Depth-Bounded Quantum Cryptography with Applications to One-Time Memory and More

Qipeng Liu ✉
Simons Institute for the Theory of Computing, Berkeley, CA, USA

Abstract

With the power of quantum information, we can achieve exciting and classically impossible cryptographic primitives. However, almost all quantum cryptography faces extreme difficulties with the near-term intermediate-scale quantum technology (NISQ technology); namely, the short lifespan of quantum states and limited sequential computation. At the same time, considering only limited quantum adversaries may still enable us to achieve never-before-possible tasks.

In this work, we consider quantum cryptographic primitives against limited quantum adversaries – depth-bounded adversaries. We introduce a model for (depth-bounded) NISQ computers, which are classical circuits interleaved with shallow quantum circuits. Then, we show one-time memory can be achieved against any depth-bounded quantum adversaries introduced in the work, with their depth being any pre-fixed polynomial. Therefore we obtain applications like one-time programs and one-time proofs. Finally, we show our one-time memory has correctness even against constant-rate errors.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols; Security and privacy → Authorization; Security and privacy → Public key (asymmetric) techniques

Keywords and phrases cryptographic protocol, one-time memory, quantum cryptography

Digital Object Identifier 10.4230/LIPIcs.ITCS.2023.82

Funding *Qipeng Liu*: supported in part by the Simons Institute for the Theory of Computing, through a Quantum Postdoctoral Fellowship and by the DARPA sieve-vespa grant HR00112020023.

Acknowledgements The authors would like to thank Shafi Goldwasser for so many insightful discussions. Without whom, this work would be impossible.

1 Introduction

Quantum information paves the way for many classically impossible cryptographic applications. Wiesner initiated the study and proposed the first application: quantum money [47] – banknotes that can not be copied. Recently, we have seen more exciting applications, including unclonable banknotes that can be publicly verified [2, 49, ...] and copy-protection of programs [1, 3, ...]. Despite the great success in the theoretical study of these protocols, almost all applications face extreme difficulties with state-of-the-art quantum technology. There are several barriers to implementing these exciting applications, most notably, quantum error-correction and thus the ability to

(i) maintain a quantum state for a long time and
(ii) coherently compute over a system for a large depth.

To illustrate, we can think of quantum money. For now, we focus on the quantum money schemes based on subspace states [2, 49], it consists of two major building blocks: (1) a quantum state treated as a banknote and (2) a quantum procedure for verifying the validity of money. The banknote needs to be persistent, as no one wants their wealth to evaporate overnight. The quantum money scheme will also become utterly useless if a verification procedure is so complicated that no quantum computers can finish the evaluation with little error and make no disturbance to the banknote. Similarly, requirements (i) and (ii) are required for almost all quantum cryptography protocols.

The current quantum technology fails to satisfy either (i) or (ii), meaning all the groundbreaking applications are still far from achievable in practice. The experiments that do show quantum advantages, for example, [33, 5], only need noisy quantum computers with short depth (i.e., bounded steps of sequential computation). While waiting for the arrival of full-scale, fully fault-tolerant quantum computers, Preskill [39] proposed the notion of Noisy Intermediate-Scale Quantum (NISQ) technology and discussed the impact of NISQ computers on the world, like quantum simulation and quantum annealing. In this worse case, just like quantum skeptics may argue [35], a good-quality quantum error-correction code might never exist and invalidate either requirements (i) or (ii). Suddenly, all efforts devoted to full-scaled quantum computers and their applications vanish. Therefore, it is intriguing to ask the following question:

> Can we take advantage of the *imperfect form* of quantum computers and build cryptographic protocols that one can not achieve in a classical world, and yet are achievable and secure for NISQ computers?

Cryptography is always about utilizing asymmetry between honest parties and malicious adversaries. More precisely, the above question wants to exploit quantum computers' limitations to restrict adversaries but keep the quantum operation for honest parties as simple as possible. Any answer to the above question demonstrates a *win-win* situation for the current quantum cryptography research. If perfect quantum computers never exist and there is a depth bound for maximal sequential computation, we can still base cryptography on the limitations of quantum computers and achieve classically impossible primitive; otherwise, we enter the new era of quantum cryptography with applications like quantum money.

In this work, we answer the question affirmatively. We formally define "imperfect quantum computers" as classical circuits with access to depth-bounded quantum circuits (hybrid circuit models, inspired by [20]) and show powerful applications like one-time memory and one-time programs [31].

1.1 Result 1: Defining Depth-Bounded Quantum Adversaries

Our first result is to give a model that captures quantum computers with all the limitations we discussed above.

To capture NISQ computers, a reasonable model should reflect both the short lifespan of quantum states and shallow sequential quantum computation. The first thought is to consider d-depth bounded quantum circuits for a fixed polynomial d, where d takes the size of the whole system as a parameter, denoted by QNC_d. Although this definition captures "shallow sequential quantum computation", it fails to represent the ability of large sequential classical computation and the resettability of quantum computers.

Chia, Chung, and Lai [20] give two formal definitions of hybrid classical-quantum circuits, corresponding to two different computational models. The first one is called d-CQ circuits: a hybrid circuit is modeled as a QNC_d circuit with oracle access to a polynomial-time classical machine in BPP. It captures the ability to arbitrarily pause a quantum machine at any stage of its computation, post-processing based on partial results of the quantum computer and resuming the quantum computation. This model describes "shallow depth" but not "short lifespan of quantum states". Because there is no particular time limit on the running time of classical computation, the quantum computer in this model should perfectly preserve its internal state for an arbitrarily (polynomially) long time, which is in contrast to the intuition behind NISQ.

We found their second model is more natural to our setting: d-QC circuits. This model describes a hybrid circuit as a BPP machine for *classical input-output* behavior with oracle access to QNC_d. To put it another way, a quantum computer in this model is mostly a classical machine, except when needed, it initiates and runs an instance of a shallow quantum machine. We propose the following model, which is similar to the d-QC circuits with one difference: in our model, a hybrid circuit computes *quantum input-output* functionality. See Figure 1 for more details.

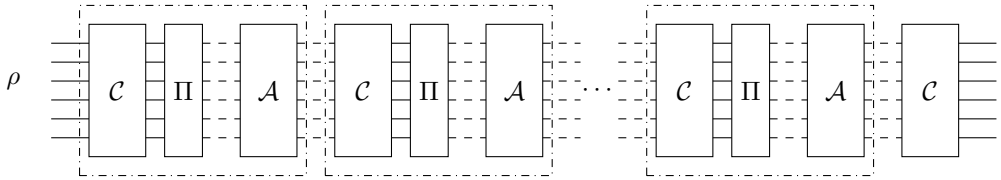

Figure 1 d-depth bounded quantum algorithms. $\mathcal{C}$ denotes quantum circuits, Π denotes measurements whereas $\mathcal{A}$ denotes classical post-processing. Dashed lines denote classical bits and solid lines denote qubits. There are polynomially many such blocks, but each $\mathcal{C}$ is at most of depth d.

In our model, shallow quantum circuits are interleaved with full measurements[1] and classical processing. Note that in our model, full measurements are crucial: first, if only a partial measurement is applied, the quantum depth of an algorithm will get at least doubled; second, the algorithm has a pure classical state at some point of its computation is essential to establish the technical proof later in the work.

Our model inherently assumes that at some stage of NISQ era, there exist two parameters $d \gg d'$ such that

- Computation over quantum states and communication can be done with no error (or some constant-rate error) for depth d' and time d';
  d' is the transmission time[2] of quantum information from one party to another plus the number/time of sequential computation for honest execution of protocols. In our work, we set d' as a constant.
- It is impossible to compute over a quantum system for d units of time.
  In this work, we will set d as any pre-fixed polynomial.

We believe our model captures the nature of NISQ computers, for the two fundamental limitations we care about: the short lifespan of quantum states and shallow sequential quantum computation. In the rest of the work, we will work with these definitions and build cryptographic primitives against d-depth bounded quantum algorithms for any pre-fixed polynomial d.

1.2 Result 2: Constructing One-Time Memory and One-Time Programs

Next, we show our second result: constructions of one-time memory against depth-bounded quantum computers.

The study of one-time memory (OTM) and one-time programs was initiated by Goldwasser, Kalai and Rothblum [31]. One-time memory is a powerful building block and can be viewed as a central hub of cryptographic protocols. A one-time memory is a piece of hardware

[1] A full measurement measures every single qubit of the system and leaves the internal state purely classical.

[2] We only consider earth-to-earth transmission, which takes much less than one sec and can be treated as a constant. For applications involved earth-to-space quantum information transmission, assuming d being relatively small may not be ideal.

that stores two values m_0 and m_1; honest clients can read out any of the values from the one-time memory, while any read operation will destroy the memory and make it impossible to read the other value. With one-time memory, Goldwasser et al. showed how to build one-time programs, which is a one-time version of virtual black-box obfuscation (VBB)[3][6]. A client can use a one-time program of any function f to compute $f(x)$ on any x of its choice without the ability to see either the actual computation or $f(y)$ on other $y \neq x$. Since VBB obfuscation is the "holy grail" of cryptography, many classically complicated or impossible functionality can be constructed with OTMs in a generic way, such as one-time programs and one-time zero-knowledge proofs [31].

Despite the power of one-time memory, one-time memory itself is clearly impossible in the classical setting without making any hardware assumptions . If a piece of classical bits encodes a one-time memory, an adversary can always read one value, rewind it to the original state and read the other value. Due to the gentle measurement lemma [4], such rewinding can also be done in the quantum world; one-time memory can not purely base its security on quantum information and computational assumption. Therefore, some hardware assumptions are still needed – in other words, certain limitations on adversaries are necessary to achieve one-time memory.

Broadbent, Gharibian, and Zhou [15] started the study on quantum OTMs with classical hardware called stateless tokens (later improved by [8]). A stateless token encodes a classical functionality in a black-box manner; i.e., one can only learn the classical input-output behavior of the functionality but nothing else, just like a virtual black-box obfuscation. Even more, since it is a piece of hardware, they further assume an adversary does not have enough energy to coherently compute over a macroscopic object (the token) and thus can only classically interact with the token. Chung, Georgiou, Lai, and Zikas [23] also followed the same idea and used classical tokens to implement OTMs and other applications. However, the assumption on VBB obfuscation and the ability to only classical evaluation over the obfuscation is less satisfying, mainly due to two reasons: (1) general-purpose VBB obfuscation is widely known to be impossible [6], and all heuristic obfuscation is more or less defective and far away from VBB; (2) an attacker can learn some bits of information about the hardware and may perform quantum computation based on the piece of information, which is not captured by "classical interaction".

In this work, we provide provable OTMs against depth-bounded quantum adversaries. In some sense, our constructions are still under some hardware assumptions: "depth-bounded" limits how a quantum computer can be built; but the assumption is rather mild for NISQ computers.

▶ **Theorem 1** (OTM, informal). *There exist secure OTMs with un-entangled quantum states, against d-depth-bounded quantum adversaries for any fixed polynomial $d(\cdot)$, in the quantum random oracle model.*

Let us elaborate more on the theorem. Our construction is a prepare-and-measure protocol, meaning all quantum computation for honest senders and receivers is to prepare un-entangled quantum states and measure under computational or Hadamard bases. The theorem provides a strong security: honest evaluation is a constant-depth quantum algorithm that only queries a random oracle classically; however security holds even if an adversary is d-depth bounded

[3] A virtual black-box obfuscation is a program compiler that takes a classical circuit and outputs an obfuscated version of that circuit; any algorithm can only interact with the obfuscation by observing its input-output behavior, but nothing else. This primitive is impossible in both the classical and quantum world.

and each superposition query to the random oracle is treated as a single quantum gate. By appropriately replacing the random oracle with a practical hash function, like SHA-3, we get a heuristic instantiation of this protocol. As discussed in the last section, we believe bounded depth characterizes two critical limitations of near-term quantum computers and is mild compared to stateless tokens (classically accessible VBB).

Applications

We can also leverage our one-time memory to one-time programs and one-time proofs in a generic way, following the ideas in [31]. A one-time program compiler takes a classical circuit C and outputs a (quantum) token. Any user with the token can compute C on any input x but only once. With one-time memory, one-time programs can be constructed in a generic way from Yao's garbled circuits [48]. Yao's garbling procedure takes a circuit C and outputs a list of input labels $\mathsf{inp}_{i,b}$ for $i \in [n], b \in \{0,1\}$; with any combination of $\mathsf{inp}_{1,x_1}, \mathsf{inp}_{2,x_2}, \cdots, \mathsf{inp}_{n,x_n}$, one can compute $C(x)$ but learn nothing except $C(x)$. To construct one-time programs, we can use our one-time memories to store each pair of $\mathsf{inp}_{j,0}$ and $\mathsf{inp}_{j,1}$. Since a user can only read out exactly one of each pair of input labels, both completeness and security hold. The resulting one-time program compiler is a prepare-and-measure protocol with un-entangled quantum states, with honest execution taking $O(1)$ quantum depth.

1.3 Result 3: Dealing with Constant Errors

In the above discussion, we must assume honest execution and transmission have no error. This may be an unrealistic assumption for NISQ techniques; for example, the experiment demonstrated in [5] has constant-rate ($< 1\%$) errors. Therefore, we model errors as depolarizing quantum channels that operate independently on each qubit, and preserve each qubit with probability $1-\varepsilon$ and replace the qubit with a maximally mixed state with probability ε for some constant $\varepsilon > 0$. We prove that our scheme still has completeness under this noise model while achieving security even against errorless depth-bounded quantum adversaries. To achieve this, we apply certain classical error-correcting codes (like expander codes [41, 19] or Justesen codes [34]) on quantum states. The resulting quantum state is still un-entangled and satisfies all the properties we want. We elaborate in the next section.

1.4 Technical Overview

The starting point is the famous Wiesner states [47] (or BB84 states [11]). A Wiesner state encodes a bit of classical information x under a special basis $\theta \in \{0,1\}$, denoted by $|x^\theta\rangle$.

$$|x^\theta\rangle = \begin{cases} |0\rangle & \text{and } x=0, \theta=0, \\ |1\rangle & \text{and } x=1, \theta=0, \\ |+\rangle & \text{and } x=0, \theta=1, \\ |-\rangle & \text{and } x=1, \theta=1, \end{cases}$$

When θ is 0, x is encoded by the set of computational basis states $|0\rangle$ or $|1\rangle$; when $\theta = 1$, the set of encoding is the Hadamard basis $|+\rangle$ or $|-\rangle$. For x and θ being n-bit classical strings, we use $|x^\theta\rangle$ to denote the concatenation of Wiesner states: $|x^\theta\rangle := |x_1^{\theta_1}\rangle |x_2^{\theta_2}\rangle \cdots |x_n^{\theta_n}\rangle$. In this work, we will mostly focus on the case where θ has hamming weight $n/2$ (assuming n is even). In this case, a Wiesner state encodes two classical $n/2$-bit strings on each basis. We call them x_c in the computational basis and x_h in the Hadamard basis. For example,

when $x = 0101$ and $\theta = 0110$, x_c will be 01, corresponding to the bits in x whose matching coordinates in θ are 0; and x_r is 10. Here we ignore the subscript or superscript θ on x_c, x_r for convenience.

Wiesner states enjoy three important properties.

1. The first one is "**unpredictability**". No (even unbounded) algorithm given $|x^\theta\rangle$ but no θ, can output either x_c or x_h. It says a Wiesner state can be viewed as symmetric-key quantum encryption of classical bits, which completely hides a random message x.
2. The second property is called "**direct-product hardness**", which says without giving θ, (even unbounded) one can not produce two strings such that one contains x_c in the corresponding coordinates and the other contains x_h. Unlike unpredictability, outputting a string that consists of x_c or x_h is easy. One can simply measure the quantum state $|x^\theta\rangle$ in either the computational or Hadamard basis; but outputting exact x_c or x_h is hard because it does not know which $n/2$ coordinates to pick. The "direct-product hardness" says that while obtaining a classical string with x_c or x_h as a substring at the right coordinates is easy, it is impossible to produce two strings with both x_c and x_h at the correct positions, respectively.
3. The last property is "**unclonability**". The security guarantee says no (even unbounded) splitting attacker Alice can produce two (potentially entangled) quantum states given $|x^\theta\rangle$, later two non-communicating parties recover x_c and x_h respectively, both with the basis information θ. Coladangelo, Liu, Liu and Zhandry [24] proposed a conjecture on this property[4] on coset state (a generalization of Wiesner states) and it is later proved by Culf and Vidick [26] for both Wiesner states and coset states.

The construction by Broadbent et al. [15] is, roughly speaking, based on Wiesner states' direct-product hardness. In their construction, a classical token computes the following functionality: if the input consists of x_c at the correct coordinates, it outputs m_0; if the input consists of x_h at the right coordinates, it outputs m_1; otherwise, it outputs a junk symbol $\perp$. Here m_0 and m_1 are messages they want to put in a one-time memory. The proof roughly says, if an algorithm can produce m_0, then it must query the classical token on a classical message with x_c at the correct coordinates; similarly for m_1 and x_h. Therefore, having the ability to recover both m_0 and m_1 implies an attack on direct-product hardness. Their proof crucially relies on the token being classical, i.e., a quantum algorithm can only interact with it classically and cannot compute coherently over it, which forces measurement over the quantum state.

Unlike the construction by Broadbent et al., which bases on direct-product hardness, our construction relies on unpredictability and unclonability, as well as a cryptographic primitive called weak time-lock puzzles [40, 12]. For convenience, we use "time-lock puzzles" to denote "weak time-lock puzzles" in the rest of the work. Time-lock puzzles (alternatively called time-released encryption) are encryption that completely hides the message before some time limit t for any polynomial-parallel machines, and decryption takes roughly $T \approx t$ time. The encryption requires circuits of size T and depth $\log(T)$. In the work by Bitansky et al. [12], they showed that classical time-lock puzzles exist assuming one-way functions and non-parallelizing languages. The reduction works in the same way for quantum. Therefore, post-quantum time-lock puzzles exist, assuming post-quantum weak one-way functions and quantum non-parallelizing languages whose existence is supported by the evidence in [22].

[4] They call it "strong monogamy-of-entanglement".

Our construction is conceptually intuitive. Let us focus on the single-bit case: a one-time memory stores two single bits. We first generate a Wiesner state $|x^\theta\rangle$ with $f(x_c)$ equals to the first message m_0 and $f(x_h)$ equals to the second message m_1, here f is a cryptographic hash function with single-bit outputs and will be treated as a random oracle in the proof. A time-lock puzzle Z on the secret basis information θ is also generated, along with the quantum state. The one-time memory is then $(|x^\theta\rangle, Z)$ (see Figure 2)[5]. Clearly, the generation procedure only takes $O(1)$ quantum depth as it only needs to apply $O(n)$ parallel Hadamard gates to prepare $|x^\theta\rangle$ and the preparation of Z can be done in a classical circuit, before preparing $|x^\theta\rangle$. By the definition of bounded-depth hybrid circuits in Figure 1, it can be computed in $O(1)$-depth hybrid circuits.

$\mathsf{Token.Gen}^f(m_0, m_1)$:

- Sample a uniformly random $\theta \leftarrow \{0,1\}^n$ with hamming weight $n/2$.
- Sample $x \leftarrow \{0,1\}^n$ such that $f(x_c) = m_0$ and $f(x_h) = m_1$.
- Let $Z \leftarrow \mathsf{Puzzle.Gen}(d, \theta)$ be a post-quantum time-lock puzzle with solution θ.
- Output $|\mathsf{tk}\rangle = (|x^\theta\rangle, Z)$.

Figure 2 One-Time Memory Construction for Single-Bit, Simplified.

To read m_b, one first measures the quantum state $|x^\theta\rangle$ under the computational basis or the Hadamard basis depending on b. Then classical post-processing reveals θ for picking the right coordinates. It can also be done by an $O(1)$-depth hybrid circuit.

The security follows the intuition below. Before a hybrid circuit goes deeper than d, the basis information θ should be completely hidden from an adversary, and the adversary has no way to interpret $|x^\theta\rangle$ in any meaningful way (unpredictability of Wiesner states). After depth d, the quantum state collapses and leaves only one out of two messages on both bases (unclonability of Wiesner states). Thus, no adversary with a depth smaller than d can read out both m_0 and m_1 simultaneously.

To formally prove it, we need to tweak the construction and prove the security in the quantum random oracle model. Ignoring many details here, the key idea is as follows. We show that if a d-depth bounded adversary can read out two values, we can turn it into an adversary for the unclonability of Wiesner states. Recall Figure 1, when the first quantum circuit $\mathcal{C}$ of depth d finishes, a measurement is applied, leaving the internal state of the algorithm purely classical. At this point, we can copy the adversary into two identical ones, and both recover m_0, m_1. With two identical adversaries, we use one for x_c and the other for extracting x_h, violating the unclonability. The random oracle is used to remove the dependence on θ of the first quantum circuit and extractions of x_c and x_h. We leave all the missing details in the formal proofs in Section 4.

Finally, when there is a constant-rate error, honest execution can no longer produce x_c or x_h as a constant fraction of all qubits gets corrupted. Our solution is to have $\mathsf{ECC}(x_c)$ and $\mathsf{ECC}(x_h)$ encoded under Wiesner states, instead of the original x_c and x_h; here ECC is some *classical* error-correcting code. Not all error-correcting codes work in our setting as their rate and alphabet size affect the "unpredictability" and "unclonability" of the resulting

[5] The construction is similar to revocable time-release encryption by Unruh [45] but the goal and proof techniques are quite different. We will mention it at the end of this section.

quantum states. To establish all the properties, we need a classical error-correcting code with a constant rate, constant relative distance, and constant alphabet size. It turns out expander codes satisfy all three properties. We name the new quantum state as Wiesner states on expander codes (WSECs) and prove all the properties in the full version.

1.5 Other Related Work

Depth-Bounded Cryptography

Cryptography against bounded-depth adversaries in the classical setting has already been studied by [32, 29, 30, 27, 25, 28, ...]. Degwekar, Vaikuntanathan and Vasudevan [27] introduced the notion of fine-grained cryptography, which is cryptography against low-depth adversaries (or adversaries with bounded resources in general). They considered adversaries being either AC^0 or NC^1. The limitation of adversaries enables cryptographers to weaken or even remove unproven assumptions. Our new notion can be viewed as generalizing depth-bounded cryptography in the quantum world. Whereas classical depth-bounded cryptography is important to theory, we find our quantum analog captures a broad class of adversaries and has more motivation for practical purposes.

Revocable Time-Released Quantum Encryption [45]

Unruh also discussed combining time-lock puzzles and Wiesner states to achieve classically impossible primitives, called revocable time-released encryption. It is an encryption scheme that encodes a classical message m into a quantum state ρ_m. It takes an honest client time t to recover the message m from ρ_m. Before time t, the server (who sends out the encryption) can ask the client to send the encryption back to it. If the server verifies the encryption is sent back appropriately, the client has no way to learn any information about m.

Unruh used a Wiesner state to encode a message m and a time-lock puzzle to store basis information for the Wiesner state. Time-lock puzzles enforce the basis information is only learned by the client after time t. Some (weaker) form of unclonability of Wiesner states is used for revocability. Although our construction is similar to Unruh's and both achieve classically impossible primitives, there are many differences between these two works:

- The goals are fundamentally different. Their goal is to have revocability before time t, while ours is to have one-time memory against depth-bounded adversaries.
- The honest execution in their work takes quantum depth t (as defined in Figure 1) whereas ours takes constant quantum depth.
- Theirs relies on a weaker form of unclonability, while ours relies on the so-called "strong monogamy-of-entanglement".
- The proofs rely on different techniques.

The similarities shared in the two works (as well as Broadbent et al. [15]) are not surprising. Because Wiesner states are the most simple yet fundamental unclonable quantum information and lead to many applications; like unclonable encryption by Broadbent and Lord [17], certifiable deletion by Broadbent and Islam [16] and most recently by Bartusek and Khurana [7] and many more.

Verification of Quantum Depth [21]

A recent work by Chia and Hung gives two protocols for verifying quantum depth by classical verifiers. Our protocol can also be viewed as an alternative verification of quantum depth: if an adversary can recover both m_0, m_1, it must have a large depth. However, if we turn our protocol into a verification protocol of quantum depth, the verifier is not purely classical; it needs minimal quantum ability to prepare Wiesner states.

1.6 Paper Organization

In the next section, we will touch on the bases of the quantum random oracle model, Wiesner states, time-lock puzzles, and other useful lemmas. In Section 3, we introduce the computational model: bounded depth quantum adversaries. We then formally define one-time memory through a simulator-based definition and prove its security in Section 4. We give a noise-tolerant version of our scheme in the full version.

2 Preliminaries

In this work, when we talk about non-uniform security, non-uniform quantum adversaries are modeled by a collection of quantum circuits $\{\mathcal{C}_\lambda\}_{\lambda\in\mathbb{N}^+}$ with auxiliary quantum inputs $\{\sigma_\lambda\}_{\lambda\in\mathbb{N}^+}$. When $\mathcal{C}_\lambda$ is unbounded, we ignore σ_λ for convenience, as it can be prepared by $\mathcal{C}_\lambda$.

For $n \in \mathbb{N}^+$, $[n] = \{1, 2, \cdots, n\}$ and $[0, n] = \{0, 1, \cdots, n-1, n\}$. For a binary vector $\theta \in \{0,1\}^n$, its hamming weight is denoted by $|\theta|_1$. When a random variable is drawn uniformly at random from a set S, we denote by $v \leftarrow S$.

We assume that readers are familiar with the background of quantum information. Please refer to [38] for more details. For two mixed states ρ, σ, we denote their trace distance by $T(\rho, \sigma)$.

2.1 Quantum Random Oracle Model

An oracle-aided quantum algorithm can perform two types of computation: quantum computation and quantum oracle access to an oracle. Let f be an oracle that a quantum algorithm has access to. Quantum oracle access to f is a unitary on input and output registers $\mathbf{XY}$: $\mathcal{O}_f : |x, y\rangle_{\mathbf{XY}} \to |x, y + f(x)\rangle_{\mathbf{XY}}$.

In practice, we usually model a hash function as a uniformly random function $f : [M] \to [N]$, instead of a fixed function that looks "random". This is so called random oracle model, introduced by Bellare and Rogaway [9]. The quantum analog, called quantum random oracle model, was first proposed by Boneh et al. [14]. In this model, f is chosen at the very beginning. An oracle-aided quantum algorithm with oracle access to f has three registers $\mathbf{XYZ}$ and performs the following operations:

- Oracle access: a unitary on $\mathbf{XY}$ such that $\mathcal{O}_f : |x, y\rangle_{\mathbf{XY}} \to |x, y + f(x)\rangle_{\mathbf{XY}}$.
- Local operation: a unitary on $\mathbf{XYZ}$.

A T-query quantum algorithm starts with its internal memory being $|\mathbf{0}\rangle$ or an auxiliary input $|\phi_0\rangle$, and the computation is then modeled as $U_T\mathcal{O}_f \cdots \mathcal{O}_f U_1 \mathcal{O}_f U_0$ followed by a measurement. We denote an oracle-aided quantum algorithm with oracle acess to f by $\mathcal{A}^{|f\rangle}$, where $|f\rangle$ specifies its ability to make quantum queries.

We will need the following lemma for extraction queries from an oracle-aided quantum algorithm.

▶ **Theorem 2** ([10]). *Let $\mathcal{A}$ be an adversary with oracle access to f with domain $[M]$ that makes at most T queries. Let $|\phi_i\rangle$ be the overall state after $\mathcal{A}$ makes the i-th query, and $W_{i,y}$ as the sum of squared amplitudes in $|\phi_i\rangle$ of terms in which $\mathcal{A}$ queries f on input y. Let $\epsilon > 0$ and let $F \subseteq [0, T-1] \times [M]$ be a set of time-input pairs such that $\sum_{(i,y)\in F} W_{i,y} \leq \epsilon^2/T$.*

Let f' be an oracle obtained by reprogramming f on inputs $(i, y) \in F$ to arbitrary outputs, i.e., the i-th oracle query to f on input y will be reprogrammed. Define $|\phi'_i\rangle$ as above for f'. Then,

$$T\left(|\phi_T\rangle, |\phi'_T\rangle\right) \leq \epsilon.$$

▶ Remark 3. In the above theorem, $\mathcal{A}$ is interacting with an oracle f. f is consistent, meaning that on the same inputs, it will always output the same value. In most of the cases, we will use this theorem for consistent f. However, we can also consider a case where the oracle access is different for every query. For example, $\mathcal{A}$ gets oracle access to f_i for the i-th query. The same theorem still holds as we can image an unified function f^* such that on input (i, x) it outputs $f_i(x)$. This compelling property of inconsistent oracles will be used in our proof.

▶ **Lemma 4** (Extraction Lemma). *Let $\mathcal{A}$ be an adversary with oracle access to f with domain $[M]$ that makes at most T queries. Let f' be an oracle obtained by reprogramming f on inputs $y \in \mathcal{I}$ to arbitrary outputs. Let $|\phi_T\rangle$ and $|\phi'_T\rangle$ be the states of $\mathcal{A}$ after making all the queries to f and f' respectively. If*

$$T\left(|\phi_T\rangle, |\phi'_T\rangle\right) > \mu,$$

then there exists a T query quantum algorithm $\mathcal{B}^{|f\rangle}$ that outputs $y \in \mathcal{I}$ with probability at least μ^2/T^2.

Since f and f' is symmetric, there exists a T query quantum algorithm $\mathcal{B}^{|f'\rangle}$ that outputs $y \in \mathcal{I}$ with probability at least μ^2/T^2.

Moreover, $\mathcal{B}$ is simply running $\mathcal{A}$ and measuring one random quantum query of $\mathcal{A}$.

Proof. This easily follows from Theorem 2. Let $F = \{(i, y), i \in \{0, 1, \cdots, T-1\}, y \in \mathcal{I}\}$. Since $T\left(|\phi_T\rangle, |\phi'_T\rangle\right) > \mu$, we have that $\sum_{(i,y)\in F} W_{i,y} > \mu^2/T$.

$\mathcal{B}$ runs $\mathcal{A}$ as a subroutine and measures a random quantum query to f. Therefore,

$$\Pr\left[B^{|f\rangle} \in \mathcal{I}\right] > \mu^2/T^2.$$

We conclude the proof. ◀

2.2 Wiesner States and Unclonability

For every n, every $x, \theta \in \{0,1\}^n$, the notion $|x^\theta\rangle$ denotes the quantum state

$$|x^\theta\rangle := H^{\theta_1}|x_1\rangle \otimes H^{\theta_2}|x_2\rangle \cdots \otimes H^{\theta_n}|x_n\rangle.$$

Wiesner introduced the collection of states in [47] for the purpose of private key quantum money. We call states of the form $|x^\theta\rangle$ Wiesner states. For a fixed θ, the set of states $\{|x^\theta\rangle\}_{x\in\{0,1\}^n}$ form a basis.

In the rest of the work, we will assume n is an even number and only focus on a subset of all Wiesner states, namely, all Wiesner states $|x^\theta\rangle$ with $|\theta|_1 = n/2$. For $x, \theta \in \{0,1\}^n$ with $|\theta|_1 = n/2$, we denote x_c^θ be a $n/2$-bit substring of x on indices $\mathsf{I}_{\mathsf{comp}} = \{i \in [n], \theta_i = 0\}$ and x_h^θ be a $n/2$-bit substring of x on indices $\mathsf{I}_{\mathsf{Hadamard}} = \{i \in [n], \theta_i = 1\}$.

We recall the following two useful properties of Wiesner states. The first one is unpredictability of Wiesner states.

▶ **Lemma 5** (Unpredictability of Wiesner States). *For any even $n \in \mathbb{N}^+$, any (unbounded) quantum adversary $\mathcal{A}$,*

$$\Pr_{\substack{x,\theta\leftarrow\{0,1\}^n \\ |\theta|_1=n/2}}\left[A(|x^\theta\rangle\langle x^\theta|) \in \{x_c^\theta, x_h^\theta\}\right] = 1/2^{n/2-1}.$$

Note that in the above lemma, the quantum algorithm is not given the basis information θ. Otherwise, this task becomes trivial.

Proof. Since $\mathcal{A}$ has no information about θ, the view of $\mathcal{A}$ remains unchanged if the input is replaced with a maximally mixed state $\mathbb{I}/2^n = \frac{1}{2^n}\sum_{y\in\{0,1\}^n}|y\rangle\langle y| = \frac{1}{2^n}\sum_{x\in\{0,1\}}|x^\theta\rangle\langle x^\theta|$. The probability in Lemma 5 is the same as

$$\Pr_{\substack{x,\theta\leftarrow\{0,1\}^n\\ |\theta|_1=n/2}}\left[A(\mathbb{I}/2^n)\in\{x_c^\theta,x_h^\theta\}\right],$$

which is at most $2/2^{n/2}$. ◀

The second one is unclonability. The weaker version is first proved by Broadbent and Lord in [17], based on a previous result on monogamy-of-entanglement of Wiesner states by Tomamichel et al. [43]. We will prove the following lemma based on stronger monogamy-of-entanglement of Wiesner states by Culf and Vidick [26].

▶ **Lemma 6** (Unclonability of Wiesner States). *For any even $n\in\mathbb{N}^+$, any (unbounded) quantum adversaries $(\mathcal{A},\mathcal{B},\mathcal{C})$,*

$$\Pr[\mathsf{CloneGame}_{\mathcal{A},\mathcal{B},\mathcal{C}}(n)=1]\le 0.924^n<2^{-0.11n},$$

where $\mathsf{CloneGame}_{\mathcal{A},\mathcal{B},\mathcal{C}}(n)$ is defined as:

1. *A challenger samples $x,\theta\leftarrow\{0,1\}^n$ conditioned on $|\theta|_1=n/2$ and sends $|x^\theta\rangle$ to $\mathcal{A}$.*
2. *$\mathcal{A}$ applies a quantum channel on $|x^\theta\rangle$ and obtains $\rho_{\mathbf{BC}}$ over registers $\mathbf{BC}$, which are then sent to $\mathcal{B}$ and $\mathcal{C}$ respectively.*
3. *$\mathcal{B}$ and $\mathcal{C}$ receives $\rho_{\mathbf{B}}$ and $\rho_{\mathbf{C}}$ from $\mathcal{A}$ respectively, together with θ from the challenger.*
4. *The game outputs 1 if and only if $\mathcal{B}$ outputs $x_{\mathcal{B}}=x_c^\theta$ and $\mathcal{C}$ outputs $x_{\mathcal{C}}=x_h^\theta$.*

Proof. This is a corollary of stronger monogamy-of-entanglement of Wiesner states by Culf and Vidick [26]. ◀

2.3 Time-Lock Puzzles

We recall the definition of time-lock puzzles. We will review the definition and construction by Bitansky et al. [12]. At the end of this section, we will generalize their definitions and constructions to the post-quantum setting.

Puzzles

A puzzle is parameterized by λ and t, where λ is the security parameter and t denotes the difficulty to solve a puzzle. The following two definitions are almost taken verbatim from [12]. Here we only focus on their definitions on "weakly efficient puzzles" and "weak time-lock puzzles" since it is sufficient for our work. Here "weak" says that the circuit for preparation of a puzzle is polynomial in t instead of $\log t$, but the depth is required to be polynomial in $\log t$.

▶ **Definition 7** (Puzzles). *A puzzle is a pair of classical algorithms* $(\mathsf{Puzzle.Gen},\mathsf{Puzzle.Sol})$ *satisfying the following requirements.*

- *Syntax:*
  - *$Z\leftarrow\mathsf{Puzzle.Gen}(t,s)$ is a probabilistic algorithm that takes as input a difficulty parameter t and a solution $s\in\{0,1\}^\lambda$, where λ is a security parameter, and outputs a puzzle Z.*
  - *$s\leftarrow\mathsf{Puzzle.Sol}(Z)$ is a deterministic algorithm that takes as input a puzzle Z and outputs a solution s.*

- *Completeness: For every security parameter λ, difficulty parameter t, solution $s \in \{0,1\}^\lambda$ and puzzle Z in the support of* $\mathsf{Puzzle.Gen}(t, s)$, $\mathsf{Puzzle.Sol}(Z)$ *outputs s with probability* 1.
- *Weak Efficiency:*
  - $\mathsf{Puzzle.Gen}(t, s)$ *can be computed by a uniform circuit of* size $\mathsf{poly}(t, \lambda)$ *and* depth $\mathsf{poly}(\log t, \lambda)$.
  - $\mathsf{Puzzle.Sol}(Z)$ *can be computed in time* $t \cdot \mathsf{poly}(\lambda)$.

The following "time-lock" property guarantees that no circuit of small depth can extract any meaningful information from the puzzle.

▶ **Definition 8** (Weak Time-Lock Puzzles)**.** *A puzzle* $(\mathsf{Puzzle.Gen}, \mathsf{Puzzle.Sol})$ *is a weak time-lock puzzle with gap $\varepsilon < 1$ if there exists a polynomial $\underline{t}(\cdot)$ such that for every polynomial $t(\cdot) \geq \underline{t}(\cdot)$ and every polysize* classical *circuit $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}^+}$ of depth $\mathsf{dep}(\mathcal{A}) \leq t^\epsilon(\lambda)$, there exists a negligible function μ, such that for every $\lambda \in \mathbb{N}^+$ and every pair of solutions $s_0, s_1 \in \{0,1\}^\lambda$:*

$$\Pr\left[b \leftarrow \mathcal{A}_\lambda(s_0, s_1, Z) : \begin{array}{l} b \leftarrow \{0,1\}, \\ Z \leftarrow \mathsf{Puzzle.Gen}(t(\lambda), s_b) \end{array}\right] \leq \frac{1}{2} + \mu(\lambda).$$

The following shows that weak time-lock puzzles exist assuming a non-parallelizing language and one-way functions (Theorem 3.10 in [12]).

▶ **Theorem 9** (Weak Time-Lock Puzzles, [12])**.** *Let $\epsilon < 1$. Assume that, for every polynomially bounded function $t(\cdot)$, there exists a non-parallelizing language $\mathcal{L} \in \mathsf{DTime}(t(\cdot))$ with gap ϵ. Assume one-way functions exist. Then for any $\underline{\epsilon} < \epsilon$, there exists a weak time-lock puzzle with gap $\underline{\epsilon}$.*

In this work, we naturally ask a puzzle to have post-quantum time-lock property, defined as follows.

▶ **Definition 10** (Post-Quantum Weak Time-Lock Puzzles)**.** *The definition is the same as Definition 8, except $\mathcal{A} = \{\mathcal{A}_\lambda\}_{\lambda \in \mathbb{N}^+}$ is modeled as a collection of polysize* quantum *circuits of depth $\mathsf{dep}(\mathcal{A}) \leq t^\epsilon(\lambda)$.*

Similar to Theorem 9, we can show the existence of post-quantum weak time-lock puzzles assuming a quantum non-parallelizing language and post-quantum one-way functions.

▶ **Definition 11** (Quantum Non-Parallelizing Language)**.** *A language $\mathcal{L} \in \mathsf{DTime}(t(\cdot))$ is quantum non-parallelizing with gap $\epsilon < 1$ if every family of non-uniform polysize* quantum *circuits $\mathcal{B} = \{\mathcal{B}_\lambda\}_{\lambda \in \mathbb{N}^+}$ where $\mathsf{dep}(\mathcal{B}_\lambda) \leq t^\epsilon(\lambda)$ and every large enough λ, $\mathcal{B}_\lambda$ fails to decide $\mathcal{L} \cup \{0,1\}^\lambda$.*

The existence of quantum non-parallelizing with any gap $\epsilon < 1$ is shown to exist relative to a random oracle by Chung, Fehr, Huang, Liao ([22], Theorem 5.25) and Blocki, Lee, Zhou ([13], Theorem 3). The problem is iterated evaluation of H t times, on which we will not elaborate.

▶ **Theorem 12** (Quantum Non-Parallelizing Language in the QROM, [22, 13])**.** *There exists quantum non-parallelizing languages with any gap $\epsilon < 1$ relative to a quantum random oracle.*

▶ **Theorem 13** (Post-Quantum Weak Time-Lock Puzzles)**.** *Let $\epsilon < 1$. Assume that, for every polynomially bounded function $t(\cdot)$, there exists a quantum non-parallelizing language $\mathcal{L} \in \mathsf{DTime}(t(\cdot))$ with gap ϵ. Assume post-quantum one-way functions exist. Then for any $\underline{\epsilon} < \epsilon$, there exists a post-quantum weak time-lock puzzle with gap $\underline{\epsilon}$.*

Proof. The proof is identical to that of Theorem 3.10 in [12], except the underlying one-way function is post-quantum and the language $\mathcal{L}$ is quantum non-parallelizing. ◀

3 Bounded Depth Quantum Adversaries

In this section, we formally introduce the notion of bounded depth quantum adversaries. We model a quantum adversary as a sequence of bounded depth quantum circuits, each followed by a complete measurement on computational basis, interleaved with any polynomial-time classical algorithms. We start by defining quantum circuit families with depth d, similarly defined by [37, 42, 20]. The following definition is adapted from the definition of QNC_d in [20].

▶ **Definition 14** (d-Depth Quantum Circuit). *A quantum circuit $\mathcal{C}$ is of depth d if $\mathcal{C}$ consists of d layers of one- and two-qubit gates.*

Let f be a classical function. $\mathcal{C}^{|f\rangle}$ is of depth d if $\mathcal{C}$ consists of d layers of one-, two-qubit gates and oracle gates $\mathcal{O}_f$.

We can now formally define a d-depth bounded quantum adversary. It takes as input a quantum state and outputs a quantum state.

▶ **Definition 15** (d-Depth Quantum Algorithm/Adversary). *Let d be a function $\mathbb{N}^+ \to \mathbb{N}^+$. A d-depth bounded quantum algorithm/adversary on input λ qubits is parameterized by polynomials $\ell(\cdot), p(\cdot)$ and a collection of non-uniform quantum circuits of the form*

$$\left\{\mathcal{C}_{\lambda,\ell(\lambda)+1} \circ \mathcal{A}_{\lambda,\ell(\lambda)} \circ \Pi \circ \mathcal{C}_{\lambda,\ell(\lambda)} \circ \cdots \circ \mathcal{A}_{\lambda,1} \circ \Pi \circ \mathcal{C}_{\lambda,1}, \sigma_\lambda\right\}_{\lambda\in\mathbb{N}^+},$$

where

- *Each $\mathcal{C}_{\lambda,j}$ is a $d(\lambda)$-depth polysize quantum circuit that operates on $\lambda + p(\lambda)$ qubits.*
- *Each $\mathcal{A}_{\lambda,j}$ is a polynomial-time randomized classical algorithm that takes input $\lambda + p(\lambda)$ bits and outputs $\lambda + p(\lambda)$ bits.*
- *$\mathcal{C}_{\lambda,1}$ operates on λ input qubits and $p(\lambda)$ ancilla qubits σ_λ.*
- *Π is a standard-basis measurement on $\lambda + p(\lambda)$ qubits, follows every $\mathcal{C}_{\lambda,j}$ except the last one.*

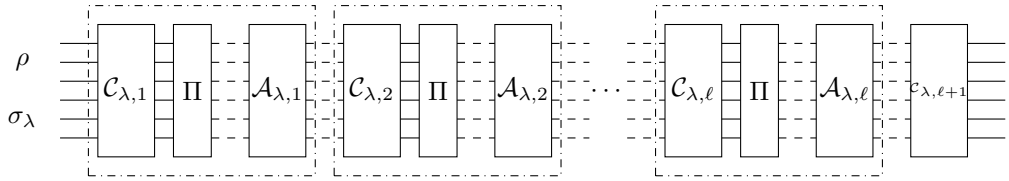

Figure 3 d-depth bounded quantum algorithm. Dashed lines denote classical bits and solid lines denote qubits. Each $\mathcal{C}_{\lambda,j}$ is a d-depth quantum circuit and $\mathcal{A}_{\lambda,j}$ is a polynomial-time randomized classical algorithm.

Note that since it takes quantum inputs, it is crucially to have the first circuit being quantum circuit. We can similarly define a d-depth quantum algorithm with oracle access to f: each $\mathcal{C}_{\lambda,j}$ and $\mathcal{A}_{\lambda,j}$ has oracle access to f.

4 One-Time Memory against Bounded Depth Quantum Adversaries

4.1 Definitions

A one-time memory scheme consists of two procedures: one for generating a quantum token and the other one for retrieving information from the token. It is parameterized by a security parameter λ and a message length n, described as follows:

▶ **Definition 16** (Memory Token). *Let f be a random oracle with domain $[2^\lambda]$ and sufficiently large range (at least $\max\{2^{2\lambda}, 2^n\}$). A memory token scheme in the QROM is a pair of quantum algorithms $\mathsf{Token.Gen}^f$ and $\mathsf{Token.Eval}^f$ satisfying the following requirements:*

- *Syntax:*
  - *Both quantum algorithms only classically query f.*
  - *$|\mathsf{tk}\rangle \leftarrow \mathsf{Token.Gen}^f(1^\lambda, m_0, m_1)$ is a quantum algorithm that takes as input a security parameter and two messages $m_0, m_1 \in \{0,1\}^n$, and outputs a quantum state $|\mathsf{tk}\rangle$.*
  - *$m \leftarrow \mathsf{TokenEval}^f(|\mathsf{tk}\rangle, b)$ is a quantum algorithm that takes as input a quantum state and a bit $b \in \{0,1\}$, and outputs a message m.*
- *Completeness: For every security parameter parameter λ, every message length n, every pair of messages $m_0, m_1 \in \{0,1\}^n$ and every bit $b \in \{0,1\}$,*

$$\Pr_f\left[\mathsf{Token.Eval}^f(|\mathsf{tk}\rangle, b) = m_b, |\mathsf{tk}\rangle \leftarrow \mathsf{Token.Gen}^f(1^\lambda, m_0, m_1)\right] = 1,$$

  where f is a uniformly random function.
- *Efficiency: Both $\mathsf{Token.Gen}$ and $\mathsf{Token.Eval}$ can be computed by a d-depth quantum algorithm for $d = O(1)$.*

Next, we formally define "one-timeness". It says that, for any small depth quantum adversary, it can either learn m_0 or m_1 but not both.

▶ **Definition 17** (One-Time Memory agaisnt Depth Bounded Adversary). *A memory token scheme ($\mathsf{Token.Gen}, \mathsf{Token.Eval}$) in the QROM is a one-time memory against d-depth quantum adversaries satisfies the followings. For every message length n, every d-depth quantum adversary $\mathcal{A}$, every inverse polynomial $\gamma(\cdot)$, there exists an efficient quantum simulator Sim, for every pair of messages $m_0, m_1 \in \{0,1\}^n$, Sim makes at most one classical query to $g^{m_0,m_1} : b \to m_b$ such that the following two distributions have statistical distance at most $\gamma(\lambda)$:*

$$\mathcal{A}^{|f\rangle}\left(\mathsf{Token.Gen}^f(1^\lambda, m_0, m_1)\right) \approx_{\gamma(\lambda)} \mathsf{Sim}^{g^{m_0,m_1}}(1^\lambda),$$

here f is a uniformly random function.

In the above definition, Sim is not necessary to be d-depth bounded. We believe this is a natural definition, as it captures the limitation of depth bounded adversary, by the behavior of some efficient quantum adversary interacting classically once with the oracle g^{m_0,m_1}, without any depth constraint.

▶ Remark 18. Simulation-based security is very subtle, because of the presence of programmable random oracles [46, 36, ...]. A stronger version than Definition 17 can be formulated either (1) by replacing the indistinguishability by that when a distinguisher additionally gets oracle access to a random oracle or a simulated oracle provided by a simulator or (2) under the quantum UC framework [18, 44]. These definitions will help establish sequential or general composability of our protocol. We leave stronger definitions as an interesting open direction. For the applications considered in this work, composability is straightforward.

We show the parallel composability of our protocol, which is useful in constructing one-time programs (following the ideas in [31], with Yao's garbled circuits [48]).

▶ **Definition 19** (Parallel Composability). *A memory token scheme ($\mathsf{Token.Gen}, \mathsf{Token.Eval}$) in the QROM with parallel composability against d-depth quantum adversaries satisfies the followings. For every message length n, every polynomial I, every d-depth quantum adversary*

$\mathcal{A}$, every inverse polynomial $\gamma(\cdot)$, there exists an efficient quantum simulator Sim*, for every $I := I(\lambda)$ pairs of messages $m_0^{(i)}, m_1^{(i)} \in \{0,1\}^n$ $(1 \le i \le I)$,* Sim *makes at most one classical query to each of $g^{m_0^{(i)}, m_1^{(i)}} : b \to m_b^{(i)}$ for each $1 \le i \le I$ such that the following two distributions have statistical distance at most $\gamma(\lambda)$:*

$$\mathcal{A}^{|f\rangle}\left(\bigotimes_{i=1}^{I} \mathsf{Token.Gen}^f(1^\lambda, m_0^{(i)}, m_1^{(i)})\right) \approx_{\gamma(\lambda)} \mathsf{Sim}^{g^{m_0^{(1)}, m_1^{(1)}}, \cdots, g^{m_0^{(I)}, m_1^{(I)}}}(1^\lambda),$$

here f is a uniformly random function.

4.2 Construction

In this section, we give our construction of one-time memory in the QROM against d-depth bounded adversary. It is based on a post-quantum weak time-lock puzzle with gap ϵ and Wiesner states.

$\mathsf{Token.Gen}^f(1^\lambda, m_0, m_1)$:

- Sample $r \leftarrow \{0,1\}^\lambda$. Sample a uniformly random $\theta \leftarrow \{0,1\}^{2\lambda}$ with hamming weight $|\theta|_1 = \lambda$ using the random coins $f(r)$ (we denote this sampling procedure by $\theta \leftarrow f(r)$).
- Sample $x \leftarrow \{0,1\}^{2\lambda}$. Let $y_0 \leftarrow f(x_c^\theta)$ and $y_1 \leftarrow f(x_h^\theta)$ be two strings with length n.
- Let $Z \leftarrow \mathsf{Puzzle.Gen}((10d \log \lambda)^{1/\epsilon}, r)$ be a post-quantum time-lock puzzle with solution r.
- Output $|\mathsf{tk}\rangle = (|x^\theta\rangle, m_0 \oplus y_0, m_1 \oplus y_1, Z)$.

$\mathsf{Token.Eval}^f(|\mathsf{tk}\rangle, b)$:

- Parse and measure $|\mathsf{tk}\rangle$ to obtain quantum ρ and classical c_0, c_1, Z'.
- Measure ρ on standard basis if $b = 0$; otherwise, measure it on Hadamard basis. Let the result be x'.
- Let $r' \leftarrow \mathsf{Puzzle.Solve}(Z')$. Compute θ' using the random coins $f(r')$.
- If $b = 0$: compute x'_c from x' and θ'; let $y'_0 \leftarrow f(x'_c)$ and output $c_0 \oplus y'_0$.
- If $b = 1$: compute x'_h from x' and θ'; let $y'_1 \leftarrow f(x'_h)$ and output $c_1 \oplus y'_1$.

Figure 4 One-Time Memory Construction.

Notations

In the above construction (Figure 4), we will appropriately truncate the output of f: (1) when sampling θ with hamming weight λ, $\mathsf{Token.Gen}$ only uses the first $\log_2 \binom{2\lambda}{\lambda}$ bits; (2) when sampling y_0 and y_1, it uses the first n bits. For convenience, the notations $\theta \leftarrow f(x)$ and $y_b \leftarrow f(x)$ denote truncating the output of f to the appropriate length. In the proof, the notations $f(x) = \theta$ and $f(x) = y_b$ denote reprogramming f on input x with output θ (or y_b) and padding the rest of the output with random coins.

▶ **Proposition 20.** *The above construction satisfies completeness and efficiency.*

Proof. Completeness follows from the symmetry of the construction. Since the only quantum operation is to prepare $|x^\theta\rangle$ and measure under standard/Hadamard basis, both $\mathsf{Token.Gen}$ and $\mathsf{Token.Eval}$ can be computed by a $O(1)$-depth quantum algorithm. ◀

4.3 Security Proof

▶ **Theorem 21.** *The above construction is a one-time memory against d-depth bounded adversaries.*

We will refer readers to the full version for the proof.

References

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

2 Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 41–60, 2012.

3 Scott Aaronson, Jiahui Liu, Qipeng Liu, Mark Zhandry, and Ruizhe Zhang. New approaches for quantum copy-protection. In *Annual International Cryptology Conference*, pages 526–555. Springer, 2021.

4 Andris Ambainis, Ashwin Nayak, Amnon Ta-Shma, and Umesh Vazirani. Dense quantum coding and quantum finite automata. *Journal of the ACM (JACM)*, 49(4):496–511, 2002.

5 Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

6 Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im) possibility of obfuscating programs. In *Annual international cryptology conference*, pages 1–18. Springer, 2001.

7 James Bartusek and Dakshita Khurana. Cryptography with certified deletion. *arXiv preprint*, 2022. arXiv:2207.01754.

8 Amit Behera, Or Sattath, and Uriel Shinar. Noise-tolerant quantum tokens for mac. *arXiv preprint*, 2021. arXiv:2105.05016.

9 Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.

10 Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM journal on Computing*, 26(5):1510–1523, 1997.

11 Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *arXiv preprint*, 2020. arXiv:2003.06557.

12 Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. Time-lock puzzles from randomized encodings. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 345–356, 2016.

13 Jeremiah Blocki, Seunghoon Lee, and Samson Zhou. On the security of proofs of sequential work in a post-quantum world. *arXiv preprint*, 2020. arXiv:2006.10972.

14 Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry. Random oracles in a quantum world. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 41–69. Springer, 2011. doi:10.1007/978-3-642-25385-0_3.

15 Anne Broadbent, Sevag Gharibian, and Hong-Sheng Zhou. Towards quantum one-time memories from stateless hardware. *Quantum*, 5:429, 2021.

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

17 Anne Broadbent and Sébastien Lord. Uncloneable quantum encryption via oracles. *arXiv preprint*, 2019. arXiv:1903.00130.

18 Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

19 Michael Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 659–668, 2002.

20 Nai-Hui Chia, Kai-Min Chung, and Ching-Yi Lai. On the need for large quantum depth. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 902–915, 2020.

21 Nai-Hui Chia and Shih-Han Hung. Classical verification of quantum depth. *arXiv preprint*, 2022. `arXiv:2205.04656`.

22 Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 598–629. Springer, 2021.

23 Kai-Min Chung, Marios Georgiou, Ching-Yi Lai, and Vassilis Zikas. Cryptography with disposable backdoors. *Cryptography*, 3(3):22, 2019.

24 Andrea Coladangelo, Jiahui Liu, Qipeng Liu, and Mark Zhandry. Hidden cosets and applications to unclonable cryptography. In *Annual International Cryptology Conference*, pages 556–584. Springer, 2021.

25 Matthew Coudron and Sanketh Menda. Computations with greater quantum depth are strictly more powerful (relative to an oracle). In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 889–901, 2020.

26 Eric Culf and Thomas Vidick. A monogamy-of-entanglement game for subspace coset states. *arXiv preprint*, 2021. `arXiv:2107.13324`.

27 Akshay Degwekar, Vinod Vaikuntanathan, and Prashant Nalini Vasudevan. Fine-grained cryptography. In *Annual International Cryptology Conference*, pages 533–562. Springer, 2016.

28 Shohei Egashira, Yuyu Wang, and Keisuke Tanaka. Fine-grained cryptography revisited. *Journal of Cryptology*, 34(3):1–43, 2021.

29 Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy N Rothblum. Verifying and decoding in constant depth. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 440–449, 2007.

30 Shafi Goldwasser, Dan Gutfreund, Alexander Healy, Tali Kaufman, and Guy N Rothblum. A (de) constructive approach to program checking. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 143–152, 2008.

31 Shafi Goldwasser, Yael Tauman Kalai, and Guy N Rothblum. One-time programs. In *Annual International Cryptology Conference*, pages 39–56. Springer, 2008.

32 Johan Hastad. One-way permutations in nc0. *Information Processing Letters*, 26(3):153–155, 1987.

33 IBM. Ibm announces advances to ibm quantum systems & ecosystem, 2017. URL: `https://newsroom.ibm.com/2017-11-10-IBM-Announces-Advances-to-IBM-Quantum-Systems-Ecosystem`.

34 Jørn Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972.

35 Gil Kalai. The argument against quantum computers. In *Quantum, Probability, Logic*, pages 399–422. Springer, 2020.

36 Yehuda Lindell. How to simulate it–a tutorial on the simulation proof technique. *Tutorials on the Foundations of Cryptography*, pages 277–346, 2017.

37 Cristopher Moore and Martin Nilsson. Parallel quantum computation and quantum codes. *SIAM journal on computing*, 31(3):799–815, 2001.

38 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. `doi:10.1017/CBO9780511976667`.

39 John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

40 R. L. Rivest, A. Shamir, and D. A. Wagner. Time-lock puzzles and timed-release crypto. Technical report, Massachusetts Institute of Technology, USA, 1996.

41 Michael Sipser and Daniel A Spielman. Expander codes. *IEEE transactions on Information Theory*, 42(6):1710–1722, 1996.

42 Barbara M Terhal and David P DiVincenzo. Adaptive quantum computation, constant depth quantum circuits and arthur-merlin games. *arXiv preprint*, 2002. arXiv:quant-ph/0205133.

43 Marco Tomamichel, Serge Fehr, Jędrzej Kaniewski, and Stephanie Wehner. A monogamy-of-entanglement game with applications to device-independent quantum cryptography. *New Journal of Physics*, 15(10):103002, 2013. doi:10.1088/1367-2630/15/10/103002.

44 Dominique Unruh. Universally composable quantum multi-party computation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–505. Springer, 2010.

45 Dominique Unruh. Revocable quantum timed-release encryption. *J. ACM*, 62(6), December 2015. doi:10.1145/2817206.

46 Hoeteck Wee. Zero knowledge in the random oracle model, revisited. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 417–434. Springer, 2009.

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

48 Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167. IEEE, 1986.

49 Mark Zhandry. Quantum lightning never strikes the same state twice. or: quantum money from cryptographic assumptions. *Journal of Cryptology*, 34(1):1–56, 2021.