# Unclonable Decryption Keys

Marios Georgiou[1] and Mark Zhandry[2,3]

[1] City University of New York, New York NY 10016, USA
mgeorgiou@gradcenter.cuny.edu
[2] Princeton University, Princeton NJ 08544, USA
mzhandry@princeton.edu
[3] NTT Research

**Abstract.** We initiate the study of encryption schemes where the decryption keys are unclonable quantum objects, which we call *single decryptor* encryption. We give a number of initial results in this area:

- We formalize the notion of single decryptor encryption.
- We show that secret-key single decryptor encryption is possible unconditionally, in the setting where a limited number of ciphertexts are given. However, given an encryption oracle, we show that unconditional security is impossible.
- We show how to use a very recent notion of *one-shot signatures*, together with sufficiently powerful witness encryption, to achieve public key single decryptor encryption.
- We demonstrate several extensions of our scheme, achieving a number of interesting properties that are not possible classically.

**Keywords:** Unclonable decryption · Quantum decryption keys · Delayed decryption.

## 1 Introduction

Throughout the vast majority of its history, cryptography has offered all-or-nothing security: either a party knows the secret key and can do everything — decrypt ciphertexts, authenticate messages, etc — or the party does not know the secret key, and can do nothing. The honest users are assumed to know the key, and the adversary does not.

More recently, nuanced and fine-grained functionalities have emerged, where users are given varying amounts of power. For example, attribute based encryption [17] allows for specifying an access policy when generating ciphertexts, and only users whose attributes satisfy the policy can decrypt. Functional encryption [7] takes this a step further, by allowing various users to only learn certain functions of the message. In these settings, the adversary may be one of the users, who has some amount of power, but desires more than their key allows.

Even if a user is permitted to decrypt a given ciphertext, or learn a certain function of the plaintext, there are still a number of security guarantees one may wish for against the secret key holder:

– Perhaps we want to enforce that a secret key holder can only decrypt a certain number of ciphertexts every day. Such a mechanism could be useful in subscription broadcasts, in order to allow different subscription tiers that can consume different amounts of content each day.
– Limiting secret key usage to a single device at a time would be very useful for digital rights management. For example, traitor tracing [11] attempts to identify users who may have broadcast their secret keys on the web. Limiting secret key usage to a single device would circumvent the need for such tracing, by preventing such behavior in the first place.
– If a key has been compromised and stolen by an intruder, it would be useful to guarantee the detection of such intrusion.

Concepts like attribute-based encryption say nothing about these scenarios. And for good reason: it is possible in principle to copy the secret key from one device to another, leaving the original copy intact. Such copying would enable breaking each of the described scenarios above.

*Quantum Information and the No Cloning Theorem.* There has been much progress recently towards building quantum computers, and it seems inevitable that full-scale quantum computing will eventually be realized.

Quantum computing suggests a solution to the scenarios above: by the *no-cloning theorem* [24] which is implied by the principle that quantum operations are linear, quantum information cannot be copied. Therefore, if we make the secret key a quantum state, in principle the key cannot be copied in order to run on more than a single device.

Actually instantiating this idea, however, is highly non-trivial. While no-cloning insists that in general a quantum state cannot be copied, in its most basic form it applies essentially to random states. As shown by Wiesner [22] and Bennett and Brassard [5], such no-cloning is useful for certain tasks like basic versions of quantum money or information-theoretic key agreement. However, these applications need no additional functionality from the quantum state, beyond the fact that quantum states cannot be copied. In our case, however, not only do we need secret keys to be unclonable, we need them to be useful secret keys. This leads to the following natural question:

*Can quantum information be used to force a secret key to only work on a single device at a time.*

In other words, we are looking for very strong variants of the no-cloning theorem, where the state is not only unclonable, but also useful. A very general notion of such no-cloning was first investigated by Aaronson [1], who describe "quantum copy protection", namely turning general programs into quantum states that cannot be copied. Such a general notion would naturally encompass unclonable secret keys. However, this notion suffers from definitional difficulties [2], and has so far not been meaningfully achievable for more than the simplest of functionalities.

Here we take a different approach, and consider unclonable secret keys for specific cryptographic tasks. This is the approach very recently taken by Amos et al. [3], who give initial results for the case of *authentication*. We will instead initiate the study of unclonable secret keys for the case of *encryption*, developing a variety of new schemes and applications.

### 1.1 This work

In this paper, we initiate the study of encryption where the encryption keys, the ciphertexts and the messages are classical whereas the decryption key is quantum and unclonable; we call such an encryption a single-decryptor encryption.

*Definitions (Section 2).* We begin by introducing essential security definitions that capture the notion of single decryptors. These definitions span in two main dimensions; secret-key versus public-key and honestly versus dishonestly generated keys. We make comparisons between them as well as between these and standard security such as indistinguishability under chosen message attacks.

*Secret-key encryption with honestly generated keys (Section 4).* We then move on to constructions. We start with the simple scenario where there is a classical encryption key $ek$ and a corresponding quantum decryption key $|dk\rangle$. We imagine the adversary is given $|dk\rangle$, and tries to use it to derive two states $|dk_0\rangle, |dk_1\rangle$ such that both states are capable of decrypting ciphertexts. In the most basic setting, the adversary must clone the key without knowledge of $ek$ and without seeing any ciphertexts. Preserving the standard definition of semantic security, we require that no adversary, given $|dk\rangle$ can output two quantum states such that both of them can be used to distinguish the encryptions of two messages. We also allow the $|dk_b\rangle$ to deviate from honest decryption keys and instead take any form.

We observe that this definition resembles the definition of unclonable encryption, defined by Broadbent and Lord [10]. There, the setting is slightly different: the secret (encryption and decryption) key is classical, whereas the ciphertext is quantum. Security requires that an adversary, given the quantum ciphertext (that encrypts $m_b$ for a random bit $b$) and no access to the secret key, cannot output two ciphertexts such that, if later two isolated adversaries are given the corresponding ciphertext and the secret key, can both predict $b$. We show that selectively secure single decryptor encryption and unclonable encryption are roughly equivalent, by demonstrating how to swap the roles of ciphertext and decryption key in the Broadbent-Lord protocol. Luckily, they prove that unclonable encryption is attainable information theoretically by cleverly manipulating BB84 states [6]. Since our black box transformation does not make any computational assumptions, the resulting scheme is also secure unconditionally.

Moreover, we show that information-theoretic security is impossible in the setting where the adversary is given arbitrarily many ciphertexts before performing the splitting attack, even if the adversary is not given the encryption key. An essential tool is the gentle measurement lemma that states that deterministic quantum computations can always be rewound.

*Public-key construction (Section 5).* We continue with a public-key construction, and also in the setting on dishonestly generated keys. This means that the adversarial decryptor generates his own (classical) encryption key $ek$, and he then tries to devise two arbitrary decryption keys $|dk_1\rangle, |dk_2\rangle$ each capable of decrypting ciphertexts to $ek$. The challenging issue here is that the adversary is free to choose $ek, |dk_1\rangle, |dk_2\rangle$ all together by whatever means he chooses.

We prove that one-shot signatures [3] and extractable witness encryption with quantum auxiliary information [13, 15] suffice to build such a primitive. A one-shot signature is a signature scheme with a quantum signing key that allows for signing any single message. However, signing two different messages with respect to the same verification key is computationally infeasible. Witness encryption allows for encrypting to NP statements, with the guarantee that (1) any witness allows for decryption, and (2) the *only* way to decrypt is, intuitively, to have a witness.

The idea behind such a scheme is to first generate a public key-secret key pair of a one-shot signature. To encrypt a message, one first picks randomness $r$ and witness encrypts $m$ with respect to the language $\{(r, \sigma)\}$ where $\sigma$ is a valid signature of $r$. Crucially, any adversary who is able to split the decryption key, should also be able to generate two different signatures with respect to the same public key, therefore violating security of one-shot signatures.

Curiously, whereas one-shot signatures are inherently a one-time object — security implies that the secret key is destroyed after signing — we demonstrate how to implement the above idea so as to preserve the decryption key for future ciphertexts. In particular, the quantum decryption key in our scheme can be re-used arbitrarily many times.

*Broadcast encryption with unclonable decryption keys (Section 6).* Motivated by the goal of traitor tracing, we consider a variant of broadcast encryption with unclonable decryption keys. Such a scheme would be useful for digital rights management, as it would prevent a user from publishing their secret key, allowing for the decryption of all broadcasts. Classically, it is impossible to prevent such behavior, and instead a tracing procedure must be performed to identify the user and take remedial action.

We develop a scheme where the encryptor can specify an arbitrary set $S$ of recipients. The attacker, who controls some subset $T$ of users, can obviously create $m := |S \cap T|$ decryption keys, one for each user in $S$ that they control. We require that the adversary cannot split his keys into $m + 1$ decryption keys that can decrypt ciphertexts to $S$. In our scheme, the sizes of public keys, secret keys, and ciphertexts are all independent of the number of users.

*Splittable attribute-based encryption (Section 7).* Here we define a version of attribute-based public-key encryption with dishonestly generated keys, where the owner of the secret key is able to split the key into two different keys, each decrypting a disjoint set of ciphertexts. In more detail, we imagine that each decryption key has an underlying predicate $p$, initially mapping all attributes to 1, and each message can be encrypted with respect to an attribute $x$. Given any

such decryption key and a predicate $q$, one can split the key into two keys: one that decrypts ciphertexts with attribute $x$ such that $p(x) \wedge q(x) = 1$ and one that decrypts ciphertexts with attribute $x$ such that $p(x) \wedge \neg q(x) = 1$. Security is defined very naturally; there is no way for two isolated adversaries to decrypt an encryption that is done with respect to the same attribute. The construction in this setting can be thought of in a modular sense. First, we can build a form of "splittable attribute-based signatures" where one can only sign messages as long as they satisfy a predicate. Using the delegation mechanism of [3] we can then split the key into two keys signing messages belonging to two disjoint sets. Then, using witness encryption we can transform signatures into encryption.

*Classically revocable time-released encryption (Section 8).* We then introduce the concept of delay decryption where the decryption key not only is unclonable but it is also "slow", in the sense that its holder is forced to wait a specific time before they can decrypt again. By being unclonable, this also implies a bound on the rate at which ciphertexts can be decrypted (as in, number of ciphertexts per period of time). We note that classically, using parallelism it is always possible to amplify the rate of decryption arbitrarily.

Moreover, we go one step further and allow a revocation mechanism: if the decryptor has not had enough time to decrypt, they can generate a classical proof that they revoke a specific ciphertext. As long as this proof is generated early enough, we are sure that the decryptor will never be able to decrypt this ciphertext. Toward this, we first make use of delay signatures as defined by Amos et al. [3] in order to create a notion of revocable delayed signatures 8.1. These are essentially proofs of sequential work [18, 12], where one can only generate proofs of work sequentially even if they have a polynomial number of parallel processors. In other words, one cannot work on two different challenges in parallel, as with regular proofs of work. Moreover, revocation guarantees that we can either provide a proof of work (which is essentially a signature) for a challenge $r$ or a proof of revocation for $r$ but not both. We then use this primitive to build delay decryption.

### 1.2 Previous Work

**Unclonable Signing Keys.** Although our work is the first studying unclonability of secret keys in the encryption setting, there has already been some work in the authentication setting.

Starting with the work of Gavinsky [14] and Pastawski et al. [19], it has been shown that it is possible to encode a secret message authentication code (MAC) key into a quantum state that can be used to sign a message, yet cannot be cloned. Crucially, the MAC key is not revealed to the adversary thus allowing for information theoretic constructions.

Later, Ben-David and Sattath [4] proved that relative to a classical oracle, but queried in superposition, there is a signature scheme where the adversary is also given access to the verification key. Zhandry [25] proved that using indistinguishability obfuscation and one-way functions it is possible to obfuscate safely

this oracle. To be precise, Zhandry proved that public-key quantum money exist under the same assumptions but his result can also be applied to the construction of Ben-David and Sattath.

Back to the private key setting, Brakersky et al. [9] showed that, under the learning-with-errors assumption, there is a way to sample a pair $(crs, td)$, in such a way that $crs$ can be used to sample a classical public key together with an unclonable quantum singing key. Using the trapdoor $td$ and the public key one can verify the validity of a signature. However, security is compromised once the trapdoor is leaked.

Last, Amos et al. [3] removed the need for a trapdoor in the verification algorithm and they designed a scheme relative to a classical oracle queried in superposition. A major open question is whether such a primitive exists under standard cryptographic assumptions.

**Unclonable Ciphertexts.** The idea of encrypting messages into quantum states that offer additional security than classical encryption was introduced by Gottesman [16], where he designed an encryption scheme such that if the eavesdropper tried to retrieve information out of it then the receiver would detect it. Recently, Broadbent and Lord [10] extended the above idea to a more natural definition of unclonable ciphertexts: an adversary who does not know the secret key cannot split the quantum ciphertext into two states such that both decrypt correctly. They even gave a stronger indistinguishability definition. Here, we prove that such a scheme is sufficient to construct selectively secure single-decryptor encryption and vice-versa. In the same context, Unruh [21] created a time-released encryption scheme with revocation. In that setting, a message is encrypted into a quantum state that needs time $t$ to be decrypted. Moreover, the sender of the ciphertext can ask from the receiver to send the ciphertext back if less than time $t$ has passed. If this is the case, the sender can verify that this ciphertext has not been decrypted. Our work can be seen as a modification of Unruh's scheme in two ways. First, in our setting the ciphertext is classical and only the decryption key is quantum. Second, the proof of revocation is also classical.

### 1.3 Deterministic Computations

In the remaining of the paper we will make implicit use of the fact that a deterministic quantum computation can always be rewound. This lemma, known in the literature as the information-disturbance trade-off, the gentle measurement lemma [23] or the almost-as-good-as-new lemma [1], will allow us to use a decryption key to decrypt a message and then rewind to retrieve a key whose distance from the original key is negligible. Since by correctness of an encryption scheme, a decryption process succeeds with overwhelming probability, we can always rewind and get the original key. We thus omit the output of a new decryption key from the interface of a decryption algorithm.

### 1.4 Notation.

A function $f$ is called negligible if $f(n) = o(n^{-c})$ for any constant $c$. We say that an event happens with overwhelming probability if it happens with probability at least $1-\varepsilon(n)$, where $\varepsilon$ is a negligible function. For a set $S$, we denote by $x \leftarrow S$, the random variable drawn uniformly at random from $S$. Similarly, for a distribution $D$, we denote by $x \leftarrow D$ the random variable sampled according to $D$. We use standard math font for classical variables and classical algorithms (e.g., $sk$ or $Alg$) and calligraphic font for quantum variables and quantum algorithms (e.g., $\mathcal{sk}$ or $\mathcal{Alg}$).

## 2 New Definitions

In this section, we provide a variety of definitions for single decryptor encryption in different settings. We start with the simplest scenario where an adversary is given a quantum decryption key and is asked to clone it. We raise this to the public key setting where now the adversary is also given access to the encryption key.

We then move on to the setting where there is a way to generate a classical common reference string $crs$ together with a classical encryption key $ek$. The adversary can use the $crs$ to generate a pair $(pk, \mathcal{sk})$. Given $(ek, pk)$ one can generate ciphertexts. We require that the adversary, is not able to clone $\mathcal{sk}$ before seeing any ciphertext. Then, as before, we consider the public-key scenario and where the adversary is also given $ek$ or, equivalently, $crs$ is sufficient to generate ciphertexts.

*Entanglement of Adversarially Chosen Keys.* Unless stated otherwise, in the following, we implicitly assume that adversarially chosen quantum states $(\mathcal{s}_0, \mathcal{s}_1) \leftarrow \mathcal{A}$ can potentially be entangled with a larger system.

### 2.1 Honest Generation of Keys

In this subsection, we are interested in definitions where the encryption and decryption keys are generated honestly and later are given to the adversary. In subsection 2.2, we study a version of them where the adversary is allowed to generate them.

**Definition 1 (Single Decryptor Encryption with Honestly Generated Keys).** *A single decryptor encryption with honestly generated keys is a triple of algorithms $(\mathcal{Gen}, Enc, \mathcal{Dec})$ with the following interface:*

- $\mathcal{Gen}(1^n) : (ek, \mathcal{dk})$ *takes a security parameter $n$ in unary and returns a secret classical encryption key $ek$ and a quantum decryption key $\mathcal{dk}$.*
- $Enc(ek, m) : c$ *takes an encryption key and a message $m$ and returns a ciphertext $c$.*
- $\mathcal{Dec}(\mathcal{dk}, c) : m$ *takes a quantum decryption key and a ciphertext and output a message $m$.*

*Correctness.* The following holds with overwhelming probability over the randomness of the algorithms. If $(ek, dk) \leftarrow \mathcal{G}en(1^n)$, then for any message $m$, $\mathcal{D}ec(dk, Enc(ek, m)) = m$.

*Secret-key Security.* For any (computationally unbounded) quantum adversaries $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$, there is a negligible function $\varepsilon$ such that

$$\Pr\left[\begin{array}{l|r} \mathcal{A}_0(dk_0, c) = b & (ek, dk) \leftarrow \mathcal{G}en(1^n) \\ \mathcal{A}_1(dk_1, c) = b & (m_0, m_1, dk_0, dk_1) \leftarrow \mathcal{A}(dk) \\ & b \leftarrow \{0,1\}, c \leftarrow Enc(ek, m_b) \end{array}\right] \leq \frac{1}{2} + \varepsilon(n).$$

Notice that the above definition potentially allows for information theoretic constructions. This is because the adversary never gets to see the encryption key and therefore cannot create valid ciphertexts.

*Selective Security.* The selective security version of the above definition states that the adversary has to decide the messages $m_0, m_1$ before receiving the decryption key. Formally, for any quantum adversaries $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$, and messages $m_0, m_1$, there is a negligible function $\varepsilon$ such that

$$\Pr\left[\begin{array}{l|r} \mathcal{A}_0(dk_0, c) = b & (ek, dk) \leftarrow \mathcal{G}en(1^n) \\ \mathcal{A}_1(dk_1, c) = b & (dk_0, dk_1) \leftarrow \mathcal{A}(dk) \\ & b \leftarrow \{0,1\}, c \leftarrow Enc(ek, m_b) \end{array}\right] \leq \frac{1}{2} + \varepsilon(n).$$

We note that in the above two definitions, the adversary $\mathcal{A}$ is asked to clone the decryption key without having access to any valid ciphertext. Indeed, it could be the case that if $\mathcal{A}$ is also provided with a ciphertext then it could clone the decryption key. We deal with such stronger scenarios below. However, even this simplest form has some advantages. As we will see below, it allows for constructions that do not require high entanglement and perhaps can be implemented even with today's technology.

*Public-key Security.* By giving to the adversary access to the encryption algorithm we move to the computational setting. We require that for any quantum polynomial time adversaries $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$, there is a negligible function $\varepsilon$ such that

$$\Pr\left[\begin{array}{l|r} \mathcal{A}_0(dk_0, c) = b & (ek, dk) \leftarrow \mathcal{G}en(1^n) \\ \mathcal{A}_1(dk_1, c) = b & (m_0, m_1, dk_0, dk_1) \leftarrow \mathcal{A}(ek, dk) \\ & b \leftarrow \{0,1\}, c \leftarrow Enc(ek, m_b) \end{array}\right] \leq \frac{1}{2} + \varepsilon(n).$$

*Remark 1.* Hybrids between the two definitions where the adversary is not given access to $ek$ but instead access to the encryption oracle $Enc(ek, \cdot)$ can also be considered. This way one can potentially achieve security against computationally unbounded adversaries, as long as they are restricted to sub-exponential number of queries to the encryption oracle.

### 2.2 Dishonest Generation of Keys

Now we introduce an additional algorithm, *ParGen* that outputs a common reference string together with a secret encryption key. Similarly to the work of Zhandry on quantum lightning [25] and Amos et al. on one-shot signatures [3], a common reference string is necessary when we deal with dishonestly generated keys. The reason is that for a fixed encryption scheme that is not parameterized by a random string, there is always an adversary that can break it.

**Definition 2 (Single Decryptor Encryption with Dishonestly Generated Keys).** *A single decryptor encryption with dishonestly generated keys is a tuple of algorithms* $(ParGen, \mathcal{Gen}, Enc, \mathcal{Dec})$ *with the following interface:*

- $ParGen(1^n) : (crs, td)$ *takes a security parameter* $n$ *in unary and returns a common reference string* $crs$ *and secret trapdoor* $td$*.*
- $\mathcal{Gen}(crs) : (ek, \mathcal{dk})$ *takes a common reference string and returns an encryption key* $ek$ *and a quantum decryption key* $\mathcal{dk}$*.*
- $Enc(td, ek, m) : c$ *takes the trapdoor* $td$*, an encryption key* $ek$ *and a message* $m$ *and returns a ciphertext* $c$*.*
- $\mathcal{Dec}(\mathcal{dk}, c) : m$ *takes a quantum decryption key and a ciphertext and output a message* $m$*.*

*Correctness.* The following holds with overwhelming probability over the randomness of the algorithms. If $(crs, td) \leftarrow ParGen(1^n)$ and $(ek, \mathcal{dk}) \leftarrow \mathcal{Gen}(crs)$, then for any message $m$, $\mathcal{Dec}(\mathcal{dk}, Enc(td, ek, m)) = m$.

*Secret-key Security.* For any quantum polynomial time adversaries $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$, there is a negligible function $\varepsilon$ such that

$$\Pr\left[\begin{array}{l|r} \mathcal{A}_0(\mathcal{dk}_0, c) = b & (crs, td) \leftarrow ParGen(1^n) \\ \mathcal{A}_1(\mathcal{dk}_1, c) = b & (m_0, m_1, \mathcal{dk}_0, \mathcal{dk}_1, ek) \leftarrow \mathcal{A}(crs) \\ & b \leftarrow \{0,1\}, c \leftarrow Enc(td, ek, m_b) \end{array}\right] \leq \frac{1}{2} + \varepsilon(n).$$

Contrary to the previous scenario, here we cannot hope for an information theoretic definition. An adversary with unlimited computational power can run the algorithm $\mathcal{Gen}$ continuously until it ends up with the same $pk$ twice. By correctness, both of the corresponding decryption keys will be valid.

Although the above definition is not as natural as the one below where there is no secret trapdoor, we include it here in an attempt to exhaustively present all different versions of unclonable decryption. Moreover, as discussed in Section 5, this version can be achieved from weaker assumptions.

*Public-key Security.* Here we completely remove the need for a secret trapdoor and we get the following definition. For any quantum polynomial time adversaries $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$, there is a negligible function $\varepsilon$ such that

$$\Pr\left[\begin{array}{l|r} \mathcal{A}_0(\mathcal{dk}_0, c) = b & crs \leftarrow ParGen(1^n) \\ \mathcal{A}_1(\mathcal{dk}_1, c) = b & (m_0, m_1, \mathcal{dk}_0, \mathcal{dk}_1, ek) \leftarrow \mathcal{A}(crs) \\ & b \leftarrow \{0,1\}, c \leftarrow Enc(crs, ek, m_b) \end{array}\right] \leq \frac{1}{2} + \varepsilon(n).$$

*Remark 2.* Similarly to the honest key generation setting, we can consider a hybrid between the above two security definitions where the adversary is given oracle access to $Enc(td, \cdot, \cdot)$. Additionally, one can consider a selective security version where the adversary is required to pick the messages $m_0, m_1$ before it is given the common reference string.

### 2.3 Comparison Between Definitions

In the computational setting the following two implications are straightforward.

- Public-key security implies secret-key security, in both honest and dishonest key generation settings.
- The existence of a secure scheme with dishonest key generation implies the existence of a secure scheme with honest key generation, in both the public-key and the secret-key settings, by incorporating the *crs* as part of the encryption key.

**IND-CCA1 Security.** A comparison between unclonable decryption with honest key generation and standard indistinguishability under chosen ciphertext attacks (IND-CCA1) is also interesting. We focus on the generalized case of IND-CCA1 where the adversary is allowed to query the decryption oracle in superposition, a scenario first studied by Boneh and Zhandry [8].

In the public key setting, it holds that security with honest generation of keys implies IND-CCA1 security. For the sake of contradiction, assume that there exist adversaries $\mathcal{A}', \mathcal{A}''$ that can break IND-CCA1 security with non-negligible probability, where $\mathcal{A}'$, given the encryption key, asks for decryption queries and then outputs two messages $m_0, m_1$ and a state $s$ that is given as input to $\mathcal{A}''$ who, given an encryption of $m_b$ is requested to find $b$ (for a random $b$). We can create adversaries $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$ that can break unclonability of the decryption key as follows. $\mathcal{A}$ receives $(ek, \mathcal{dk})$ and hands a copy of $ek$ to $\mathcal{A}'$. $\mathcal{A}$ responds to the decryption queries of $\mathcal{A}'$ using $\mathcal{dk}$ and at the end of the first phase $\mathcal{A}'$ outputs a state $s$ and two messages $m_0, m_1$, which $\mathcal{A}$ subsequently forwards as its challenge messages. Moreover, $\mathcal{A}$ outputs $\mathcal{dk}$ as an input to $\mathcal{A}_0$ and $s$ as an input to $\mathcal{A}_1$. Now $\mathcal{A}_0$ receives the original $\mathcal{dk}$ and can perform an honest decryption of the challenge ciphertext and distinguish with overwhelming probability. Moreover, $\mathcal{A}_1$ given the state $s$ and the challenge ciphertext, it forwards both to $\mathcal{A}''$ and returns what $\mathcal{A}''$ returns. Since $\mathcal{A}''$ distinguishes with non-negligible probability, $\mathcal{A}_1$ will also distinguish with non-negligible probability.

In the secret key setting, the implication is the same as long as the cloning adversary is also given access to the encryption oracle.

*Relation to Quantum Money and Quantum Lightning.* Single-decryptor encryption with honest key generation and public-key security yields immediately quantum money. Similarly, the dishonest version of it yields quantum lightning. Indeed, an unclonable decryption key can be thought of as a coin (bolt). To verify

the coin (bolt), we pick a random message $m$ and we encrypt it using the encryption key to a ciphertext $c$. We then use the decryption key to decrypt $c$ and finally we verify that we get the original message $m$. If an adversary $\mathcal{A}$ could clone the coin (bolt) in a way that it is accepted by the above verification process, then this attack would also break the single-decryption property.

Similarly, the secret-key security versions of single-decryptor encryption imply secret-key quantum money as well as semi-quantum money as defined by Radian [20]. A semi-quantum money scheme is a secret-key quantum money scheme where the generation of new quantum coins does not take place on the bank's side. Instead a user can run a classical protocol with the bank, at the end of which, it ends up with a valid coin.

## 3 Existing tools

In this section we define existing primitives that we use for our constructions. In particular, we present one-shot signatures, ordered and delayed signatures and witness encryption.

### 3.1 Ordered Signatures

In an ordered signature each message comes with a time tag. The holder of the signing key is forced to produce signatures in the order of increasing tags. Although Amos et al. [3] constructed a scheme that satisfies a simulation-based definition, for our purposes it is sufficient to consider a simpler game-based definition.

**Definition 3 (Ordered Signatures).** *An ordered signature scheme is a tuple of algorithms* $(ParGen, \mathcal{Gen}, \mathcal{Sign}, Ver)$ *with the following interface:*

- $ParGen(1^n) : crs$ *takes a security parameter* $n$ *in unary and returns a common reference string* $crs$.
  $\mathcal{Gen}(crs) : (pk, \mathcal{sk})$ *takes a common reference string* $crs$ *and outputs a classical public key* $pk$ *and a quantum secret key* $\mathcal{sk}$.
  $\mathcal{Sign}(\mathcal{sk}, m, t) : (\mathcal{sk}', \sigma)$ *takes a secret key* $\mathcal{sk}$, *a message* $m$, *and tag* $t$, *and outputs an updated secret key* $\mathcal{sk}'$ *and a signature* $\sigma$.
  $Ver(crs, pk, m, t, \sigma) : b$ *takes a common reference string* $crs$, *a public key* $pk$, *a message* $m$, *a tag* $t$, *and a signature* $\sigma$ *and outputs a bit* $b$.

*Correctness. For any sequence of message/tag pairs* $(m_1, t_1), \ldots, (m_n, t_n)$ *such that* $t_1 < \ldots < t_n$, *the following hold with overwhelming probability. Suppose* $(pk, \mathcal{sk}_0) \leftarrow \mathcal{Gen}(crs)$. *Then for* $i = 1, \ldots, n$, *let* $(\mathcal{sk}_i, \sigma_i) \leftarrow \mathcal{Sign}(\mathcal{sk}_{i-1}, m_i, t_i)$. *Then we have that* $Ver(crs, pk, m_i, t_i, \sigma) = 1$ *for all* $i$.

*Security. For any quantum polynomial time adversaries $\mathcal{A}, \mathcal{A}'$, there is a negligible function $\varepsilon$ such that*

$$\Pr\left[\begin{array}{c} \forall i \in [k+1]: \\ Ver(crs, pk, m_i, t_i, \sigma_i) = 1 \\ \forall i \in [k]: t_{i+1} > t_i \end{array} \middle| \begin{array}{r} crs \leftarrow ParGen(1^n) \\ ((m_i, \sigma_i, t_i)_{i\in[k]}, pk, s\mathcal{K}) \leftarrow \mathcal{A}(crs) \\ m_{k+1} \leftarrow \{0,1\}^n \\ \sigma_{k+1} \leftarrow \mathcal{A}'(s\mathcal{K}, m_{k+1}) \end{array}\right] \leq \varepsilon(n).$$

*Remark 3 (One-shot Signatures).* A one-shot signature is an ordered signature where the verification algorithm also checks that $t = \infty$; i.e., one cannot sign more than once. In this case, the signing algorithm does not have to return an updated key.

**Theorem 1 ([3]).** *Ordered signatures exist relative to a classical oracle.*

### 3.2 Delayed Signatures

In a delayed signature [3], each message comes with a reverse delay *rd* and a forward delay *fd*. The holder of the signing key has to wait at least time *rd* in order to sign a message. After signing the message, the holder has to wait at least time *fd* before starting to sign any other message. As is the case with ordered signatures, here we only care about security with respect to random messages.

*Remark 4 (Notion of Time).* In the descriptions following, we purposefully omit the exact definition of time, or number of steps, and we prefer to simply maintain a consistent view of it throughout the paper. In most cases, as is for example in the setting studied by Döttling, Lai and Malavolta [12], the number of steps is measured as the number of queries to a random oracle that is available to both honest and malicious parties.

**Definition 4 (Delay Signatures).** *A delay signature scheme is a tuple of algorithms $(ParGen, \mathcal{Gen}, \mathcal{Sign}, Ver)$ with the following syntax:*

> *$ParGen(1^n)$ : crs takes a security parameter $n$ in unary and returns a common reference string crs.*
> *$\mathcal{Gen}(crs)$ : $(pk, s\mathcal{K})$ takes a common reference string crs and outputs a classical public key pk and a quantum secret key $s\mathcal{K}$.*
> *$\mathcal{Sign}(s\mathcal{K}, m, rd, fd)$ : $(s\mathcal{K}', \sigma)$ takes a secret key $s\mathcal{K}$, a message $m$, a reverse delay rd, and a forward delay fd and outputs an updated secret key $s\mathcal{K}'$ and a signature $\sigma$.*
> *$Ver(crs, pk, m, rd, fd, \sigma) : b$ takes a common reference string crs, a public key pk, a message $m$, a reverse delay rd, and a forward delay fd and a signature $\sigma$ and outputs a bit $b$.*

*Correctness. For any sequence of messages $(m_1, rd_1, fd_1), \ldots, (m_n, rd_n, fd_n)$, the following holds with overwhelming probability. Let $(pk, s\mathcal{K}_0) \leftarrow \mathcal{Gen}(crs)$. Then for $i \in [n]$, let $(s\mathcal{K}_i, \sigma_i) \leftarrow \mathcal{Sign}(s\mathcal{K}_{i-1}, m_i, rd_i, fd_i)$. Then we have that $Ver(crs, pk, m_i, rd_i, fd_i, \sigma) = 1$ for all $i$.*

*Security. We want the following two properties.*

- *One cannot sign a message in time less that $rd$. Formally, for any adversary $\mathcal{A}, \mathcal{A}'$ and any constant $\alpha$ there exists a negligible function $\varepsilon$ such that*

$$\Pr\left[ Ver(crs, pk, m, rd, fd, \sigma) = 1 \;\middle|\; \begin{array}{r} crs \leftarrow ParGen(1^n) \\ (rd, fd, pk, s\mathcal{k}) \leftarrow \mathcal{A}(crs) \\ m \leftarrow \{0,1\}^n \\ \sigma \leftarrow \mathcal{A}'(s\mathcal{k}, rd, m, fd) \end{array} \right] \leq \varepsilon(n),$$

  *where $\mathcal{A}'$ runs in time $(1-\alpha)rd$.*
- *One cannot sign all messages in time less that $\sum_i rd_i + fd_i - \max_i fd_i$. Formally, for any adversary $\mathcal{A}, \mathcal{A}'$ and constant $\alpha$ there exists a negligible function $\varepsilon$ such that*

$$\Pr\left[ \begin{array}{c} \forall i \in [k]: \\ Ver(crs, pk, m_i, rd_i, fd_i, \sigma_i) = 1 \end{array} \;\middle|\; \begin{array}{r} crs \leftarrow ParGen(1^n) \\ ((rd_i, fd_i)_{i\in[k]}, pk, s\mathcal{k}) \leftarrow \mathcal{A}(crs) \\ \forall i \in [k], m_i \leftarrow \{0,1\}^n \\ (\sigma_i)_i \leftarrow \mathcal{A}'(s\mathcal{k}, (rd_i, m_i, fd_i)_i) \end{array} \right] \leq \varepsilon(n),$$

  *where $\mathcal{A}'$ runs in time $(1-\alpha)\left(\sum_i rd_i + fd_i - \max_i fd_i\right)$.*

The above properties, ensure that any algorithm has to sign sequentially. In other words, starting at time $t = 0$, one has to spend time $rd$ in order to generate a signature with backward delay $rd$. After generating a signature, one has to spend time $fd$ in order to retrieve a key that can be used to continue signing messages.

### 3.3 Quantum Tokens for Digital Signatures

The notion of quantum tokens for digital signatures was initiated by Ben-David and Sattath [4]. They also devised a construction relative to a classical oracle, but query-able in superposition.

**Definition 5 (Quantum Tokens for Digital Signatures [4]).** *A quantum token for digital signatures is a tuple of algorithms $(\mathcal{Gen}, \mathcal{Sign}, Ver)$ with the following interface:*

- *$\mathcal{Gen}(1^n) : (pk, s\mathcal{k})$ takes a security parameter $n$ in unary and returns a classical public key $pk$ and a quantum secret key $s\mathcal{k}$.*
- *$\mathcal{Sign}(s\mathcal{k}, m) : \sigma$ takes quantum secret key $s\mathcal{k}$ and a message $m$ and returns a signatures $\sigma$.*
- *$Ver(pk, m, \sigma) : b$ takes a public key, a message and a signature and returns a bit $b$.*

*Correctness. The following holds with overwhelming probability over the randomness of the algorithms. If $(pk, s\mathcal{k}) \leftarrow \mathcal{Gen}(1^n)$ then for any message $m$, $Ver(pk, m, \mathcal{Sign}(s\mathcal{k}, m)) = 1$.*

*Security. For any adversary $\mathcal{A}$ there is a negligible function $\varepsilon$ such that*

$$\Pr\left[\begin{array}{l} Ver(pk, m_0, \sigma_0) = 1 \\ Ver(pk, m_1, \sigma_1) = 1 \end{array} \middle| \begin{array}{r} (pk, sk) \leftarrow Gen(1^n) \\ (m_0 \neq m_1, \sigma_0, \sigma_1) \leftarrow \mathcal{A}(pk, sk) \end{array}\right] \leq \varepsilon(n).$$

One-shot signatures imply quantum tokens for digital signatures by incorporating the random *crs* as part of the public key.

### 3.4 Witness Encryption

Here we include the definition of witness encryption defined by Garg et al. [13]. Additionally to the standard security notion that requires indistinguishability of encryptions with respect to instances not in the language, we also present a stronger version, that of extractable security, first conceived by Goldwasser et al. [15].

**Definition 6 (Witness Encryption [13]).** *A witness encryption for an NP language $L$, is a pair of algorithms $(Enc, Dec)$ with the following interface:*

- *$Enc(1^n, x, M)$ : $c$ takes a security parameter $n$ in unary, an instance $x$ and a message $m$ and outputs a ciphertext $c$.*
- *$Dec(c, w)$ : $m$ takes a ciphertext $c$ and a witness $w$ and outputs a message $m$.*

*Correctness. The following holds with overwhelming probability over the randomness of $Enc, Dec$. For any $(x, w) \in R_L$ and any message $m$, it holds that*

$$Dec(Enc(1^n, x, m), w) = m.$$

*Security. For any instance $x \notin L$ and any two messages $m_0, m_1$, it holds that*

$$Enc(1^n, x, m_0) \sim_c Enc(1^n, x, m_1).$$

*Extractable Security [15]. For any quantum polynomial time adversary $\mathcal{A}$, polynomial $p$ and messages $m_0, m_1$, there is a quantum polynomial time extractor $\mathcal{E}$ and polynomial $q$ such that for any mixed state $aux$ potentially entangled with an external register, if*

$$|\Pr[\mathcal{A}(aux, Enc(1^n, x, m_0)) = 1] - \Pr[\mathcal{A}(aux, Enc(1^n, x, m_1)) = 1]| \geq \frac{1}{p(n)}$$

*then*

$$\Pr[(x, \mathcal{E}(1^n, x, aux)) \in R_L] \geq \frac{1}{q(n)}.$$

## 4 Single-Decryptor Secret-key Encryption with Honestly Generated Keys

We begin our constructions of single decryptor encryption with the simplest possible scenario; namely, secret-key encryption with security in the setting where the honestly generated keys are given to the adversary. The key idea is to swap the roles of secret key and ciphertext in the construction of Broadbent and Lord [10].

**Definition 7 (Unclonable Encryption [10]).** *An unclonable secret-key encryption scheme is a triple of algorithms* $(\mathit{Gen}, \mathcal{Enc}, \mathcal{Dec})$ *with the following interface:*

- $\mathit{Gen}(1^n)$ : *sk takes a security parameter* $n$ *in unary and returns a classical secret key* $sk$.
- $\mathcal{Enc}(sk, m)$ : $c$ *takes a secret key* $sk$ *and a message* $m$ *and returns a quantum ciphertext* $c$.
- $\mathcal{Dec}(sk, c)$ : $m$ *takes a secret key* $sk$ *and a quantum ciphertext* $c$ *and returns a message* $m$.

*Correctness. If* $sk \leftarrow \mathit{Gen}(1^n)$ *then for any message* $m$, $\mathcal{Dec}(sk, \mathcal{Enc}(sk, m)) = m$ *with overwhelming probability.*

*Security. For any adversary* $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$ *and messages* $m_0, m_1$ *there is a negligible function* $\varepsilon$ *such that*

$$\Pr\left[\begin{array}{l}\mathcal{A}_0(c_0, sk) = b \\ \mathcal{A}_1(c_1, sk) = b\end{array}\middle| \begin{array}{r} sk \leftarrow \mathit{Gen}(1^n) \\ b \leftarrow \{0,1\}, c \leftarrow \mathcal{Enc}(sk, m_b) \\ (c_0, c_1) \leftarrow \mathcal{A}(c)\end{array}\right] \leq \frac{1}{2} + \varepsilon(n).$$

Broadbent and Lord have proven that unclonable encryption is possible information theoretically in the random oracle model if $\mathcal{A}_0$ and $\mathcal{A}_1$ are not entangled.

**Lemma 1 (Unconditional Unclonable Encryption [10]).** *There is an unconditional unclonable encryption scheme in the random oracle model if* $\mathcal{A}_0$ *and* $\mathcal{A}_1$ *are not entangled.*

Below we prove the equivalence between existence of unclonable encryption and existence of single-decryptor encryption. We note that we focus on selective security of single-decryptor encryption; namely, the adversary picks the messages $m_0, m_1$ before seeing the decryption key.

**Theorem 2.** *Single-decryptor selectively secure secret-key encryption in the setting of honestly generated keys exists if and only if unclonable encryption exists.*

*Proof.* We start by constructing a single-decryptor encryption from unclonable encryption. Let $(\mathit{Gen}', \mathcal{Enc}', \mathcal{Dec}')$ be an unclonable encryption scheme. We define the single-decryptor scheme $(\mathcal{Gen}, \mathit{Enc}, \mathcal{Dec})$ as follows:

- $\mathcal{Gen}(1^n)$ : Sample $sk \leftarrow Gen'(1^n)$ and a randomness $r$. Let $ek = (sk, r)$ and $d\mathcal{k} \leftarrow \mathcal{Enc}'(sk, r)$ and return $(ek, d\mathcal{k})$.
- $Enc(ek, m)$ : Parse $ek = (sk, r)$ and return $c = (sk, m \oplus r)$.
- $\mathcal{Dec}(d\mathcal{k}, c = (sk, d))$ : Run $r \leftarrow \mathcal{Dec}'(sk, d\mathcal{k})$ and return $d \oplus r$.

Correctness is implied by the correctness of the underlying unclonable encryption scheme. We go on to prove selective security. Assume that there exist adversaries $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$, messages $m_0, m_1$ and non-negligible function $\mu$ such that

$$\Pr\left[\begin{array}{l|r} \mathcal{A}_0(d\mathcal{k}_0, c) = b & (ek, d\mathcal{k}) \leftarrow \mathcal{Gen}(1^n) \\ \mathcal{A}_1(d\mathcal{k}_1, c) = b & (d\mathcal{k}_0, d\mathcal{k}_1) \leftarrow \mathcal{A}(d\mathcal{k}) \\ & b \leftarrow \{0,1\}, c \leftarrow Enc(ek, m_b) \end{array}\right] \geq \frac{1}{2} + \mu(n).$$

For random $r$, let $m'_b = m_b \oplus r$ and $\mathcal{A}', \mathcal{A}'_0, \mathcal{A}'_1$ be the following:

- $\mathcal{A}'(c)$ : Run $(c_0, c_1) \leftarrow \mathcal{A}(c)$.
- $\mathcal{A}'_0(c_0, sk)$ : Return $b \leftarrow \mathcal{A}_0(c_0, (sk, r))$.
- $\mathcal{A}'_1(c_1, sk)$ : Return $b \leftarrow \mathcal{A}_1(c_1, (sk, r))$.

In the above adversaries, if $c = \mathcal{Enc}'(sk, m'_b)$, then $(sk, r) = Enc((sk, m'_b), m_b)$ and follows the correct distribution since $m'_b$ is random and independent of $m_b$. Therefore, we get that there are messages $m'_0, m'_1$ and adversaries $\mathcal{A}', \mathcal{A}'_0, \mathcal{A}'_1$ that break unclonable encryption with probability $\frac{1}{2} + \mu(n)$.

For the opposite direction, assume that there exists a secure single-decryptor encryption scheme $(\mathcal{Gen}, Enc, \mathcal{Dec})$. We define the following unclonable encryption scheme $(Gen', \mathcal{Enc}', \mathcal{Dec}')$:

- $Gen'(1^n)$ : Sample $sk \leftarrow \{0,1\}^n$ and return $sk$.
- $\mathcal{Enc}'(sk, m)$ : Sample $(ek, d\mathcal{k}) \leftarrow \mathcal{Gen}(1^n)$ and compute $c \leftarrow Enc(ek, m)$. Return $c\mathcal{t} = (c \oplus sk, d\mathcal{k})$.
- $\mathcal{Dec}'(sk, c\mathcal{t} = (d, d\mathcal{k}))$ : Return $\mathcal{Dec}(d\mathcal{k}, d \oplus sk)$.

Correctness is implied by the correctness of the underlying single-decryptor encryption. For security, assume that there are adversaries $\mathcal{A}', \mathcal{A}'_0, \mathcal{A}'_1$ and messages $m'_0, m'_1$ that can break the unclonability of ciphertexts with non-negligible probability. Let $m_b = m'_b$ and define adversaries $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$ against single-decryptor security as follows:

- $\mathcal{A}(d\mathcal{k})$ : Sample random $sk \leftarrow \{0,1\}^n$, set $c\mathcal{t} = (sk, d\mathcal{k})$, run $(c\mathcal{t}_0, c\mathcal{t}_1) \leftarrow \mathcal{A}'(c\mathcal{t})$ and return $((sk, c\mathcal{t}_0), (sk, c\mathcal{t}_1))$.
- $\mathcal{A}_0((sk, c\mathcal{t}_0), c)$: Return $b = \mathcal{A}'_0(c\mathcal{t}_0, c \oplus sk)$.
- $\mathcal{A}_1((sk, c\mathcal{t}_1), c)$: Return $b = \mathcal{A}'_1(c\mathcal{t}_1, c \oplus sk)$.

In the above, we note that the states $c\mathcal{t}_0, c\mathcal{t}_1$ are the quantum ciphertexts output by $\mathcal{A}'$ as attempted copies of the ciphertext $c\mathcal{t} = (sk, d\mathcal{k})$. It holds that if $c = Enc(ek, m_b)$ then $c\mathcal{t} = (sk, d\mathcal{k}) = \mathcal{Enc}'(c \oplus sk, m_b)$ and follows the correct distribution since $sk$ is uniformly random. Therefore, there are messages $m_0, m_1$ and adversaries $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$ that break single-decryptor security with the same non-negligible probability. □

From the above reductions, if $\mathcal{A}_0, \mathcal{A}_1$ do not share any entanglement then $\mathcal{A}'_0, \mathcal{A}'_1$ do not share any entanglement either.

**Corollary 1.** *There exists an unconditional selectively secure single-decryptor secret-key encryption in the random oracle model as long as $\mathcal{A}_0$ and $\mathcal{A}_1$ do not share any entanglement.*

We note that in the above equivalence we only deal with adversaries who do not have access to a ciphertext when they attempt to clone the decryption key. In fact, in our construction of single-decryptor encryption from unclonable encryption, a single ciphertext can be combined with the decryption key to retrieve both $sk$ and $r$ and hence create more copies of the decryption key.

**Impossibility of Unconditional Security.** A natural question is whether one can achieve unconditional security given access to the encryption oracle (but queried polynomially many times). Unfortunately, unconditional decryption unclonability is impossible even in a weaker form where the adversary is only given access to arbitrarily many valid ciphertexts of random messages and no access to the encryption oracle. Notice that by correctness of the encryption scheme and by the information-disturbance trade-off, an adversary, given several ciphertexts $c_1, \ldots, c_k$ can efficiently find the corresponding plaintexts $m_1, \ldots, m_k$, such that $c_i = \mathit{Enc}(ek, m_i)$. This is done by decrypting a ciphertext to get the corresponding plaintext and subsequently rewinding to retrieve the original ciphertext and decryption key. Continuing this way, one can decrypt all given ciphertexts. However, the computational security of *Enc* implies that it should behave like a pseudo-random function which does not exist information theoretically. An interesting question is whether we can achieve single-decryption encryption given many ciphertexts from standard cryptographic assumptions.

## 5 Single-Decryptor Public-key Encryption with Dishonestly Generated Keys

In this section we aim for a construction that satisfies security against dishonestly generated keys in the public-key setting. Toward this, we assume the existence of one-shot signatures [3]; i.e., signatures where no adversary, given a common reference string, can output a public key, two different messages and a valid signature for each of the messages. We also assume extractable witness encryption [13, 15].

**Theorem 3.** *If one-shot signatures and extractable witness encryption exist, then unclonable decryption with dishonest generation of keys exists.*

*Proof.* Let $(\mathit{ParGen}', \mathcal{Gen}', \mathcal{Sign}', \mathit{Ver}')$ be a one-shot signature. We define our encryption scheme $(\mathit{ParGen}, \mathcal{Gen}, \mathit{Enc}, \mathcal{Dec})$ as follows:

- $\mathit{ParGen}(1^n)$ : Return $\mathit{crs} \leftarrow \mathit{ParGen}'(1^n)$.

- $\mathcal{Gen}(crs)$ : Return $(pk, s\mathcal{k}) \leftarrow \mathcal{Gen}'(crs)$.
- $Enc(crs, pk, m)$ : Pick a random $x \leftarrow \{0,1\}^n$, run $c \leftarrow Enc'(x, m)$ and return $(x, c)$, where $(Enc', Dec')$ is a witness encryption scheme with respect to the language $R_L = \{(x, \sigma) : Ver(crs, pk, x, \sigma) = 1\}$.
- $Dec(s\mathcal{k}, (x, c))$ : Run on superposition $s \leftarrow \mathcal{Sign}(s\mathcal{k}, x)$ and do not measure $s$. Then run on superposition $m \leftarrow Dec'(c, s)$ and measure $m$ to retrieve a classical value $m$. Finally, rewind the computation and retrieve $s\mathcal{k}$.

The correctness of the construction is implied by the underlying correctness of one-shot signatures and witness encryption.

To argue security, assume that there exist algorithms $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$ that break unclonability with dishonestly generated keys with non-negligible probability. In the original game, call it $H_0$, the challenge ciphertext $c$ is chosen such that $c \leftarrow Enc'(x, m_b)$ for randomly chosen $x$.

Let $H_1$ be a hybrid where we generate two ciphertexts $c_0, c_1$ with corresponding language instances $x_0, x_1$ chosen uniformly at random such that $x_0 \neq x_1$. The ciphertext $c_b$ is given to the adversary $\mathcal{A}_b$. Since the two adversaries are isolated, the winning probabilities in $H_0$ and $H_1$ differ by a negligible amount.

By assumption, we know that $\Pr[\mathcal{A}_0(s\mathcal{k}_0, Enc(crs, pk, m_b)) = b_0] \geq 1/2 + \varepsilon(n)$ for some non-negligible function $\varepsilon$. Thus, by invoking the witness extractor $\mathcal{E}(1^n, s\mathcal{k}_0, x_0)$ we can get a witness $\sigma_0$ such that $Ver(crs, pk, x_0, \sigma_0) = 1$ with non-negligible probability. Similarly, we can get a witness $\sigma_1$ for $x_1$. Hence, we can break security of one-shot signatures with non-negligible probability.

*Remark 5 (Running $Dec'$ on Superposition).* The above construction has a desirable property that the secret key can remain unchanged after arbitrarily many decryptions. As we described above, this is a result of the gentle measurement lemma that states that a deterministic computation does not disturb the system and hence we can rewind to our original key. On the negative side, this construction has the disadvantage that the honest decryptor is required to run the decryption algorithm of the witness encryption scheme $Dec'$ on superposition. A way to avoid this, is to start with a single-signer signature as defined by Amos et al. [3] instead of a one-shot signature. Such a signature guarantees that two isolated adversaries cannot sign messages with respect to the same verification key. In this case, the secret key has to evolve after every signature and the signature and secret key sizes increase with the number of applications of the signing algorithm. On the positive side, by using such a signature scheme we can run $Dec'$ classically and without the need to rewind to retrieve the original key.

*Secret key Security.* In the case of secret key security, while still in the scenario of dishonestly generated keys (Definition 2), one can use a weaker cryptographic primitive than one-shot signatures, namely privately verifiable one-shot signatures. Here, together with the *crs* the verifier outputs also a secret trapdoor *td* that can be used to verify a signature. Since Brakersky et al. [9] have shown that privately-verifiable one-shot signatures exist under the learning-with-errors (LWE) assumption, we can conclude that:

**Corollary 2.** *If LWE holds and extractable witness encryption exists, then unclonable secret-key decryption with dishonestly generated keys exists.*

*Honestly Generated Keys.* In the case of honestly generated keys, it is enough to use tokens for digital signatures as defined by Ben-David and Sattath [4], instead of one-shot signatures, which can be thought of as the honest key generation variant of one-shot signatures. Ben-David and Sattath have shown that such signatures are possible from a classical oracle. Later Zhandry [25] showed how to securely obfuscate the classical oracle using indistinguishability obfuscation and one-way functions.

**Corollary 3.** *If indistinguishability obfuscation, one-way functions and witness-extractable witness encryption exist, then unclonable decryption with honestly generated keys exists.*

## 6 Broadcast Encryption with Unclonable Decryption

In a broadcast encryption scheme, there is a way to generate a public key together with a set of $N$ decryption keys in an way that allows us to encrypt a message with respect to any subset $S \subseteq [N]$ of the holders of these keys. Security guarantees that only the authorized holders can decrypt. Here we impose the requirement that the decryption keys are unclonable.

**Definition 8 (Broadcast Encryption with Unclonable Decryption).** *A broadcast encryption scheme with unclonable decryption is a tuple of algorithms* $(\mathcal{Gen}, Enc, \mathcal{Dec})$ *with the following interface:*

- $\mathcal{Gen}(1^n, N) : (pk, \mathsf{sk}_1, \ldots, \mathsf{sk}_N)$ *takes a security parameter* $n$ *in unary and an integer* $N$ *and returns a master public key* $mpk$ *and* $N$ *quantum secret keys.*
- $Enc(mpk, S, m) : c$ *takes a master public key* $mpk$*, a set* $S \subseteq [N]$ *and a message* $m$ *and returns a ciphertext* $c$*.*
- $\mathcal{Dec}(S, i, \mathsf{sk}, c) : m$ *takes a set* $S$*, an index* $i$*, a quantum secret key* $\mathsf{sk}$ *and a ciphertext* $c$ *and returns a message* $m$*.*

*Correctness. The following holds with overwhelming probability. For all messages* $m$*, sets* $S \subseteq [N]$ *and* $i \in S$*, if* $c \leftarrow Enc(mpk, S, m)$ *then* $\mathcal{Dec}(S, i, \mathsf{sk}_i, c) = m$*, where* $(mpk, \mathsf{sk}_1, \ldots, \mathsf{sk}_N) \leftarrow \mathcal{Gen}(1^n, N)$*.*

*Security. For any integer* $N$ *and any quantum polynomial time adversaries* $\mathcal{A}, \mathcal{A}_1, \ldots, \mathcal{A}_N$*, there is a negligible function* $\varepsilon$ *such that*

$$\Pr\left[\begin{array}{c|c} k > |S \cap T| & (mpk, \mathsf{sk}_1, \ldots, \mathsf{sk}_N) \leftarrow \mathcal{Gen}(1^n, N) \\ \forall i \in [k], & (S, m_0, m_1, s_1, \ldots, s_k) \leftarrow \mathcal{A}^{\mathcal{KDer}}(mpk) \\ \mathcal{A}_i(s_i, c) = b & b \leftarrow \{0,1\}, c \leftarrow Enc(mpk, S, m_b) \end{array}\right] \leq \frac{1}{2} + \varepsilon(n),$$

*where* $\mathcal{KDer}$ *queried classically on an index* $i$*, returns* $\mathsf{sk}_i$ *and adds* $i$ *to the initially empty set* $T$*.*

*Succinctness. The size of the master public key mpk and the size of the ciphertexts c is independent of $N$; the size of the decryption keys $\mathsf{sk}_i$ is logarithmic in $N$.*

Security can be interpreted as follows. The adversary picks a set $S$ that he will be challenged on. He also gets to query secret keys for a number of users through the $\mathcal{KDer}$ algorithm; call this set $T$. He may be even be able to query keys from $S$, in which case $S \cap T \neq \emptyset$. He then creates $k$ quantum states that he distributes to $k$ non-communicating adversaries. Finally, the challenger encrypts either $m_0$ or $m_1$ at random, with respect to the set of users $S$ and sends the ciphertext $c$ to all $k$ adversaries. We require that at most $|S \cap T|$ can distinguish the encryptions. The intuition is that the adversary could have distributed the $k$ secret keys he possesses in $S \cap T$ to different adversaries, who would all then be able to predict $b$ with certainty. But any more, and at least one of the adversaries will fail.

**Theorem 4.** *Broadcast encryption with unclonable decryption exists if tokens for digital signatures, extractable witness encryption and collision resistant hash functions exist.*

*Proof.* First, if we do not require succinctness, then we can trivially create such a scheme by setting $mpk = (pk_1, \dots, pk_N)$ and using a standard single-decryptor public-key encryption scheme with honestly generated keys.

To achieve succinctness, we make use of Merkle trees. Let $(\mathcal{Gen}', \mathcal{Sign}, \mathcal{Ver})$ be a token for signature scheme. For a list $l$ of $N$ elements, let $H(l)$ be the Merkle hash of $l$ with respect to a collision resistant hash function and let $\mathcal{Open}(H(l), i, \pi) = l_i$ be the corresponding opening function; i.e. $\pi$ contains the nodes in the path to the root together with their siblings in the Merkle tree. We create a broadcast encryption scheme with unclonable decryption $(\mathcal{Gen}, \mathcal{Enc}, \mathcal{Dec})$ as follows:

- $\mathcal{Gen}(1^n, N)$ : For $i \in [N]$, generate $(pk_i, \mathsf{sk}_i) \leftarrow \mathcal{Gen}'(1^n)$ and set $mpk = H(pk_1, \dots, pk_N)$ and $\mathsf{sk}_i = (pk_i, \tau_i, \mathsf{sk}_i)$, where $\tau_i$ is a proof that $pk_i$ is in the $i$'th position in the list of public keys that map to $mpk$.
- $\mathcal{Enc}(mpk, S, m)$ : Let $v \in \{0,1\}^N$ such that $v_i = 1$ if and only if $i \in S$ and let $h = H(v_1, \dots, v_N)$. Moreover, sample $x \leftarrow \{0,1\}^n$ and return $(h, x, \mathcal{Enc}'((mpk, h, x), m))$, where $\mathcal{Enc}'$ is a witness encryption for the language

$$L = \{((mpk, h, x), (i, pk, \tau, \pi, \sigma)) : \begin{array}{l} \mathcal{Open}(mpk, \tau, i) = pk, \\ \mathcal{Open}(h, \pi, i) = 1, \\ \mathcal{Ver}(pk, x, \sigma) = 1\}. \end{array}$$

- $\mathcal{Dec}(S, i, (pk, \tau, \mathsf{sk}), (h, x, c))$ : Use $S, i$ to find $\pi$ such that $\mathcal{Open}(h, \pi, i) = 1$. Moreover, in superposition, generate $s \leftarrow \mathcal{Sign}(\mathsf{sk}, x)$ and again in superposition run $m \leftarrow \mathcal{Dec}'(c, (i, pk, \tau, \pi, \sigma))$. Measure $m$ to retrieve a classical message $m$, return $m$ and rewind to the original decryption key.

*Correctness.* Given $S, i$ such that $i \in S$ one can easily generate a proof $\pi$ such that $\mathit{Open}(h, \pi, i) = 1$. Moreover, by construction, $\mathit{Open}(mpk, \tau, i) = pk$ and by correctness of tokens for signatures, $\mathit{Ver}(pk, x, \mathit{Sign}(sk, x)) = 1$. Hence, decryption succeeds.

*Security.* Assume that there exists an adversary $\mathcal{A}, \mathcal{A}_1, \ldots, \mathcal{A}_N$ that can break security with non-negligible probability. Let $S, T$ be the corresponding sets and $k > |S \cap T|$. By extractable security, there exist extractors $\mathcal{E}_1, \ldots, \mathcal{E}_k$, such that $(i, pk_i, \tau_i, \pi_i, \sigma_i) \leftarrow \mathcal{E}_i(s_i, (h, x, c))$ and $\mathit{Open}(mpk, \tau_i, i) = pk_i$, $\mathit{Open}(h, \pi_i, i) = 1$ and $\mathit{Ver}(pk_i, x, \sigma_i) = 1$ with non-negligible probability. Let $R$ be the set of indices returned by the extractors and $\{pk_i^*\}_{i \in R}$ the corresponding public keys returned by the extractors.

Define the following events:

$$E_1 = \{R \subseteq S \cap T \text{ and } \{pk_i^*\}_{i \in R} \subseteq \{pk_i\}_{i \in S \cap T}$$

$$E_2 = \{R \subseteq S \cap T \text{ and } \{pk_i^*\}_{i \in R} \not\subseteq \{pk_i\}_{i \in S \cap T}\}$$

$$E_3 = \{R \not\subseteq S\}$$

$$E_4 = \{R \not\subseteq T\}$$

and by assumption, it holds that the sum $\Pr[E_1] + \Pr[E_2] + \Pr[E_3] + \Pr[E_4]$ is non-negligible, which implies that at least one of these events happens with non-negligible probability.

- Suppose $\Pr[E_1]$ is non-negligible and let $pk = pk_i^* = pk_j^*$, for some $i \neq j \in S \cap T$. Sampling random $x, x'$, and running $\mathcal{E}_i(s_i, (h, x, \mathit{Enc}'((mpk, h, x), m_b))$ and $\mathcal{E}_j(s_j, (h, x', \mathit{Enc}'((mpk, h, x'), m_b))$, we retrieve two signatures for two different $x$ under the same public key $pk$ which constitutes an attack against the tokens for signatures.
- Suppose $\Pr[E_2]$ is non-negligible and let $pk^* \neq pk_i$ for all $i \in S \cap T$. It follows that the adversary was able to open $mpk$ in some position $i$ with two different public keys, which yields a collision within the Merkle tree of $mpk$ with non-negligible probability.
- Suppose $\Pr[E_3]$ is non-negligible and let $i \in R \backslash S$ which implies that $v_i = 0$. Let $\pi_i^*$ be the corresponding opening such that $\mathit{Open}(h, \pi_i^*, i) = 1$. It follows that $\pi_i^*$ yields a collision within the Merkle tree of $h$ with non-negligible probability.
- Suppose $\Pr[E_4]$ is non-negligible and let $i \in R \backslash T$; i.e. the adversary never queried the secret key for $i$, yet it managed to sign with respect to $pk_i$. This implies a break in the security of quantum signing tokens. To see this, notice that an adversary could sign two different messages by first running the extractor defined above and subsequently use the original quantum signing message to generate a signature for a second message. Formally, our adversary $\mathcal{B}$ on input $(pk, sk)$ first generates $N - 1$ more pairs and sets up the master public key. Subsequently he runs the adversary $\mathcal{A}$ and the extractors out of which, one will return a signature under $pk$ with non-negligible probability. Since $\Pr[E_4]$ is non-negligible, it holds that $\mathcal{B}$ never sent $sk$ to $\mathcal{A}$. Hence, $\mathcal{A}$ can use $sk$ to generate a second signature.

*Succinctness.* To argue succinctness notice that $|mpk| = n$ the size of the security parameter, assuming that our hash function's co-domain is $\{0,1\}^n$. Moreover, the size of the ciphertext $|c| = O(n)$ assuming the witness encryption ciphertext grows with the size of the instance. Moreover, the size of the decryption key is $O(n \log N)$, since it includes two hash values for each level in the Merkle trees and $O(n)$ qubits for the quantum token.

*Remark 6.* As is the case with all primitives in this work, we can define two different versions of a primitive: one with honestly generated keys and one with dishonestly generated keys. For the purpose of defining a generic definition of broadcast encryption with unclonable decryption and for simplicity of the definitions, in the above we considered only the version where the decryption keys are generated honestly by the key generation algorithm $\mathcal{Gen}$. This is exactly the reason why we can get away with just signing tokens. By replacing the tokens with one-shot signatures, we can give a more powerful construction where the adversary gets to pick *all* decryption keys, yet $|S|+1$ isolated adversaries cannot decrypt an encryption with respect to $S$.

## 7 Splittable Attribute-based Encryption

**Definition 9 (Splittable Attribute-based Encryption).** *A splittable attribute based encryption is a tuple of algorithms* $(ParGen, \mathcal{Gen}, \mathcal{Split}, Enc, \mathcal{Dec})$ *with the following interface:*

- $ParGen(1^n) : crs$ *takes a security parameter* $n$ *in unary and returns a common reference string* $crs$*.*
- $\mathcal{Gen}(crs) : (pk, \mathsf{sk})$ *takes a common reference string* $crs$ *and outputs a classical encryption key* $pk$ *and a quantum decryption key* $\mathsf{sk}$*.*
- $\mathcal{Split}(\mathsf{sk}, q) : (\mathsf{sk}_0, \mathsf{sk}_1)$ *takes a quantum decryption key and a predicate* $q$ *and outputs two keys* $\mathsf{sk}_0, \mathsf{sk}_1$*.*
- $Enc(crs, pk, m, x) : c$ *takes a public key* $pk$*, a message* $m$ *and an attribute* $x$ *and outputs a ciphertext* $c$*.*
- $\mathcal{Dec}(\mathsf{sk}, c) : m$ *takes a quantum secret key* $\mathsf{sk}$ *and a ciphertext* $c$ *and outputs a message* $m$*.*

*Correctness. The following hold with overwhelming probability over* $crs$ *and the randomness of the algorithms. For a secret key* $\mathsf{sk}$*, let* $p_{\mathsf{sk}}$ *be its predicate such that if* $(pk, \mathsf{sk}) \leftarrow \mathcal{Gen}(crs)$ *then* $p_{\mathsf{sk}}(x) = 1$ *for all attributes* $x$*. Moreover, if* $(\mathsf{sk}_0, \mathsf{sk}_1) \leftarrow \mathcal{Split}(\mathsf{sk}, q)$*, then* $p_{\mathsf{sk}_0}(x) = p_{\mathsf{sk}}(x) \wedge q(x)$ *and* $p_{\mathsf{sk}_1}(x) = p_{\mathsf{sk}}(x) \wedge \neg q(x)$*. Last,* $\mathcal{Dec}(\mathsf{sk}, Enc(crs, pk, m, x)) = m$ *if* $p_{\mathsf{sk}}(x) = 1$*.*

*Security. For any quantum polynomial time algorithms* $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$ *there exists a negligible function* $\varepsilon$ *such that*

$$\Pr\left[\begin{array}{l}\mathcal{A}_0(\mathsf{sk}_0, c) = b \\ \mathcal{A}_1(\mathsf{sk}_1, c) = b\end{array}\middle|\begin{array}{r} crs \leftarrow ParGen(1^n) \\ (m_0, m_1, \mathsf{sk}_0, \mathsf{sk}_1, pk, x) \leftarrow \mathcal{A}(crs) \\ b \leftarrow \{0,1\}, c \leftarrow Enc\,(crs, pk, m_b, x)\end{array}\right] \leq \frac{1}{2} + \varepsilon(n),$$

*where $\mathit{sk}_0, \mathit{sk}_1$ can potentially be entangled.*

**Theorem 5.** *If one-shot signatures and extractable witness encryption exist, then splittable attribute-based encryption exists.*

*Proof.* Let $(\mathit{ParGen}', \mathit{Gen}', \mathit{Sign}, \mathit{Ver})$ be a one-shot signature. Define splittable attribute-based encryption as follows:

- $\mathit{ParGen}(1^n)$ : Return $\mathit{crs} \leftarrow \mathit{ParGen}(1^n)$.
- $\mathit{Gen}(\mathit{crs})$ : Return $\mathit{Gen}'(\mathit{crs})$.
- $\mathit{Split}(\mathit{sk}, q)$ :
  1. Parse $\mathit{sk} = (l, \mathit{sk}')$ where $l$ is a list $l = [(pk_i, pk'_i, q_i, \sigma_i)]_{i\in[k]}$.
  2. Sample $(pk_{k+1}, \mathit{sk}_{k+1}) \leftarrow \mathit{Gen}(\mathit{crs})$ and $(pk'_{k+1}, \mathit{sk}'_{k+1}) \leftarrow \mathit{Gen}(\mathit{crs})$.
  3. Generate signature $\sigma_{k+1} \leftarrow \mathit{Sign}(\mathit{sk}', (pk_{k+1}, pk'_{k+1}, q))$.
  4. Append $(pk_{k+1}, pk'_{k+1}, q_{k+1}, \sigma_{k+1})$ to $l$.
  5. $\mathit{sk}_0 = (l, \mathit{sk}_{k+1})$ and $\mathit{sk}_1 = (l, \mathit{sk}'_{k+1})$.
  6. Return $(\mathit{sk}_0, \mathit{sk}_1)$.
- $\mathit{Enc}(\mathit{crs}, pk, m, x)$ : Let $(\mathit{Enc}', \mathit{Dec}')$ be a witness encryption for the language

$$\begin{aligned} R_L = \{&((x, r), (l, \sigma)) : l = [(pk_i, pk'_i, q_i, \sigma_i)]_{i\in[k]}, \\ &\big(\mathit{Ver}(\mathit{crs}, pk_k, r, \sigma) \vee \mathit{Ver}(\mathit{crs}, pk'_k, r, \sigma)\big) \\ &\forall_{i\in[k]}\big((\mathit{Ver}(\mathit{crs}, pk_{i-1}, (pk_i, pk'_i, q_i), \sigma_i) \wedge q_i(x) = 1) \\ &\vee(\mathit{Ver}(\mathit{crs}, pk'_{i-1}, (pk_i, pk'_i, q_i), \sigma_i) \wedge \neg q_i(x) = 1)\big)\}, \end{aligned}$$

  where $pk_0 = pk'_0 = pk$. Return $(r, c = \mathit{Enc}'(x, m))$, for random $r$.
- $\mathit{Dec}(\mathit{sk}, (r, c))$ :
  1. Parse $\mathit{sk} = (l, \mathit{sk}')$ where $l$ is a list $l = [(pk_i, pk'_i, q_i, \sigma_i)]_{i\in[k]}$
  2. On superposition, run $s \leftarrow \mathit{Sign}(\mathit{sk}', r)$, without measuring $s$.
  3. On superposition, run $m \leftarrow \mathit{Dec}'(c, (l, s))$, measure $m$ to retrieve a classical message $m$. Rewind to retrieve $\mathit{sk}'$ and return $m$.

Intuitively, our construction can be split into two parts: in the first part, we create a type of splittable attribute-based signatures where the key, initially being able to sign a message with any attribute, is split into two keys each being able to sign messages with attributes $x$ satisfying $q(x)$ and $\neg q(x)$ respectively. Subsequently, we use witness encryption to turn our scheme into an encryption. We now go on to prove correctness and security.

*Correctness.* We will prove correctness by induction. In the base case, for any $x$, it holds that $(x, ([], \mathit{Sign}(\mathit{sk}, x))) \in R_L$, where $(pk, \mathit{sk}) \leftarrow \mathit{Gen}(\mathit{crs})$ and therefore, $p_{\mathit{sk}}(x) = 1$.

For the induction step, suppose that for a secret key $\mathit{sk} = (l, \mathit{sk}')$, with $|l| = k$, it holds that $\mathit{Dec}(\mathit{sk}, \mathit{Enc}(\mathit{crs}, pk, m, x)) = m$ for any $x$ such that $p_{\mathit{sk}}(x) = 1$. Let $(\mathit{sk}_0, \mathit{sk}_1) \leftarrow \mathit{Split}(\mathit{sk}, q)$. We have that $\mathit{sk}_0 = (l|(pk_{k+1}, pk'_{k+1}, q, \sigma), \mathit{sk}_{k+1})$ and $\mathit{Ver}(\mathit{crs}, pk_k, (pk_{k+1}, pk'_{k+1}, q), \sigma_{k+1}) = 1$. Moreover, for any $r$, we have that $\mathit{Ver}(\mathit{crs}, pk_{k+1}, r, \mathit{Sign}(\mathit{sk}_{k+1}, r)) = 1$ with overwhelming probability. It follows that for any $x$ such that $p_{\mathit{sk}}(x) = 1$ and $q(x) = 1$, the pair

$$((l|(pk_{k+1}, pk'_{k+1}, q, \sigma)), \mathit{Sign}(\mathit{sk}_{k+1}, r)) \in R_L$$

and by correctness of the underlying witness encryption, decryption succeeds. Similarly for $\mathit{sk}_1$.

*Security.* To argue security, we will use a witness extractor. Assume for the sake of contradiction, that there exists an adversary $\mathcal{A}, \mathcal{A}_0, \mathcal{A}_1$ that breaks unclonability with non-negligible probability.

Define a hybrid game in which the two ciphertexts $c_0, c_1$ are encrypted using randomness $r_0 \neq r_1$ respectively. This hybrid is information theoretically indistinguishable from the original game, since the probability that $r_0 = r_1$ is negligible.

Using the existence of $\mathcal{A}_0, \mathcal{A}_1$, the extractable security of witness encryption, and a standard forking lemma, there exist extractors $\mathcal{E}_0, \mathcal{E}_1$ such that

$$\Pr\left[\begin{array}{l|r} ((x,r_0),\mathcal{E}_0(s\mathcal{k}_0,c)) \in R_L & crs \leftarrow ParGen(1^n) \\ ((x,r_1),\mathcal{E}_1(s\mathcal{k}_1,c)) \in R_L & (m_0,m_1,s\mathcal{k}_0,s\mathcal{k}_1,pk,x) \leftarrow \mathcal{A}(crs) \\ & b \leftarrow \{0,1\}, c \leftarrow Enc\,(crs,pk,m_b,x) \end{array}\right] \geq \frac{1}{2} + \mu(n),$$

for some non-negligible function $\mu$. Let $(l_b, \sigma_b)$ be the output of the extractor $\mathcal{E}_b$. In case $l_0 = l_1$, it holds that $Ver(crs, pk_k, r_0, \sigma_0) = Ver(crs, pk_k, r_1, \sigma_1)$ or $Ver(crs, pk'_k, r_0, \sigma_0) = Ver(crs, pk'_k, r_1, \sigma_1)$, depending on whether $x$ satisfies the predicate corresponding to $pk_k$ or not. Thus, security of one-shot signatures is compromised with non-negligible probability.

In case $l_0 \neq l_1$, let $i$ be the length of the longest common prefix between $l_0$ and $l_1$ and let $q_i$ be the corresponding predicate. If $q_i(x) = 1$, the $(i+1)$'th elements in both lists contain signatures with respect to $pk_i$. If $q_i(x) = 0$, the $(i+1)$'th elements in both lists contain signatures with respect to $pk'_i$. Thus, with non-negligible probability the two witnesses contain two signatures with respect to the same public key, violating security of one-shot signatures.

## 8 Revocable Time-Released Encryption

There are several potential ways we can strengthen the notion of public-key encryption with unclonable decryption. In this section we aim for two different directions. First, we introduce the notion of delay decryption. Here an adversary not only is unable to clone the decryption key, but in fact, it can decrypt only once per some time interval $t$. Subsequently, we further strengthen the definition to allow revocation. Here one can provide a classical proof that they stopped decrypting. As long as this proof is generated before time $t$ has passed, we are sure that no-one can decrypt this ciphertext.

### 8.1 Revocable Delayed Signatures

To build revocable delayed decryption, we first extend the notion of delayed signatures to support revocation. We can then use this primitive in the encryption setting.

**Definition 10 (Revocable Delayed Signatures).** *A delay signature scheme is revocable if there are two additional algorithms $(\mathcal{R}ev, RVer)$ with the following interface:*

– $\mathcal{Rev}(sk, m) : (\pi, sk')$ *takes a quantum secret key* $sk$ *and a message* $m$ *and returns a proof* $\pi$ *that* $m$ *is revoked, and an updated key* $sk'$*.*
– $RVer(crs, pk, m, \pi) : b$ *takes a common reference string* $crs$*, a public key* $pk$*, a message* $m$ *and a proof* $\pi$ *and outputs a bit* $b$*.*

*Correctness. We require two properties:*

– *If a message has not been revoked, then it can be signed.*
– *If a message has not been signed, then it can be revoked.*

*Security. A message cannot both be revoked and signed; i.e., for any adversary* $\mathcal{A}$ *there is a negligible function* $\varepsilon$ *such that*

$$\Pr\left[\begin{array}{c} RVer(crs, pk, m, \pi) = 1 \\ Ver(crs, pk, m, rd, fd, \sigma) = 1 \end{array} \middle| \begin{array}{c} crs \leftarrow ParGen(1^n) \\ (pk, m, rd, fd, \sigma, \pi) \leftarrow \mathcal{A}(crs) \end{array}\right] \leq \varepsilon(n).$$

Notice that security of delayed signatures requires that at least time $rd$ passes before one signs a message. Therefore, if one is able to generate a revocation proof before time $rd$ passes, then we can be sure that no one will be able to sign this message.

**Theorem 6 (Revocable Delayed Signatures).** *Revocable delayed signatures exist if one-shot signatures and proofs of sequential work exist.*

*Proof.* We make use of the construction of delayed signatures by Amos et al. [3] from one-shot signatures and proofs of sequential work. In their construction, they use a delegation mechanism where the signature of one message has to include the signatures of all previous messages. As a result, their construction is also an ordered signature where each tag $t_i = t_{i-1} + 1$.

To revoke a specific message, all we have to do is to sign it along the flag $rev$ without any delays. In order to verify if a message has been revoked, we check the chain of signatures to verify that this message is in it. Moreover, we modify our signature verification scheme to reject a signature of any message $m$, if $m$ has been signed with the flag $rev$. Formally,

– $\mathcal{Rev}(sk, m)$ : Return $\mathcal{Sign}(sk, (m, rev), 0, 0)$.
– $Ver(crs, pk, m, rd, fd, \sigma)$ : Run the original verification. If it accepts, parse $\sigma$ as a list of signatures $(m_i, \sigma_i, rd_i, fd_i)$ and accept if $m_i \neq (m, rev)$ for all $i$.
– $RVer(crs, pk, m, \pi)$ : Run the original verification. If it accepts, parse $\pi$ as a list of signatures $(m_i, \sigma_i, rd_i, fd_i)$ and accept if $m_i = (m, rev)$ for some $i$.

To prove correctness, notice that if $(m, rev)$ has never been signed then by correctness of the original delayed signature the new verification accepts. Moreover, if $(m, rev)$ has been signed then, again by correctness of the original delayed signature, $RVer$ accepts. To prove security, we make use of the fact that all previous signatures have to be included in the new one. Therefore, if $RVer$ accepts a proof for a message $m$ then $(m, rev)$ is in the list, in which case $Ver$ will reject. On the other direction, if $Ver$ accepts a signature for a message $m$ then $(m, rev)$ is not in the list, in which case $RVer$ rejects.

### 8.2 Delayed Decryption

We move one to define a delayed decryption scheme. Similarly to the signature case, the encryption and decryption algorithms are parameterized by two integers $rd, fd$, corresponding to time before and after a decryption.

**Definition 11 (Delayed Decryption).** *A delayed decryption scheme is an unclonable encryption scheme with the following modifications.*

- $Enc(crs, pk, m, rd, fd) : c$ *the encryption algorithm takes additionally two non-negative integers* $rd, fd$ *and outputs a ciphertext* $c$*.*

*Correctness. The following holds with overwhelming probability. For any integers* $rd, fd$*,* $\mathcal{Dec}(sk, Enc(crs, pk, m, rd, fd)) = m$*.*

*Security. Similarly to delayed signatures, we want the following two properties:*

- *One cannot decrypt a message in time less than* $rd$*. Formally, for any adversary* $\mathcal{A}, \mathcal{A}'$ *and any constant* $\alpha$ *there exists a negligible function* $\varepsilon$ *such that*

$$\Pr\left[\mathcal{A}'(sk, c) = b \;\middle|\; \begin{array}{r} crs \leftarrow ParGen(1^n) \\ (m_0, m_1, rd, fd, pk, sk) \leftarrow \mathcal{A}(crs) \\ b \leftarrow \{0,1\}, c \leftarrow Enc(crs, pk, m_b, rd, fd) \end{array}\right] \leq \frac{1}{2} + \varepsilon(n),$$

  *where* $\mathcal{A}'$ *runs in time* $(1-\alpha)rd$*.*
- *One cannot decrypt all ciphertexts in time less than* $\sum_i rd_i + fd_i - \max_i fd_i$*. Formally, for any adversary* $\mathcal{A}, \mathcal{A}'$ *and constant* $\alpha$ *there exists a negligible function* $\varepsilon$ *such that*

$$\Pr\left[\mathcal{A}'(sk, (c_i)_i) = (b_i)_i \;\middle|\; \begin{array}{r} crs \leftarrow ParGen(1^n) \\ ((rd_i, fd_i)_{i\in[k]}, m_0, m_1, pk, sk) \leftarrow \mathcal{A}(crs) \\ \forall i \in [k], b_i \leftarrow \{0,1\} \\ c_i \leftarrow Enc(crs, pk, m_{b_i}, rd_i, fd_i) \end{array}\right] \leq \frac{1}{2} + \varepsilon(n),$$

  *where* $\mathcal{A}'$ *runs in time* $(1-\alpha)\left(\sum_i rd_i + fd_i - \max_i fd_i\right)$*.*

Notice that this definition trivially implies unclonable decryption, since if one could clone the decryption key then they could decrypt two messages in parallel in time $\max\{rd_1, rd_2\} < rd_1 + rd_2$.

**Proposition 1.** *An encryption scheme with delayed decryption is also an encryption scheme with unclonable decryption.*

### 8.3 Classically Revocable Decryption

Similarly to delayed signatures, public-key encryption with classically revocable decryption is a delay encryption scheme equipped with an additional revoking functionality. By revoking a specific ciphertext, we irreversibly update our secret key into a new one that is unable to decrypt this ciphertext. For such a definition to make sense we have to update our key before time $t$ has passed. Otherwise, we can decrypt and then rewind and revoke this ciphertext.

**Definition 12 (Classically Revocable Decryption).** *An encryption scheme with classically revocable decryption is a delay encryption scheme with two additional algorithms* $(\mathcal{Rev}, Ver)$ *with the following interface:*

- $\mathcal{Rev}(s\mathcal{k}, c) : (\pi, s\mathcal{k}')$ *takes a quantum secret key* $s\mathcal{k}$*, a ciphertext* $c$ *and outputs a proof* $\pi$ *and an updated secret key* $s\mathcal{k}'$*.*
- $Ver(crs, pk, c, \pi) : b$ *takes a common reference string* $crs$*, a ciphertext* $c$ *and a proof* $\pi$ *and outputs a bit* $b$*.*

*Correctness. Additionally, the following hold with overwhelming probability.*

- *If a ciphertext has not been revoked then it can be decrypted.*
- *If a ciphertext has not been decrypted then it can be revoked.*

*Security. A ciphertext cannot be decrypted after being revoked in time; i.e., for any quantum polynomial time adversary* $\mathcal{A}, \mathcal{A}', \mathcal{A}''$ *and any constant* $\alpha$ *there is a negligible function* $\varepsilon$ *such that*

$$\Pr\left[\begin{array}{c} RVer(crs, pk, c, \pi) = 1 \\ b = b' \end{array} \middle| \begin{array}{r} b \leftarrow \{0,1\}, crs \leftarrow ParGen(1^n) \\ (m_0, m_1, rd, fd, pk, s\mathcal{k}) \leftarrow \mathcal{A}(crs) \\ c \leftarrow Enc(crs, pk, m_b, rd, fd) \\ (\pi, s\mathcal{k}') \leftarrow \mathcal{A}'(s\mathcal{k}, c) \\ b' \leftarrow \mathcal{A}''(s\mathcal{k}', c) \end{array}\right] \leq \frac{1}{2} + \varepsilon(n),$$

*where* $\mathcal{A}'$ *runs in time* $(1-\alpha)rd$*.*

**Theorem 7.** *If classically revocable proofs of sequential work and extractable witness encryption exist, then public key encryption with classically revocable decryption exists.*

*Proof.* Without loss of generality, we make the assumption that a witness extractor in our witness encryption scheme, runs in the same time as that of the distinguisher. This is to simplify the proofs. Without this assumption, we would have to incorporate any additional delay posed by the extractor into the construction of our encryption scheme.

Let $(ParGen', \mathcal{Gen}', \mathcal{Sign}', Ver', \mathcal{Rev}', RVer')$ be a revocable signature scheme. We define the corresponding encryption algorithms:

- $ParGen(1^n)$ : Return $crs \leftarrow ParGen'(1^n)$.
- $\mathcal{Gen}(crs)$ : Return $(pk, s\mathcal{k}) \leftarrow \mathcal{Gen}'(crs)$.
- $Enc(crs, pk, m, rd, fd)$ : Pick randomness $r$ and return $(r, rd, fd, Enc'(m))$, where $Enc'$ is a witness encryption for $\{(r, w) : Ver'(crs, pk, r, rd, fd, w) = 1\}$.
- $\mathcal{Dec}(s\mathcal{k}, (r, rd, fd, c))$ : Generate $w \leftarrow \mathcal{Sign}(s\mathcal{k}, r, rd, fd)$ and return $Dec'(c, w)$. Subsequently, rewind to retrieve the original key.
- $\mathcal{Rev}(s\mathcal{k}, (r, rd, fd, c))$ : Return $(\pi, s\mathcal{k}') \leftarrow \mathcal{Rev}'(s\mathcal{k}, r)$.
- $RVer(crs, pk, (r, rd, fd, c), pk)$ : Return $b \leftarrow RVer'(crs, pk, r, \pi)$.

Correctness follows from the correctness of revocable signatures and the correctness of witness encryption. We go on to prove our two security properties; namely delayed security and revocable security.

*Delayed Security.* Suppose that there is a constant $\alpha$ and an adversary $\mathcal{A}, \mathcal{A}'$ that can distinguish between encryptions of two messages in less than $(1-\alpha)rd$ time. $\mathcal{A}'$ implies the existence of an extractor $\mathcal{E}$ that can find a signature for the randomness incorporated in the ciphertext. We create an adversary $\mathcal{B}, \mathcal{B}'$ that can sign a random message in almost the same time as follows:

- $\mathcal{B}(crs)$ : Generate $(m_0, m_1, rd, fd, pk, s\mathcal{k}) \leftarrow \mathcal{A}(crs)$ and return the tuple $(rd, fd, pk, (m_0, m_1, s\mathcal{k}))$ where the last tuple is the secret key for $\mathcal{B}'$.
- $\mathcal{B}'((m_0, m_1, s\mathcal{k}), (r, rd, fd, c))$ : Run $w \leftarrow \mathcal{E}(r, (m_0, m_1, c, rd, fd, s\mathcal{k}))$ and return $w$.

By extractable security of the witness encryption, $Ver'(crs, pk, r, fd, rd, w) = 1$ with non-negligible probability. Moreover, since the running time of $\mathcal{E}$ is that of $\mathcal{A}'$, we conclude that $\mathcal{B}'$ runs in time $(1-\alpha)rd + c$, for constant $c$.

Moreover, suppose that there is a constant $\alpha$ and an adversary $\mathcal{A}, \mathcal{A}'$ that can distinguish encryptions of two messages in time $(1-\alpha)(\sum_{i\in[l]} rd_i + fd_i - \max_{i\in[l]} fd_i)$. Again, by invoking a signature extractor from the distinguisher $\mathcal{A}'$ we are able to sign all $l$ random messages in time $(1-\alpha)(\sum_{i\in[l]} rd_i + fd_i - \max_{i\in[l]} fd_i) + c$, for constant $c$ reaching a contradiction.

*Revocable Security.* Assume adversaries $\mathcal{A}, \mathcal{A}', \mathcal{A}''$ such that although $\mathcal{A}'$ revokes a ciphertext in time less than $rd$, $\mathcal{A}''$ is still able decrypt it subsequently with non-negligible probability. We will create an adversary $\mathcal{B}$ that can both sign and revoke a message with non-negligible probability. The adversary $\mathcal{B}$ first runs $(m_0, m_1, rd, fd, pk, s\mathcal{k}) \leftarrow \mathcal{A}(crs)$ then picks a randomness $b, r$ and computes $c \leftarrow Enc(crs, pk, m_b, rd, fd)$. Subsequently, it runs $(\pi, sk') \leftarrow \mathcal{A}'(s\mathcal{k}, c)$. Then, by security of witness encryption, it uses an extractor $\mathcal{E}(s\mathcal{k}', c)$, corresponding to the distinguisher $\mathcal{A}''$, in order to find a signature $\sigma$ for $r$. It finally returns $(pk, r, rd, fd, \sigma, \pi)$ which breaks security with non-negligible probability.

Notice that if $\mathcal{A}'$ could run for time $(1-\alpha)rd$, then it could both decrypt and rewind and subsequently generate a proof of revocation without ever learning a classical signature for the randomness $r$.

## Acknowledgements

This work is supported in part by NSF. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of NSF.

## References

1. Aaronson, S.: Quantum copy-protection and quantum money. In: 2009 24th Annual IEEE Conference on Computational Complexity. pp. 229–242. IEEE (2009)
2. Alagic, G., Fefferman, B.: On quantum obfuscation. arXiv preprint arXiv:1602.01771 (02 2016)

3. Amos, R., Georgiou, M., Kiayias, A., Zhandry, M.: One-shot signatures and applications to hybrid quantum/classical authentication. In: Proceedings of STOC 2020 (2020)
4. Ben-David, S., Sattath, O.: Quantum tokens for digital signatures (2016)
5. Bennett, C.H., Brassard, G.: Quantum cryptography: public key distribution and coin tossing. Theor. Comput. Sci. **560**(12), 7–11 (2014)
6. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. arXiv preprint arXiv:2003.06557 (2020)
7. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Ishai, Y. (ed.) Theory of Cryptography. pp. 253–273. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
8. Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Annual Cryptology Conference. pp. 361–379. Springer (2013)
9. Brakerski, Z., Christiano, P., Mahadev, U., Vazirani, U., Vidick, T.: A cryptographic test of quantumness and certifiable randomness from a single quantum device. In: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS). pp. 320–331. IEEE (2018)
10. Broadbent, A., Lord, S.: Uncloneable quantum encryption via oracles (2019)
11. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Desmedt, Y.G. (ed.) Advances in Cryptology — CRYPTO '94. pp. 257–270. Springer Berlin Heidelberg, Berlin, Heidelberg (1994)
12. Döttling, N., Lai, R.W., Malavolta, G.: Incremental proofs of sequential work. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 292–323. Springer (2019)
13. Garg, S., Gentry, C., Sahai, A., Waters, B.: Witness encryption and its applications. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing. pp. 467–476 (2013)
14. Gavinsky, D.: Quantum money with classical verification. In: 2012 IEEE 27th Conference on Computational Complexity. pp. 42–52. IEEE (2012)
15. Goldwasser, S., Kalai, Y.T., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: How to run turing machines on encrypted data. In: Annual Cryptology Conference. pp. 536–553. Springer (2013)
16. Gottesman, D.: Uncloneable encryption. arXiv preprint quant-ph/0210062 (2002)
17. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. Proceedings of the ACM Conference on Computer and Communications Security pp. 89–98 (12 2006). https://doi.org/10.1145/1180405.1180418
18. Mahmoody, M., Moran, T., Vadhan, S.: Publicly verifiable proofs of sequential work. In: Proceedings of the 4th conference on Innovations in Theoretical Computer Science. pp. 373–388 (2013)
19. Pastawski, F., Yao, N.Y., Jiang, L., Lukin, M.D., Cirac, J.I.: Unforgeable noise-tolerant quantum tokens. Proceedings of the National Academy of Sciences **109**(40), 16079–16082 (2012)
20. Radian, R.: Semi-quantum money. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies. pp. 132–146 (2019)
21. Unruh, D.: Revocable quantum timed-release encryption. Journal of the ACM (JACM) **62**(6), 1–76 (2015)
22. Wiesner, S.: Conjugate coding. ACM Sigact News **15**(1), 78–88 (1983)
23. Winter, A.: Coding theorem and strong converse for quantum channels. IEEE Transactions on Information Theory **45**(7), 2481–2485 (1999)

24. Wootters, W.K., Zurek, W.H.: A single quantum cannot be cloned. Nature **299**(5886), 802–803 (1982)
25. Zhandry, M.: Quantum lightning never strikes the same state twice. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 408–438. Springer (2019)