of a tuple of algorithms* (Setup, Enc, Dec)*, where* Setup, Enc *are defined below and* Dec *is defined as in a private-key uncloneable encryption scheme:*

- $\mathsf{Setup}(1^\lambda)$*: on input the security parameter* λ*, output a public key* PK *and a secret key* SK*.*
- $\mathsf{Enc}(\mathsf{PK}, m)$*: on input a public key* PK*, message* m*, output a ciphertext* CT*.*

A public-key uncloneable encryption scheme needs to satisfy the definitions of semantic security (see above) and uncloneable security (Definition 8).

For a construction of public-key encryption using functional encryption, see Section 5.

4 Private-Key Uncloneable Encryption (PK-UE)

We present a construction of (reusable) private-key uncloneable encryption in this section. One of the tools required in our construction is a private-key encryption with fake-key property. We first define and construct this primitive.

4.1 Private-Key Encryption with Fake-Key Property

We augment the traditional notion of private-key encryption with a property, termed as fake-key property. This property allows an authority to issue a fake decryption key fk, as a function of m along with an encryption of m, denoted by CT, in such a way that a QPT distinguisher will not be able to distinguish whether it received the real decryption key or a fake decryption key. A consequence of this definition is that, the decryption algorithm on input the fake decryption key fk and CT should yield the message m.

Definition 14 (Fake-Key Property)**.** *We say that a classical encryption scheme* (Setup, Enc, Dec) *satisfies the "**fake-key property**" if there exists a polynomial time algorithm* $FakeGen : \mathcal{CT} \times \mathcal{M} \to \mathcal{K}$ *such that for any* $m \in \mathcal{M}$*,*

$$\{(ct^m \leftarrow \mathsf{Enc}(k, m), k)\} \approx_c \{\big(ct^0 \leftarrow \mathsf{Enc}(k, 0), fk \leftarrow FakeGen(ct^0, m)\big)\}, \tag{1}$$

where $k \leftarrow \mathsf{Setup}(1^\lambda)$*.*

Note that in particular, the fake-key property requires that $\mathsf{Dec}(fk, ct^0) = m$.

Theorem 5. *Assuming the existence of post-quantum pseudorandom functions, there exists a classical private-key encryption scheme (PKE) that satisfies the fake-key property.*

Proof. Let $\{PRF_k : \{0,1\}^\ell \to \{0,1\}^n \ : \ k \in \{0,1\}^\lambda\}$ be a class of post-quantum pseudo-random functions, where ℓ is set to be λ and n is the length of the messages encrypted.

Consider the following scheme:

- **Setup,** $\mathsf{Setup}(1^\lambda)$**:** on input λ, it outputs (k, otp), where $k \leftarrow \{0,1\}^\lambda$ and $otp \leftarrow \{0,1\}^n$ are uniformly sampled.
- **Encryption,** $\mathsf{Enc}((k, otp), m)$**:** on input key (k, otp), message $m \in \{0,1\}^n$, it outputs $ct = (ct_1, ct_2)$, where $\mathsf{CT}_1 = r$ and $ct_2 = PRF_k(r) \oplus m \oplus otp$ with $r \leftarrow \{0,1\}^\ell$ being uniformly sampled.
- **Decryption,** $\mathsf{Dec}((k, otp), ct)$**:** on input (k, otp), ciphertext ct parsed as (ct_1, ct_2), output μ, where $\mu = ct_2 \oplus PRF_k(ct_1) \oplus otp$.
- **Fake Key Generation,** $FakeGen(ct^0, m)$**:** on input ciphertext ct^0 parsed as (ct_1^0, ct_2^0), message m, it outputs the fake decryption key $fk = (k', otp')$, where $k' \leftarrow \{0,1\}^\lambda$ is uniformly sampled and $otp' = ct_2^0 \oplus PRF_{k'}(ct_1^0) \oplus m$.
  // Note: this choice of otp' *yields* $\mathsf{Dec}((k', otp'), ct^0) = m$*.*

Correctness and Semantic Security: Correctness can easily be checked. Semantic security follows from the security of pseudorandom functions using a standard argument.

Fake-Key Property. Note that given $\{ct, (k, otp)\} \in \mathcal{C} \times \mathcal{K}$, one can perform the reversible operation:

$$\big\{(ct_1, ct_2), (k, otp)\big\} \longrightarrow \big\{(ct_1, ct_2 \oplus otp \oplus PRF_k(r)), (k, otp)\big\}\,.$$

Thus, the fake-key property (eq. (1)) can be rewritten as:

$$
\begin{aligned}
& \{\left(ct^m \leftarrow \mathsf{Enc}((k, otp), m), k\right)\} \approx_c \{\left(ct^0 \leftarrow \mathsf{Enc}((k, otp), 0), fk \leftarrow FakeGen(ct^0, m)\right)\} \\
\iff & \{(r, PRF_k(r) \oplus m \oplus otp), (k, otp)\} \approx_c \{(r, PRF_k(r) \oplus otp), (k', otp')\} \\
\iff & \{(r, m), (k, otp)\} \approx_c \{(r, PRF_k(r) \oplus otp \oplus otp' \oplus PRF_{k'}(r)), (k', otp')\} \\
\iff & \{(r, m), (k, otp)\} \approx_c \{(r, m), (k', otp')\} \\
\iff & \{(r, m), (k, otp)\} \approx_c \{(r, m), (k', PRF_k(r) \oplus m \oplus otp)\} \\
\iff & \{(r, m), (k, otp)\} \approx_c \{(r, m), (k, PRF_{k'}(r) \oplus m \oplus otp)\}, \qquad (2)
\end{aligned}
$$

where in the last step we swapped k and k', which is allowed since they are independently sampled. Therefore, observing in eq. (2) that k doesn't occur in the second part of the key, the fake-key property reduces to the following:

$$\{((r, m), otp)\} \approx_c \{((r, m), otp \oplus m \oplus PRF_{k'}(r))\},$$

which follows[10] from the fact that otp is sampled independently from r, m, and k'.

□

4.2 Construction

We first describe the tools used in our construction of PK-UE scheme.

Tools. Let PKE be a post-quantum private-key encryption scheme with fake-key property (defined in Section 4.1) and let UE be a one-time uncloneable encryption scheme (defined in Section 3.1).

We present the construction of a PK-UE scheme below, which combines these tools such that it inherits semantic security from the first and uncloneability from the second.

Setup, $\mathsf{Setup}(1^\lambda)$**:** on input a security parameter λ, it outputs k_{PKE}, where $k_{PKE} \leftarrow \mathsf{PKE.Setup}(1^\lambda)$.

Encryption, $\mathsf{Enc}(k_{PKE}, m)$**:** on input a key k_{PKE}, message m, it first generates $k_{UE} \leftarrow \mathsf{UE.Setup}(1^\lambda)$ and outputs $ct = (ct_1, ct_2)$, where $ct_1 \leftarrow \mathsf{PKE.Enc}(k_{PKE}, k_{UE})$ and $ct_2 \leftarrow \mathsf{UE.Enc}(k_{UE}, m)$.

Decryption, $\mathsf{Dec}(k_{PKE}, ct)$**:** on input the decryption key k_{PKE}, ciphertext ct, it computes $\mu = \mathsf{UE.Dec}(k_{UE}, ct_2)$, where $k_{UE} = \mathsf{PKE.Dec}(k_{PKE}, ct_1)$. Output μ.

Correctness follows from the correctness of the uncloneable encryption scheme and the private-key encryption scheme. The semantic security follows from a standard hybrid argument and hence we omit the details; informally speaking, we first invoke the security of the underlying PKE scheme to replace the message under PKE to be 0 and then we invoke the indistinguishability security of UE

[10]Note that this proof in fact demonstrates perfect fake-key property, even though we only need computational fake-key property in our construction.

to replace the message m. We perform this for all the q messages, where $q = \text{poly}(\lambda)$ is the number of messages chosen by the adversary in the semantic security experiment.

4.2.1 Uncloneable Security

Suppose that for a parameter t, the proposed scheme is not t-uncloneable secure; meaning there exists an adversary A which breaks the corresponding cloning experiment (Hybrid 1) with probability $p = 2^{-n+t} + \frac{1}{poly(\lambda)}$. We define another experiment Hybrid 2, which we claim the adversary breaks with probability $p - \mathsf{negl}(\lambda)$.

Hybrid 1: This corresponds to the the cloning experiment of the above proposed PK-UE scheme.

Hybrid 2:

- In phase 1, the challenger samples $k_{PKE} \leftarrow \mathsf{PKE.Setup}(1^\lambda)$ and $k_{UE} \leftarrow \mathsf{UE.Setup}(1^\lambda)$, then sends $(ct^0 \leftarrow \mathsf{PKE.Enc}(k_{PKE}, 0), ct_2 \leftarrow \mathsf{UE.Enc}(k_{UE}, m))$ to the adversary $\mathcal{A}$, who then applies a CPTP map $\phi : \mathcal{D}(\mathcal{H}_A) \rightarrow \mathcal{D}(\mathcal{H}_B) \otimes \mathcal{D}(\mathcal{H}_C)$ to split it into two registers (B, C).
- In phase 2, the challenger reveals $fk \leftarrow FakeGen(ct^0, k_{UE})$ to both $\mathcal{B}$ and $\mathcal{C}$, who then need to output $m_B = m_C = m$ in order to win the experiment.

Claim 1. *If A wins in Hybrid 2 with probability p', then $|p - p'| = \mathsf{negl}(\lambda)$.*

Proof. Assume to the contrary that $|p - p'| \geq \frac{1}{\text{poly}(\lambda)}$. We will describe an adversary $\tilde{\mathcal{A}}$ which breaks the fake-key property of PKE.

Given (ct^*, k^*_{PKE}), $\tilde{\mathcal{A}}$ samples $k_{UE} \leftarrow \mathsf{UE.Setup}(1^\lambda)$, computes $ct^m \leftarrow \mathsf{UE.Enc}(k_{UE}, m)$ and sends (ct^*, ct^m) to A, who then applies a CPTP map $\phi : \mathcal{D}(\mathcal{H}_A) \rightarrow \mathcal{D}(\mathcal{H}_B) \otimes \mathcal{D}(\mathcal{H}_C)$ to split it into two registers (B, C). In phase 2, $\tilde{\mathcal{A}}$ reveals k^*_{PKE} to $\mathcal{B}$ and $\mathcal{C}$. Observe that depending on whether the key k^*_{PKE} is real or fake, we are either in Hybrid 1 or Hybrid 2. Hence, by assumption $\tilde{\mathcal{A}}$ can distinguish the two cases, breaking the fake-key property. □

Now that we know $\mathcal{A}$ breaks Hybrid 2 with probability at least $p - \mathsf{negl}(\lambda)$, we can construct an adversary $\tilde{\tilde{\mathcal{A}}}$ that breaks the uncloneability experiment of UE:

- In Phase 1, the challenger samples $k_{UE} \leftarrow \mathsf{UE.Setup}(1^\lambda)$ and sends $ct^m \leftarrow \mathsf{UE.Enc}(k_{UE}, m)$ to $\tilde{\tilde{\mathcal{A}}}$. Then, $\tilde{\tilde{\mathcal{A}}}$ samples $k_{PKE} \leftarrow \mathsf{PKE.Setup}(1^\lambda)$ and computes $ct^0 \leftarrow \mathsf{PKE.Enc}(k_{PKE}, 0)$. After that, $\tilde{\tilde{\mathcal{A}}}$ runs A on input (ct^0, ct^m) to obtain bipartite state $\rho_{BC} \in \mathcal{D}(\mathcal{H}_B) \otimes \mathcal{D}(\mathcal{H}_C)$, which she sends to $\tilde{\tilde{\mathcal{B}}}$ and $\tilde{\tilde{\mathcal{C}}}$. In addition, $\tilde{\tilde{\mathcal{A}}}$ samples a randomness r for the algorithm $\mathsf{PKE}.FakeGen()$ and sends r to both $\tilde{\tilde{\mathcal{B}}}$ and $\tilde{\tilde{\mathcal{C}}}$.
- In phase 2, the challenger reveals k_{UE} to both $\tilde{\tilde{\mathcal{B}}}$ and $\tilde{\tilde{\mathcal{C}}}$. Then, $\tilde{\tilde{\mathcal{B}}}$ runs $\mathcal{B}$ on his register[11], revealing fk as the key, to obtain and output m_B, where $fk \leftarrow FakeGen(ct^0, k_{UE})$ is sampled using randomness r. Similarly, $\tilde{\tilde{\mathcal{C}}}$ obtains and outputs m_C by running $\mathcal{C}$ on his register (C),

[11] That is, the B register of ρ_{BC}.

revealing fk as the key, where fk is generated using randomness r so that it matches what is generated by $\mathcal{B}$.

Because the view of the adversary $(\mathcal{A}, \mathcal{B}, \mathcal{C})$ run as a subprotocol in this experiment matches exactly that in Hybrid 2, we conclude that $\tilde{\tilde{\mathcal{A}}}$ breaks the uncloneability experiment of UE with probability p', meaning UE is not $t-$uncloneable secure.

Therefore, we just proved the following theorem.

Theorem 6. *If* UE *is* t*-uncloneable secure, then the proposed scheme is also* t*-uncloneable secure.*

Corollary 2. *The above proposed scheme is* $n \log_2(1+\frac{1}{\sqrt{2}})$*-uncloneable secure, where* n *is the message length.*

5 Public-Key Uncloneable Encryption

We now focus on constructing uncloneable encryption in the public-key setting using functional encryption. We adopt the Trojan technique of [ABSV15], proposed in a completely different context, to prove the uncloneability property.

We describe all the tools that we use in the scheme below.

Tools.

- A one-time uncloneable encryption scheme, denoted by $\mathsf{UE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$.
- A post-quantum secure symmetric-key encryption scheme with pseudorandom ciphertexts, denoted by $\mathsf{SKE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$. That is, this scheme has the property that the ciphertexts are computationally indistinguishable from the uniform distribution. Such a scheme can be constructed from one-way functions[12].
- A post-quantum secure single-key public-key functional encryption scheme, denoted by $\mathsf{FE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. Such a scheme can be instantiated using [SS10, GVW12]. See Section 2.4.

5.1 Construction

We denote the public-key uncloneabe encryption scheme that we construct as $\mathsf{PBKUE} = (\mathsf{PBKUE.Setup}, \mathsf{PBKUE.Enc}, \mathsf{PBKUE.Dec})$. We describe the algorithms below.

[12]The scheme is quite simple and presented in [Gol07]: suppose $\mathsf{PRF} : \{0,1\}^\lambda \to \{0,1\}^\ell$ is a pseudorandom function. To encrypt a message $x \in \{0,1\}^\ell$ using a symmetric key k, compute $(r, \mathsf{PRF}(k,r) \oplus x)$, where $r \xleftarrow{\$} \{0,1\}^\lambda$. From the security of pseudorandom functions, it follows that the ciphertext is computationally indistinguishable from the uniform distribution.

Setup, $\mathsf{Setup}(1^\lambda)$**:** on input a security parameter λ, compute $(\mathsf{FE.MSK}, \mathsf{FE.mpk}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$. Compute $\mathsf{FE}.sk \leftarrow \mathsf{FE.KeyGen}(\mathsf{FE.MSK}, F[ct])$, where $ct \xleftarrow{\$} \{0,1\}^{\mathrm{poly}(\lambda)}$ and $F[ct]$ is the following function:

$$F[ct](b, K, m) = \begin{cases} Dec(K, ct) & \text{if } b = 0, \\ m, & \text{otherwise} \end{cases}$$

Set the secret key to be $k = \mathsf{FE}.sk$ and the public key to be $pk = \mathsf{FE.mpk}$.

Encryption, $\mathsf{Enc}(pk, m)$**:** on input key pk, message m, it first generates $k_{UE} \leftarrow \mathsf{UE.Setup}(1^\lambda)$, and outputs $ct = (ct_1, ct_2)$, where $ct_1 \leftarrow \mathsf{FE.Enc}(\mathsf{FE.mpk}, (1, \perp, k_{UE})$ and $ct_2 \leftarrow \mathsf{UE}.Enc(k_{UE}, m)$.

Decryption, $\mathsf{Dec}(k, ct)$**:** On input k, ciphertext $ct = (ct_1, ct_2)$, first compute $\mathsf{FE.Dec}(\mathsf{FE}.sk, ct_1)$ to obtain k^*_{UE}. Then, compute $\mathsf{UE}.Dec(k^*_{UE}, ct_2)$ to obtain m^*. Output m^*.

The correctness follows from the correctness of the underlying UE and FE schemes. As in the private-key setting, the semantic security follows by a standard argument and hence, we omit the details.

5.1.1 Uncloneable Security

We show that our construction achieves the same uncloneable security as the underlying one-time scheme UE. Formally, we prove the following theorem.

Theorem 7. *If* UE *is t-uncloneable secure, then* PBKUE *is also t-uncloneable secure.*

Proof. Suppose that there exists an adversary $(\mathcal{A}, \mathcal{B}, C)$ which succeeds in the cloning experiment of PBKUE with probability $p = 2^{-n+t} + \frac{1}{\mathrm{poly}(\lambda)}$. Through a sequence of hybrid experiments, we will construct an adversary which breaks the t-uncloneability of UE.

Hybrid 1: This corresponds to the cloning experiment of PBKUE.

Hybrid 2: Same as Hybrid 1, except ct in $\mathsf{PBKUE.Setup}()$, instead of being randomly sampled, is generated as $ct \leftarrow \mathsf{SKE.Enc}(k_{SKE}, k_{UE})$, where $k_{SKE} \leftarrow \mathsf{SKE.Setup}(1^\lambda)$.

Claim 2. $(\mathcal{A}, \mathcal{B}, C)$ *succeeds in Hybrid 2 with probability at least* $p - \mathsf{negl}(\lambda)$.

Proof. Hybrids 1 and 2 are computationally indistinguishable by the pseudorandom ciphertext property of SKE. Indeed, an adversary given a random text r or a real ciphertext ct can run the cloning experiment with $(\mathcal{A}, \mathcal{B}, C)$ to distinguish both the hybrids, hence distinguishing r and ct. □

Hybrid 3: Same as Hybrid 2, except ct_1 in $\mathsf{PBKUE.Enc}()$, is generated as $ct_1 \leftarrow \mathsf{FE.Enc}(\mathsf{FE.mpk}, (0, k_{SKE}, \perp))$.

Claim 3. $(\mathcal{A}, \mathcal{B}, C)$ *succeeds in Hybrid 3 with probability at least* $p - \mathsf{negl}(\lambda)$.

Proof. Hybrids 2 and 3 are indistinguishable by the (selective) security of FE. Indeed, suppose that Hybrids 2 and 3 can be distinguished by $(\mathcal{A}, \mathcal{B}, C)$, and consider the following adversary $\mathcal{A}'$ which breaks the (selective) security of FE:

- The challenger runs $(\mathsf{FE.mpk}, \mathsf{FE.MSK}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$.
- $\mathcal{A}'$ runs $k_{UE} \leftarrow \mathsf{UE.Setup}(1^\lambda)$ and $k_{SKE} \leftarrow \mathsf{SKE.Setup}(1^\lambda)$, then sets $m_0 = (1, \perp, k_{UE})$ and $m_1 = (0, k_{SKE}, \perp)$. Then, $\mathcal{A}'$ sends (m_0, m_1) to the challenger.
- The challenger chooses a random bit b sends back $\mathsf{FE.mpk}$ and $ct_1^b \leftarrow \mathsf{FE.Enc}(\mathsf{FE.mpk}, m_b)$.
- $\mathcal{A}'$ implements the function $\tilde{f} := F[\mathsf{SKE.Enc}(k_{SKE}, k_{UE})]$ and makes a query to the challenger to receive $\mathsf{FE}.sk \leftarrow \mathsf{FE.KeyGen}(\mathsf{FE.MSK}, \tilde{f})$. This query is valid since $\tilde{f}(m_0) = \tilde{f}(m_1) = k_{UE}$.
- Now $\mathcal{A}'$ can perform a simulation, which matches Hybrid 2 with adversary $(\mathcal{A}, \mathcal{B}, C)$ when $b = 0$, and Hybrid 3 with adversary $(\mathcal{A}, \mathcal{B}, C)$ when $b = 1$. This will let $\mathcal{A}'$ to distinguish the cases $b = 0$ and $b = 1$, breaking FE security. After sampling a random message $m \leftarrow \{0,1\}^n$, $\mathcal{A}'$ has everything she needs to perform the simulation. Note that even though she doesn't know $\mathsf{FE.MSK}$, she has learned $\mathsf{FE}.sk$, which is the only time $\mathsf{FE.MSK}$ is used.

□

Having established that $(\mathcal{A}, \mathcal{B}, C)$ succeeds in Hybrid 3 with probability $p - \mathsf{negl}(\lambda)$, we will now construct an adversary $(\tilde{\tilde{\mathcal{A}}}, \tilde{\tilde{\mathcal{B}}}, \tilde{\tilde{C}})$ that succeeds in the cloning experiment of UE with probability $p - \mathsf{negl}(\lambda)$, contradicting the t-uncloneable security:

- The challenger samples $k_{UE} \leftarrow \mathsf{UE.Setup}(\lambda)$ and $m \leftarrow \{0,1\}^n$, then sends $ct_2 \leftarrow \mathsf{UE.Enc}(k_{UE}, m)$ to $\tilde{\tilde{\mathcal{A}}}$.
- In Phase 1, $\tilde{\tilde{\mathcal{A}}}$ samples $(\mathsf{FE.MSK}, \mathsf{FE.mpk}) \leftarrow \mathsf{FE.Setup}(1^\lambda)$ and $k_{SKE} \leftarrow \mathsf{SKE.Setup}(1^\lambda)$. She then computes $ct_1 \leftarrow \mathsf{FE.Enc}(\mathsf{FE.mpk}, (0, k_{SKE}, \perp))$. At the end of the phase $\tilde{\tilde{\mathcal{A}}}$ runs $\mathcal{A}$ on input $ct^* = (ct_1, ct_2)$ to have $\tilde{\tilde{\mathcal{B}}}$ and $\tilde{\tilde{C}}$ receive bipartite state $\rho_{BC} \in \mathcal{D}(\mathcal{H}_B) \otimes \mathcal{D}(\mathcal{H}_C)$. $\tilde{\tilde{\mathcal{A}}}$ also samples a random string r for $\mathsf{SKE.Enc}$ and sends a copy of r attached to the corresponding registers to both $\tilde{\tilde{\mathcal{B}}}$ and $\tilde{\tilde{C}}$.
- In Phase 2, the challenger reveals k_{UE} to both $\tilde{\tilde{\mathcal{B}}}$ and $\tilde{\tilde{C}}$. $\tilde{\tilde{\mathcal{B}}}$ computes $ct \leftarrow \mathsf{SKE.Enc}(k_{SKE}, k_{UE})$ (using randomness r), and $\mathsf{FE}.sk \leftarrow \mathsf{FE.KeyGen}(\mathsf{FE.MSK}, F[ct])$. Then, he runs $\mathcal{B}$ on the B register of ρ_{BC}, revealing $\mathsf{FE}.sk$) as the key, to obtain output m_B, which he outputs as is. Similarly, $\tilde{\tilde{C}}$ runs C to obtain and output m_C.

Described above, $(\tilde{\tilde{\mathcal{A}}}, \tilde{\tilde{\mathcal{B}}}, \tilde{\tilde{C}})$ perfectly simulates the challenger of Hybrid 3 against $(\mathcal{A}, \mathcal{B}, C)$. Therefore, the success probability of $(\tilde{\tilde{\mathcal{A}}}, \tilde{\tilde{\mathcal{B}}}, \tilde{\tilde{C}})$ is $p - \mathsf{negl}(\lambda)$.

□

5.2 Generalized Conjugate Encryption

The conjugate encryption scheme of [BL20] uses the BB84 monogamy-of-entanglement (MOE) game studied in [TFKW13]. The success probability of a cloning adversary exactly equals that of a MOE adversary restricted in state preparation. In this section we make the obesrvation that their proof easily extends to a class of uncloneable encryption schemes based on a class of MOE games, which we define below:

Definition 15 (Real Orthogonal Basis)**.** *Let $(|x\rangle\langle x|)_{x\in X}$ be the standard basis for $\mathcal{D}(\mathcal{H}_X)$, with $X = \{0, 1, \ldots, \dim \mathcal{H}_X - 1\}$. An orthonormal basis $\beta = (|\psi_x\rangle\langle\psi_x|)_{x\in X}$ for $\mathcal{D}(\mathcal{H}_X)$ is called **real orthogonal** if there exist real coefficients $\{\alpha_{xx'}\}_{x,x'\in X}$ such that*

$$|\psi_x\rangle = \sum_{x'\in X} \alpha_{xx'}|x'\rangle$$

for all $x \in X$.

The following lemma, which is the main fact used to generalize conjugate encryption, states that an EPR pair defined in a real-orthogonal basis does not depend on the basis. It follows easily by properties of orthogonal matrices.

Lemma 3. *If $\beta = (|\psi_x\rangle\langle\psi_x|)_{x\in X}$ is a real orthogonal basis, then*

$$\sum_{x\in X} |\psi_x\psi_x\rangle = \sum_{x\in X} |xx\rangle \tag{3}$$

and hence

$$\sum_{x,x'\in X} |x\rangle\langle x'| \otimes |x\rangle\langle x'| = \sum_{x,x'\in X} |\psi_x\rangle\langle\psi_{x'}| \otimes |\psi_x\rangle\langle\psi_{x'}|$$

by taking the outer product of each side by itself in eq. (3).

Proof. By definition of a real orthogonal basis, the basis transition matrix $M = (\alpha_{xx'})_{x,x'\in X}$ is an orthogonal matrix, and so is M^T. Thus, the columns of M like its rows form an orthonormal basis, meaning

$$\sum_{x\in X} \alpha_{xx'}\alpha_{xx''} = \delta_{x'x''} \tag{4}$$

for all $x', x'' \in X$. Hence,

$$\begin{aligned}\sum_{x\in X} |\psi_x \psi_x\rangle &= \sum_{x\in X}\left(\sum_{x'\in X} \alpha_{xx'}|x'\rangle\right)\left(\sum_{x''\in X} \alpha_{xx''}|x''\rangle\right)\\ &= \sum_{x,x',x''\in X} \alpha_{xx'}\alpha_{xx''}|x'x''\rangle\\ &= \sum_{x',x''\in X} \delta_{x'x''}|x'x''\rangle\\ &= \sum_{x\in X} |xx\rangle\end{aligned}$$

□

Corollary 3. *If $X = \{0,1\}^n$ and $\beta = (|\psi_x\rangle\langle\psi_x|)_{x\in X}$ is a real orthogonal basis for $\mathcal{D}(\mathcal{H}_X)$, then*

$$|\mathsf{EPR}_n\rangle\langle\mathsf{EPR}_n| = \sum_{x,x'\in X} |\psi_x\rangle\langle\psi_{x'}| \otimes |\psi_x\rangle\langle\psi_{x'}|.$$

Definition 16 (Real-Orthogonal Monogamy Game)**.** *Let $X = \{0,1\}^n$. A real-orthogonal monogamy game (ROMG) $\mathcal{G}$ of order n is defined by the Hilbert space $\mathcal{H}_A$ of n-qubit states and a collection of real orthogonal bases $\left(\beta^\theta = \left(|\psi_x^\theta\rangle\langle\psi_x^\theta|\right)_{x\in X}\right)_{\theta\in\Theta}$. An adversary for $\mathcal{G}$ is defined by finite-dimensional Hilbert states $\mathcal{H}_B$ and $\mathcal{H}_C$, a tripartite state $\rho_{ABC} \in \mathcal{D}(\mathcal{H}_A)\otimes\mathcal{D}(\mathcal{H}_B)\otimes\mathcal{D}(\mathcal{H}_C)$, along with two collections of POVMs: $\left(\left(B_x^\theta\right)_{x\in X}\right)_{\theta\in\Theta}$ and $\left(\left(C_x^\theta\right)_{x\in X}\right)_{\theta\in\Theta}$. The value of $\mathcal{G}$, denoted by p_G, is the maximimum value the following expression can take for an optimal adversary:*

$$p_{win} = \frac{1}{|\Theta|}\sum_{\theta\in\Theta} \mathsf{Tr}\left(\Pi^\theta \rho_{ABC}\right),$$

so that

$$p_G = \max_{\substack{\rho_{ABC}\in\mathcal{D}(\mathcal{H}_A)\otimes\mathcal{D}(\mathcal{H}_B)\otimes\mathcal{D}(\mathcal{H}_C)\\ \left(\left(B_x^\theta\right)_{x\in X}\right)_{\theta\in\Theta}\\ \left(\left(C_x^\theta\right)_{x\in X}\right)_{\theta\in\Theta}}} p_{win},$$

where

$$\Pi^\theta = \sum_{x\in X} |\psi_x^\theta\rangle\langle\psi_x^\theta| \otimes B_x^\theta \otimes C_x^\theta.$$

p_{win} is the probability that $\mathcal{B}$ and $\mathcal{C}$ (the adversary) win in a monogamy game where:

- *$\mathcal{B}$ and $\mathcal{C}$, who are far away from each other, prepare a tripartite state ρ_{ABC} and send the A register to $\mathcal{A}$. $\mathcal{B}$ keeps the B register and $\mathcal{C}$ keeps the C register of this state.*
- *$\mathcal{A}$ samples $\theta \in \Theta$ uniformly at random and measures her register in basis β^θ to obtain $x \in X$. She then sends θ to both $\mathcal{B}$ and $\mathcal{C}$.*

- *$\mathcal{B}$ and C guess the value x, and they win if they are both correct.*

Theorem 8. *Let $\mathcal{G}$ be a ROMG of order n with value $p_G = 2^{-n+t} + \mathsf{negl}(\lambda)$, then there exists an uncloneable encryption scheme otUE_G with (constant) message length n, which is t-uncloneable secure.*

Proof. We will construct otUE_G such that the success probability of a cloning adversary equals that of a ROMG adversary, which is bounded by p_G. The same construction and analysis is done by [*BL*20] for the case of conjugate encoding [Wie83], where G is the BB84 game and (β^θ) are the Wiesner bases.[13][14]

Setup: On input security parameter 1^λ, Setup uniformly samples a key $(\theta, r) \leftarrow \Theta \times \{0,1\}^\lambda$.

Encryption: On input $m \in \mathcal{M}$ and $(\Theta, r) \in \mathcal{K}$, Enc outputs the pure state $\rho = |\psi^\theta_{(m\oplus r)}\rangle\langle\psi^\theta_{(m\oplus r)}|$.

Decryption: On input ciphertext ρ_{ct} and key (θ, r), Dec measures ρ_{ct} in the basis β^θ to obtain x, then outputs $x \oplus r$.

Indistinguishable Security: It suffices to show that for any message, the view of an adversary with no knowledge of the key (θ, r) equals the completely mixed state, which can easily be done as

$$\frac{1}{2^\lambda|\Theta|}\sum_{\theta,r}|\psi^\theta_{(m\oplus r)}\rangle\langle\psi^\theta_{(m\oplus r)}| = \mathbb{E}_\theta\frac{1}{2^\lambda}\sum_x|\psi^\theta_x\rangle\langle\psi^\theta_x| = \mathbb{E}_\theta(\mathbf{id}/2^\lambda) = (\mathbf{id}/2^\lambda).$$

t-uncloneable security: Let $(\mathcal{A}, \mathcal{B}, C)$ be a cloning adversary which uses the splitting CPTP map $\phi : \mathcal{D}(\mathcal{H}_A) \to \mathcal{D}(\mathcal{H}_B) \otimes \mathcal{D}(\mathcal{H}_C)$ as well as POVMs $\left(B^{(\theta,r)}\right)_{\theta,r\in\{0,1\}^\lambda}$ and $\left(C^{(\theta,r)}\right)_{\theta,r\in\{0,1\}^\lambda}$. We will construct a ROMG adversary $\mathcal{A}'$ for $\mathcal{G}$ that succeeds with the same probability. It uses the same Hilbert spaces $\mathcal{H}_B$, and $\mathcal{H}_C$, and it uses POVMs $(B')^\theta, (C')^\theta$, defined as

$$(B')^\theta_x = \frac{1}{2^\lambda}\sum_{r\in\{0,1\}^\lambda}B^{(\theta,r)}_{x\oplus r}, \quad (C')^\theta_x = \frac{1}{2^\lambda}\sum_{r\in\{0,1\}^\lambda}C^{(\theta,r)}_{x\oplus r}.$$

[13]The BB84 game of order n is defined as follows: $\beta^\theta = \left(|\psi^\theta_x\rangle\langle\psi^\theta_x|\right)_{x\in X}$, where

$$|\psi^\theta_x\rangle = \bigotimes_{j=1}^n H^{\theta_j}|x_j\rangle$$

and H denotes the single-qubit Hadamard gate.

[14]In [BL20], conjugate encryption is defined as having message length $n = \lambda$. We present n to be a constant instead so that in the definition of t-uncloneable security, the winning probability of a cloning adversary, which is negligible in n, is not negligible in λ.

Finally, the tripartite state ρ_{ABC} is defined below using Corollary 3:

$$\begin{aligned}
\rho_{ABC} &= \left(\mathbf{id}_A \otimes \phi\right) |\mathsf{EPR}_\lambda\rangle\langle\mathsf{EPR}_\lambda| \\
&= \left(\mathbf{id}_A \otimes \phi\right) \frac{1}{2^\lambda} \sum_{s,u\in\{0,1\}^\lambda} |s\rangle\langle u| \otimes |s\rangle\langle u| \\
&= \left(\mathbf{id}_A \otimes \phi\right) \frac{1}{2^\lambda} \sum_{s,u\in\{0,1\}^\lambda} |\psi_s^\theta\rangle\langle\psi_u^\theta| \otimes |\psi_s^\theta\rangle\langle\psi_u^\theta| \\
&= \frac{1}{2^\lambda} \sum_{s,u\in\{0,1\}^\lambda} |\psi_s^\theta\rangle\langle\psi_u^\theta| \otimes \phi\left(|\psi_s^\theta\rangle\langle\psi_u^\theta|\right).
\end{aligned}$$

The success probability of $\mathcal{A}'$ is then given by

$$\begin{aligned}
p_G &\geq \frac{1}{|\Theta|} \sum_{\theta\in\Theta} \sum_{x\in\{0,1\}^\lambda} \mathsf{Tr}\left[\left(|\psi_x^\theta\rangle\langle\psi_x^\theta| \otimes (B'_x)^\theta \otimes (C'_x)^\theta\right) \rho_{ABC}\right] \\
&= \frac{1}{2^{2\lambda}|\Theta|} \sum_{\theta\in\Theta} \sum_{x,r,s,u\in\{0,1\}^\lambda} \mathsf{Tr}\left[\left(|\psi_x^\theta\rangle\langle\psi_x^\theta| \otimes B_{x\oplus r}^{(\theta,r)} \otimes C_{x\oplus r}^{(\theta,r)}\right)\left(|\psi_s^\theta\rangle\langle\psi_u^\theta| \otimes \phi\left(|\psi_s^\theta\rangle\langle\psi_u^\theta|\right)\right)\right] \\
&= \frac{1}{2^{2\lambda}|\Theta|} \sum_{\theta\in\Theta} \sum_{x,r\in\{0,1\}^\lambda} \mathsf{Tr}\left[\left(B_{x\oplus r}^{(\theta,r)} \otimes C_{x\oplus r}^{(\theta,r)}\right) \phi\left(|\psi_x^\theta\rangle\langle\psi_x^\theta|\right)\right] \qquad (5) \\
&= \mathbb{E}_{m,\theta,r}\, \mathsf{Tr}\left[\left(B_m^{(\theta,r)} \otimes C_m^{(\theta,r)}\right) \phi\left(|\psi_{m\oplus r}^\theta\rangle\langle\psi_{m\oplus r}^\theta|\right)\right] \qquad (6)
\end{aligned}$$

After putting $x = m \oplus r$ in (5), we see that (6) above equals the winning probability of $(\mathcal{A}, \mathcal{B}, C)$ in the cloning experiment, and it is bounded by $p_G = 2^{-n+t} + \mathsf{negl}(\lambda)$ which suffices for the proof. □

We are not aware of a MOE game with value provably less than $(1/2 + 1/2\sqrt{2})^n$, nor are we aware of a proof that it does not exist. Nevertheless, any advancement on this front will give insight to optimal uncloneable-security by Theorem 8.

6 Additional Results on Uncloneable Encryption

6.1 A Lower Bound for Conjugate Encryption.

A natural question to explore is whether 0-uncloneable security[15] is possible, even for single-bit messages, since 0-uncloneable security means that a cloning adversary $(\mathcal{A}, \mathcal{B}, C)$ does not benefit from cloning the ciphertext at all, and hence cannot do better than the trivial strategy of giving the ciphertext to $\mathcal{B}$ and having C randomly guess the message. In this section we show that the conjugate encryption of [*BL20*] is not 0−uncloneable secure. To show this, we note that the valid ciphertexts in conjugate encryption for one-bit messages all lie on the xz-plane of the Bloch Sphere, i.e. they do not have an imaginary phase in the computational basis. Besides, encrypting multi-bit

[15] [BL20] show that 0-uncloenable security implies uncloneable-indistinguishable security, making this question more interesting.

messages is done simply by encrypting each bit separately. The following lemma, which refers to the optimal equatorial cloner studied in [BCMDM00], will take advantage of this fact:

Lemma 4. *Let $\mathcal{D} = \mathcal{D}(\mathcal{H}_2)$ denote the space of one-qubit states. Then, there exists a cloning map $\Phi : \mathcal{D} \to \mathcal{D} \otimes \mathcal{D}$ such that $F(\rho, \mathsf{Tr}_C(\Phi(\rho))) \geq 1/2 + 1/2\sqrt{2}$ and $F(\rho, \mathsf{Tr}_B(\Phi(\rho))) \geq 1/2 + 1/2\sqrt{2}$ for any ρ which is a valid ciphertext in conjugate encryption, where Tr_X is the partial trace operation of tracing out the X register.*

The following result, then is imminent:

Theorem 9. *Conjugate encryption is not (cn)-uncloneable secure for any constant $c < 1/2$.*

Proof. It suffices to construct a cloning adversary adversary $(\mathcal{A}, \mathcal{B}, C)$ which succeeds with probability $2^{-n/2}$. At a high level, we do the following: since every qubit is encrypted individually, $\mathcal{B}$ and C will independently guess each qubit of the message.

By Lemma 4, there exists a cloner $\Phi : \mathcal{D}(\mathcal{H}_2) \to \mathcal{D}(\mathcal{H}_2) \otimes \mathcal{D}(\mathcal{H}_2)$ which clones every qubit of a valid ciphertext ρ_{CT} with fidelity $f = 1/2 + 1/2\sqrt{2}$. Given a ciphertext ρ_{CT} in phase 1, $\mathcal{A}$ will use the map $\Phi^{\otimes n}$ to split it into two registers of n-qubits, so that if $\rho_B = \mathsf{Tr}_C(\Phi^{\otimes n}(\rho_{\mathsf{CT}})) = \bigotimes_{i=1}^{n} \rho_{B,i}$ is the local view of $\mathcal{B}$, then $F(\rho_{B,i}, \rho_{\mathsf{CT},i}) \geq f$ (similarly for C).

In phase 2, after the key k is revealed, $\mathcal{B}$ and C each apply $\mathsf{Dec}(k, .)$ to their register, which can be applied qubit-wise. Since fidelity cannot decrease with quantum operations Lemma 1, the local view of $\rho'_{B,i}$ of $\mathcal{B}$ after decrypting has fidelity at least f to $|m_i\rangle\langle m_i| = \mathsf{Dec}(k, \rho_{ct})$, meaning $\langle m_i|\rho'_{B,i}|m_i\rangle \geq f$ (similarly for C).

Next, $\mathcal{B}$ and C measure their register in the standard basis. By definition of fidelity, then $\Pr[m_{B,i} = m_i] \geq f$ and $\Pr[m_{C,i} = m_i] \geq f$. By union bound, this implies $\Pr[m_{B,i} = m_{C,i} = m_i] \geq 2f - 1 = 2^{-1/2}$. Since every bit of the message m is independent, it follows that $\Pr[m_B = m_C = m] \geq (2f-1)^n = 2^{-n/2}$ as desired. □

7 Construction of Copy-Protection from Uncloneable Encryption

In this section, we present an application of uncloneable encryption by showing that the existence of an uncloneable-indistinguishable secure scheme (see Definition 9) implies a copy-protection scheme over a special class of point functions. Uncloneable-indistinguishable security seems to be a stronger notion than uncloneable security, and it remains open question whether it is possible. The main drawback of our construction is that the copy-protected program ρ_f for the point function $f_{a,b}$ is reusable only if it is used to evaluate the function on the "correct" input a. When f is evaluated on inpnuts $x \neq a$, our scheme does not guarantee that ρ_f will not be destroyed.

Construction: Let $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ be a post-quantum signature scheme, and let UE be an uncloneable-indistinguishable secure uncloneable-encryption scheme encrypting n-bit messages, where n is the size of a signature created by $\mathsf{Sign}()$. We construct a copy-protection scheme $(\mathsf{CopyProtect}, \mathsf{Eval})$ for the family $\mathcal{F} = \{f_{k,(vk||\sigma)} : k \leftarrow \mathsf{UE.Setup}(1^\lambda), (vk, sk) \leftarrow \mathsf{Gen}(1^\lambda), \sigma \leftarrow \mathsf{Sign}(sk, 0)\}$ of point functions. Let X, Y denote the domain and codomain of $f \in \mathcal{F}$.

- **Copy-Protected State Generation:** On input the security parameter 1^λ and description (a, b) of a function $f_{a,b} \in \mathcal{F}$, CopyProtect does the following:
  - Parse a as k and b as $vk||\sigma$.
  - Compute $\rho \leftarrow \mathsf{UE.Enc}(k, \sigma)$.
  - Output $\tilde{\rho} = \rho \otimes |vk\rangle\langle vk|$.
- **Evaluation:** On input the security parameter 1^λ, a value $x \in X$ and a copy-protected state $\tilde{\rho}$, Eval does the following:
  1. Measure the second register of $\tilde{\rho}$ to obtain the state $\rho \otimes |vk\rangle\langle vk|$
  2. Compute $\sigma' \leftarrow \mathsf{UE.Dec}(x, \rho)$ and $\delta \leftarrow \mathsf{Ver}(vk, 0, \sigma')$.
  3. If $\delta = 0$, set $y = 0$; if $\delta = 1$, set $y = vk||\sigma'$. Output $|vk\rangle\langle vk| \otimes |y\rangle\langle y|$.

  Using Corollary 1, we can reimplement the second and third steps above so that Eval outputs a state $(\rho' \otimes |vk\rangle\langle vk|) \otimes |y\rangle\langle y|$, where ρ' is close to ρ on correct inputs. We assume that Eval does this for reusability purposes.

Computational Correctness: Our construction satisfies computational correctness (Definition 5). In order to find an input that evaluates incorrectly, an adversary must be able to forge a signature using only the verification key vk. We formalize this argument below:

Claim 4. *Assuming uncloneable-indistinguishability property of UE and the unforgeability property of the unique signature scheme,* (CopyProtect, Eval) *satisfies computational* $(\mathsf{negl}(\lambda), \mathsf{negl}(\lambda))$*-correctness.*

Proof. The first bullet point of the computational correctness property follows from the statistical correctness of UE and Corollary 1. This is because when the function is evaluated with the correct key ($a = k$), the decryption succeeds with probability $1 - \mathsf{negl}(\lambda)$, which implies that it is (almost) reversible.

We prove the second bullet via proof by contradiction. Consider the following hybrid experiments:

Hyb_1: This corresponds to the real correctness experiment, where the adversary receives as input a copy-protection of the point function $f_{a,b}$ and needs to find a value $x' \neq a$ such that the evaluation of the copy-protected state on the input x' yields a non-zero value. Let the success probability of $\mathcal{A}$ in this experiment be ε_1.

Hyb_2: This hybrid is identical to Hyb_1, except that we change the way we are computing the UE ciphertext. Instead of computing $\rho \leftarrow \mathsf{UE.Enc}(k, \sigma)$, we compute $\rho \leftarrow \mathsf{UE.Enc}(k, 0)$. Let the success probability of $\mathcal{A}$ in this experiment be ε_2.

We first argue that $|\varepsilon_1 - \varepsilon_2| \leq \mathsf{negl}(\lambda)$. To prove this, we will construct an adversary $\mathcal{A}'$ which tries to break the one-time indistinguishable security of UE:

- $\mathcal{A}'$ samples $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and computes $\sigma \leftarrow \mathsf{Sign}(sk, 0)$. She then sends two messages $m_0 = \sigma$ and $m_1 = 0$ to the challenger.

- The challenger samples $k \leftarrow \mathsf{UE.Setup}(1^\lambda)$ and a uniformly random bit b. He sends $\rho_{\mathsf{CT}} \leftarrow \mathsf{UE.Enc}(k, m_b)$ to $\mathcal{A}'$.
- $\mathcal{A}'$ sets $\rho_f^{(0)} = \rho_{\mathsf{CT}} \otimes |vk\rangle\langle vk|$ and simulates the correctness experiment corresponding to $f_{k,(vk||\sigma)}$ by running $\mathcal{A}$ and playing the role of the challenger in that experiment. She outputs 1 if $\mathcal{A}$ succeeds; otherwise, she outputs 0.

If $b = 0$, then $m_b = \sigma$ and the view of $\mathcal{A}$ is Hyb_1. Hence, $\mathcal{A}'$ outputs 1 with probability ε_1.
On the other hand, if $b = 1$, then $m_b = 0$ and the view of $\mathcal{A}$ is Hyb_2, so that $\mathcal{A}'$ outputs 1 with probability ε_2.
Therefore, by one-time indistinguishable security of UE, it follows that $|\varepsilon_1 - \varepsilon_2| \leq \mathsf{negl}(\lambda)$.

Secondly, we argue that $\varepsilon_2 \leq \mathsf{negl}(\lambda)$ by constructing an adversary Forger which tries to break the unforgeability property of the signature scheme:

- The challenger samples $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and sends vk to Forger.
- Forger samples $k \leftarrow \mathsf{UE.Setup}(1^\lambda)$ and computes $\rho_{\mathsf{CT}} \leftarrow \mathsf{UE.Enc}(k, 0)$. He sets $\rho_f^{(0)} = \rho_{\mathsf{CT}} \otimes |vk\rangle\langle vk|$ and simulates Hyb_2 by running $\mathcal{A}$ and playing the role of the challenger in that experiment. If there exists a query x_i such that the answer $y_i = vk'||\sigma'$ to that query satisfies $y_i \neq 0$, then Forger outputs σ'; otherwise, Forger outputs 0.

With probability ε_2, $\mathcal{A}$ will succeed in the experiment Hyb_2, and $\mathcal{A}'$ will output σ' such that $y_i = vk'||\sigma'$, where $\rho_f^{(1)} \otimes |y_i\rangle\langle y_i| \leftarrow \mathsf{Eval}(1^\lambda, \rho_f^{(i)}, x_i)$ for a query x_i.

Note that in our construction, even though the states $\left(\rho_f^{(j)}\right)_{j=0}^{\mathrm{poly}(\lambda)}$ could be different, they preserve the initial verification key vk. Hence, Eval always runs signature verification using vk. Therefore, Forger outputs a valid signature σ' on 0 with probability ε_2, so it is negligible by the unforgability of the signature scheme.

□

Copy-Protection Security:

Claim 5. *The construction above is a secure copy-protection scheme assuming the one-time existential unforgeability property of the signature scheme* $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ *and the uncloneable-indistinguishable security of the uncloneable encryption scheme* UE.

Proof. Suppose there exists an adversary $(\mathcal{A}, \mathcal{B}, C)$ that breaks the copy-protection security of our construction (see Definition 6). Let Hyb_1 be the corresponding pirating experiment, where the challenger always sends $x = k$ (in the original pirating experiment, he sends this input only half the time). It follows that with non-negligible probability p, both $\mathcal{B}$ and C output $(vk||\sigma)$ in Hyb_1. In other words, given a copy-protected program ρ_f for a point function $f_{k,(vk||\sigma)}$, $\mathcal{A}$ can prepare a bipartite state on registers B and C such that on input k, both $\mathcal{B}$ and C output σ with probability p. (We ignore vk in the output for simplified notation in this proof.)

We define a new experiment Hyb_2, which is identical to Hyb_1 except when the challenger is computing the copy-protected state $\rho_f = \mathsf{UE.Enc}(k, \sigma) \otimes |vk\rangle\langle vk|$, he insteads computes $\rho'_f = \mathsf{UE.Enc}(k, 0) \otimes |vk\rangle\langle vk|$ and sends it to $\mathcal{A}$.

We first argue that in Hyb_2, the probability that either $\mathcal{B}$ or C outputs σ is negligible in λ. This follows from the fact that if w.l.o.g. $\mathcal{B}$ outputs σ with non-negligible probability, then there exists an adversary Forger which breaks the unforgeability of the signature scheme:

- Forger Given the security parameter 1^λ and vk such that $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, Forger samples a key $k \leftarrow \mathsf{UE.Setup}(1^\lambda)$ and computes $\rho \leftarrow \mathsf{UE.Enc}(k, 0)$.
- Forger then runs $(\mathcal{A}, \mathcal{B}, C)$ by sending $\rho'_f = \rho \otimes |vk\rangle\langle vk|$ to $\mathcal{A}$ and simulating the experiment Hyb_2. It outputs the output of $\mathcal{B}$, which is a valid signature on 0 with non-negligible probability.

Now we construct a cloning-distinguishing adversary $(\mathcal{A}', \mathcal{B}', C')$ which breaks the uncloneable-indistinguishable security of UE:

- In phase 1, $\mathcal{A}'$ samples $(vk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and computes $\sigma \leftarrow \mathsf{Sign}(sk, 0)$. She then sends messages $m_0 = \sigma$ and $m_1 = 0$ to the challenger.
- In phase 2, the challenger computes $\rho_{\mathsf{CT}} = \mathsf{UE.Enc}(k, m_b)$ for $k \leftarrow \mathsf{UE.Setup}(1^\lambda)$ and a uniformly random bit b. He sends ρ_{CT} to $\mathcal{A}'$.
- $\mathcal{A}'$ runs $\mathcal{A}$ by sending $\rho_{\mathsf{CT}} \otimes |vk\rangle\langle vk|$ as the copy-protected program and to create a bipartite state over registers B, C. She sends the B (resp., C) register to $\mathcal{B}'$ (resp., C').
- In phase 3, the key k is revealed to $\mathcal{B}'$ and C'. $\mathcal{B}'$ then runs $\mathcal{B}$ as if $x_B = k$ in the pirating experiment, similarly for C'. Note that if $b = 0$, the view of $\mathcal{B}$ and C is exactly Hyb_1 and if $b = 1$ it is Hyb_2. Let the output of $\mathcal{B}$ and C be y_B and y_C, respectively. In the end, $\mathcal{B}'$ (resp., C') outputs the bit $b_B = 0$ if and only if $y_B = \sigma$ (resp., $y_C = \sigma$).

The probability that $\mathcal{B}'$ and C' simultaneously predict the bit b correctly is given by

$$\frac{1}{2}\left(\Pr[y_B = y_C = \sigma \mid b = 0] + \Pr[y_B \neq \sigma \wedge y_C \neq \sigma \mid b = 1]\right) \geq \frac{1}{2}(p + 1 - \mathsf{negl}(\lambda)) \geq \frac{1}{2} + \frac{p}{2} - \mathsf{negl}(\lambda),$$

thus breaking the uncloneable-indistinguishable security.

□

References

[Aar04] Scott Aaronson. Limitations of quantum advice and one-way communication. In *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, CCC '04, page 320–332, USA, 2004. IEEE Computer Society.

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

[Aar16] Scott Aaronson. The complexity of quantum states and transformations: From quantum money to black holes, 2016.

[ABSV15] Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In *Annual Cryptology Conference*, pages 657–677. Springer, 2015.

[AGKZ20] Ryan Amos, Marios Georgiou, Aggelos Kiayias, and Mark Zhandry. One-shot signatures and applications to hybrid quantum/classical authentication. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 255–268, 2020.

[ALL+20] Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. *arXiv preprint arXiv:2004.09674*, 2020.

[ALP21] Prabhanjan Ananth and Rolando L La Placa. Secure software leasing. *Eurocrypt*, 2021.

[BCG+] H. Barnum, C. Crepeau, D. Gottesman, A. Smith, and A. Tapp. Authentication of quantum messages. *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*

[BCMDM00] Dagmar Bruß, Mirko Cinchetti, G. Mauro D'Ariano, and Chiara Macchiavello. Phase-covariant quantum cloning. *Physical Review A*, 62(1), Jun 2000.

[BGS13] Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs. In *Annual Cryptology Conference*, pages 344–360. Springer, 2013.

[BI20] Anne Broadbent and Rabib Islam. Quantum encryption with certified deletion. In *Theory of Cryptography Conference*, pages 92–122. Springer, 2020.

[BJL+21] Anne Broadbent, Stacey Jeffery, Sébastien Lord, Supartha Podder, and Aarthi Sundaram. Secure software leasing without assumptions, 2021.

[BL20] Anne Broadbent and Sébastien Lord. Uncloneable quantum encryption via oracles. In *TQC*, 2020.

[BLMZ19] James Bartusek, Tancrède Lepoint, Fermi Ma, and Mark Zhandry. New techniques for obfuscating conjunctions. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 636–666. Springer, 2019.

[BSW11] Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography*, pages 253–273. Springer, 2011.

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

[GM97] N. Gisin and S. Massar. Optimal quantum cloning machines. *Phys. Rev. Lett.*, 79:2153–2156, Sep 1997.

[Gol07] Oded Goldreich. *Foundations of cryptography: volume 1, basic tools*. Cambridge university press, 2007.

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

[GVW12] Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption with bounded collusions via multi-party computation. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 162–179, 2012.

[GZ20] Marios Georgiou and Mark Zhandry. Unclonable decryption keys. *IACR Cryptol. ePrint Arch*, 877(2020):3, 2020.

[HJO+16] Brett Hemenway, Zahra Jafargholi, Rafail Ostrovsky, Alessandra Scafuro, and Daniel Wichs. Adaptively secure garbled circuits from one-way functions. In *Annual International Cryptology Conference*, pages 149–178. Springer, 2016.

[HMNY21] Taiga Hiroka, Tomoyuki Morimae, Ryo Nishimaki, and Takashi Yamakawa. Quantum encryption with certified deletion, revisited: Public key, attribute-based, and classical communication, 2021.

[KNY20] Fuyuki Kitagawa, Ryo Nishimaki, and Takashi Yamakawa. Secure software leasing from standard assumptions. *arXiv preprint arXiv:2010.11186*, 2020.

[MST21] Christian Majenz, Christian Schaffner, and Mehrdad Tahmasbi. Limitations on uncloneable encryption and simultaneous one-way-to-hiding. Cryptology ePrint Archive, Report 2021/408, 2021. https://eprint.iacr.org/2021/408.

[Nie96] M. A. Nielsen. The entanglement fidelity and quantum error correction. *arXiv e-prints*, pages quant–ph/9606012, June 1996.

[O'N10] Adam O'Neill. Definitional issues in functional encryption. *IACR Cryptology ePrint Archive*, 2010:556, 2010.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[Rom90] John Rompel. One-way functions are necessary and sufficient for secure signatures. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 387–394, 1990.

[SS10] Amit Sahai and Hakan Seyalioglu. Worry-free encryption: functional encryption with public keys. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 463–472. ACM, 2010.

[TFKW13] Marco Tomamichel, Serge Fehr, Jedrzej Kaniewski, and Stephanie Wehner. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics*, 15(10):103002, Oct 2013.

[Wie83] Stephen Wiesner. Conjugate coding. *ACM Sigact News*, 15(1):78–88, 1983.

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

[WZ82] William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[Zha19] Mark Zhandry. Quantum lightning never strikes the same state twice. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 408–438. Springer, 2019.