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. The constructions proposed by Broadbent and Lord have the disadvantage that they only guarantee one-time security; that is, the encryption key can only be used once to encrypt the message.

In this work, we study uncloneable encryption schemes, where the encryption key can be reused 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.

1 Introduction

Quantum mechanics has led to the discovery of many fascinating cryptographic primitives 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 simulatenously. If they do, we declare that the adversary wins this game. An encryption scheme satisfies uncloneability property if any adversary wins 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 both their encryption schemes is that they 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 comprimising on security. Another (related) drawback is that their scheme was inherently a private-key scheme, meaning that only the entity possessing the private encryption key could compute the ciphertext.

Our Work. 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].

In addition to the above results, we also revist the Broadbent and Lord's construction of one-time uncloneable encryption from monogamy of entanglement games [TFKW13]. Their scheme shows that the success probability of any adversary cannot be better than 0.85^n, where n is the message length. Firstly, it is natural to ask if this bound is an artifact of the proof and whether it is possible to come up with a better proof that gets the ideal bound of 0.5^n. We argue that the bound is not artifact of the proof by demonstrating the existence of an adversary that can violate their scheme with probability 0.7^n. Moreover, we also generalize 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.

1.1 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.

[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 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.

- The first scheme is a (one-time) uncloneable encryption scheme, as considered in the work of [BL20]. We denote this scheme by otUE.
  An instantiation of such a scheme is

- 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 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 could 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.

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

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.

Instantiations. 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 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.

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 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 $f(x)$. To achieve the fake-key property, the fake key generation algorithm generates a functional key with the message embedded in it; we defer the details to the technical sections. 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.

1.2 Structure of this Paper

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]. In section 4, we give a construction for the private-key setting. Due to the similarity in analysis, we give the public-key construction in appendix B.

[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.

2 Preliminaries

2.1 Notation

We denote the security parameter by λ. We denote by $\mathsf{negl}()$ a negligible function. 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.

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 $\mathcal{D}(\mathcal{H}_X)$. Valid quantum operations from register X to register Y are represented by the set of linear, completely positive trace-preserving (CPTP) maps $\phi : \mathcal{D}(\mathcal{H}_X) \to \mathcal{D}(\mathcal{H}_Y)$. 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 $\sum_x F_x = \mathbf{id}_X$. The probability of measuring outcome x on state ρ equals $\mathsf{Tr}(F_x\rho)$.

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 $\mathcal{D}_1$.

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 definition was introduced by [BL20] in the context of quantum encryption of classical messages (QECMs). A one-time uncloneable encryption scheme otUE consists of the following 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 the key k and message $m \in \{0,1\}^n$, it outputs the ciphertext CT.
- **Decryption,** $\mathsf{otUE.Dec}(k, \mathsf{CT})$: on input the key k, ciphertext CT, it outputs the message m'.

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

Definition 1 ((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 2 (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, $\mathcal{A}$ is given a ciphertext $\mathsf{Enc}(k, m)$ for $k \leftarrow \mathsf{Setup}(1^\lambda)$. Then, $\mathcal{A}$ 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), which she sends to $\mathcal{B}$ and $\mathcal{C}$, respectively.*
- *In phase 2, 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 message m_B (resp., m_C).*
- *The adversary wins iff $m_B = m_C = m$.*

Instantiations. The work of Broadbent and Lord [BL20] presented two constructions of one-time uncloneable encryption. Their first construction, "conjugate encryption", which encrypts messages of length $n = \lambda$, is information-theoretic and $\lambda \log_2(1 + 1/\sqrt{2})$-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 $\log_2(9)$-uncloneable security. This scheme does not provide a bound for the single-bit message case.

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 to possibly obtain better security.

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 C.1 and C.2.

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 3 (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* $q = poly(\lambda)$*,*

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

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

We define a private-key uncloneable encryption scheme below.

Definition 4. *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 2).*

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 5 (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 6. *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 2).

For a construction of public-key encryption using functional encryption, see appendix B.

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 7 (Fake-Key Property)**.** *We say that a classical encryption scheme* $(\mathsf{Setup}, \mathsf{Enc}, \mathsf{Dec})$ *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 3. *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: See C.3

□

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.

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 to replace the message m. We perform this for all the q messages, where $q = \mathrm{poly}(\lambda)$ 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 $\phi : \mathcal{D}(\mathcal{H}_A) \to \mathcal{D}(\mathcal{H}_B) \otimes \mathcal{D}(\mathcal{H}_C)$ 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 $\phi : \mathcal{D}(\mathcal{H}_A) \to \mathcal{D}(\mathcal{H}_B) \otimes \mathcal{D}(\mathcal{H}_C)$ 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 $p + \mathsf{negl}(\lambda)$, 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 [5], 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), 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 4. *If* UE *is t-uncloneable secure, then the proposed scheme is also t-uncloneable secure.*

Corollary 5. *The above proposed scheme is $n \log_2(1 + \frac{1}{\sqrt{2}})$-uncloneable secure, where n is the message length.*

References

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

[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.

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

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

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

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

[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.

[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.

[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.

[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.

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

A 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 8. *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.

B 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[6].

- 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 Appendix A.

B.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.

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.

B.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 6. *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.

[6]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.

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

Hybrid 2: Same as Hybrid 1, except ct in 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 $p + \mathsf{negl}(\lambda)$.*

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 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 $p + \mathsf{negl}(\lambda)$.*

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 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 FE.MSK, she has learned $\mathsf{FE}.sk$, which is the only time 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 $ct_1 \leftarrow \mathsf{FE.Enc}(\mathsf{FE.mpk}, (0, k_{SKE}, \bot))$. 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 $\tilde{\tilde{\mathcal{C}}}$ 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 $\tilde{\tilde{\mathcal{C}}}$.
- In Phase 2, the challenger reveals k_{UE} to both $\tilde{\tilde{\mathcal{B}}}$ and $\tilde{\tilde{\mathcal{C}}}$. $\tilde{\tilde{\mathcal{B}}}$ 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, $\tilde{\tilde{\mathcal{C}}}$ runs C to obtain and output m_C.

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

□

C Additional Proofs:

C.1 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 9 (Real Orthogonal Basis)**.** *Let $(|x\rangle\langle x|)_{x\in X}$ be the standard basis for $\mathcal{D}(\mathcal{H}_X)$, with $X = \{0, 1, \ldots, \dim \mathcal{H}_X - 1\}$. A basis $\beta = (|\psi_x\rangle\langle\psi_x|)_{x\in X}$ is called real-orthogonal if for every $x \in X$, there exist real coefficients $\alpha_{x'}$ such that*

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

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 1. *If β is a real orthogonal basis, then*

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

and hence

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

Definition 10 (Real-Orthogonal Monogamy Game)**.** *Let $X = \{0, 1, \ldots, 2^n - 1\}$ 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 $\mathcal{G}$ is the maximimum value the following expression can take for the optimal adversary:*

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

where

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

Intuitively, this is the probability that both $\mathcal{B}$ and $\mathcal{C}$, who only interact with registers B and C, respectively, simultaneously guess the outcome x $\mathcal{A}$ gets after she measures the A register in a uniformly random basis β^θ and announces θ.

Theorem 7. *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 message length n, which is t-uncloneable secure.*

Proof. We generalize the conjugate encryption of [BL20]:

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

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 $x \oplus r$.

Note that conjugate encryption is a special case of this construction, where the real-orthogonal bases are chosen to be Wiesner bases.

Indistinguishable Security: It suffices to show that for any message, the view of the adversary equals the completely mixed state, which can easily be calculated as

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

***t*-uncloneable security (sketch):** Follows by the proof of uncloneable security of conjugate encryption given in [BL20][7]. It turns out that the success probability of a cloning adversary equals

[7] See Lemma 14 and Corollary 2 in [BL20] with proofs.

that of a ROMG adversary, which is restricted in preparing the state ρ_{ABC} to create an EPR pair and apply the splitting map Φ to the BC register. We only need Lemma 1 for this proof to go through, as the EPR pair then does not depend on the state, before and after the splitting. □

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 7.

C.2 A Lower Bound for Conjugate Encryption.

A natural question to explore is whether 0-uncloneable security 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 messages is done simply by encrypting each bit separately. The following lemma, which corresponds to the optimal equatorial cloner studied in [BCMDM00], will take advantage of this fact:

Lemma 2. *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 $F(\rho, \mathsf{Tr}_B(\Phi(\rho))) \geq 1/2 + 1/2\sqrt{2}$ and $F(\rho, \mathsf{Tr}_C(\Phi(\rho))) \geq 1/2 + 1/2\sqrt{2}$ for any ρ which is a valid ciphertext in conjugate encryption.*

The following result, then is imminent:

Theorem 8. *Conjugate encryption is not t-uncloneable secure for any $t < \lambda/2$.*

Proof. It suffices to come up with a cloning adversary adversary $(\mathcal{A}, \mathcal{B}, C)$ which succeeds with probability $2^{-n/2}$. Since every qubit is encrypted separately, we can assume w.l.o.g. $n = 1$. By Lemma 2, there exists a cloner $\Phi : \mathcal{D}(\mathcal{H}_A) \to \mathcal{D}(\mathcal{H}_B) \otimes \mathcal{D}(\mathcal{H}_C)$ which clones every ciphertext ρ_{ct} with fidelity $f = 1/2 + 1/2\sqrt{2}$. Given a ciphertext ρ_{ct} in phase 1, $\mathcal{A}$ will use this map Φ to split into two register, resulting in local views of $\mathcal{B}$ and C each having fidelity f to ρ_{ct}.
In phase 2, after the key k is revealed, $\mathcal{B}$ and C each apply $\mathsf{Dec}(k, .)$ to their register. Since fidelity cannot decrease with quantum operations, the local views of $\mathcal{B}$ and C have fidelity at least f to $|m\rangle\langle m| = \mathsf{Dec}(k, \rho_{ct})$.
Next, $\mathcal{B}$ and C measure their register in the stnadard basis. By definition of fidelity, then $\Pr[m_B = m] \geq f$ and $\Pr[m_C = m] \geq f$. By union bound, this implies $\Pr[m_b = m_C = m] \geq 2f - 1 = 2^{-1/2}$ as desired. □

C.3 Proof of Fake-Key Property in Theorem 3

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

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

Thus, the fake-key property (Equation 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 (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[8] from the fact that otp is sampled independently from r, m, and k'.

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