Secure Software Leasing and Implications for Quantum Copy-Protection and Obfuscation

Gorjan Alagic Prabhanjan Ananth Zvika Brakerski Yfke Dulek
Rolando L. La Placa Christian Schaffner

Joint submission of arXiv:2005.05289 and arXiv:2005.06432.

Most proprietary software comes with a license, a legal document that governs against illegal distribution, or "piracy", of the software. Often the license forbids the user from distributing the software, i.e. creating additional copies.

While ad hoc heuristics for preventing unlicensed distribution exist, the first[1] theoretical approach appeared only in 2009 in Aaronson's seminal work [2]. That work introduced and formalized the notion of quantum software copy-protection, a quantum cryptographic primitive that uses quantum no-cloning techniques to prevent piracy. Roughly, quantum copy-protection says that given a quantum state computing a function f, the adversary cannot produce two (possibly entangled) quantum states that each individually compute f.[2] This protection prevents a pirate from creating a new software from his own copy and re-distributing it; of course he can circulate his own copy, but he himself will lose access to it.

Constructing this primitive, however, has been notoriously difficult. Since its introduction over a decade ago, there are no provably secure constructions of quantum copy-protection for *any* class of circuits. Existing constructions are either proven in an oracle model [2, 3] or are heuristic candidates for simple classes, such as point functions [2]. In a recent blog post, Aaronson [1] even mentioned constructing quantum copy-protection from cryptographic assumptions as one of the five big questions he wishes to solve.

This state of affairs not only prompted us to explore the possibility of copy-protection, but also to consider relaxed notions to protect against unlicensed distribution that may still be useful.

New Notion: Secure Software Leasing. We introduce a relaxed variant of copy-protection that only prevents an adversary from being able to create *authenticated* pirated copies. We call this variant secure software leasing (SSL).

The use case for SSL that we consider is as follows. An authority (the lessor) can lease a classical circuit C to a user (the lessee) by providing a corresponding quantum state ρ_C. The lessee can execute ρ_C to compute C on any input. Once the lease expires, the lessee returns ρ_C to the lessor. From that point on, the lessee should no longer be able to compute C. So far this functionality is the same as copy-protection, however in SSL we only guarantee that the lessee is unable to compute C using a "prescribed" evaluation algorithm (which thus also serves as a means for authentication).

More formally, a secure software leasing (SSL) scheme for a family of circuits $\mathcal{C}$ is a collection $(\mathsf{Gen}, \mathsf{Lessor}, \mathsf{Run}, \mathsf{Check})$ of quantum polynomial-time (QPT) algorithms satisfying the following conditions. $\mathsf{Gen}(1^\lambda)$, on input a security parameter λ, outputs a secret key sk. For any circuit

[1] A related notion, called software watermarking [6], provides a solution for software anti-piracy in a restricted setting (where the user is allowed to distribute copies but is not allowed to modify the ownership of the software). However, our primary goal is to study the general problem where no such restriction on the license is placed.

[2] More generally, Aaronson also considers a setting where the adversary gets multiple copies and is required to create one additional copy.

$C : \{0,1\}^n \to \{0,1\}^m$ in $\mathcal{C}$, $\mathsf{Lessor}(\mathsf{sk}, C)$ outputs a quantum state ρ_C, where ρ_C allows Run to evaluate C. Specifically, for any $x \in \{0,1\}^n$, we want that $\mathsf{Run}(\rho_C, x) = C(x)$; this algorithm is executed by the lessee. Finally, $\mathsf{Check}(\mathsf{sk}, \rho_C)$ validates the returned state ρ_C after the expiration of the lease. Any state produced by the lessor is a valid state and will pass the verification check.

An SSL scheme can have two different security guarantees depending on whether the leased state is supposed to be returned or not.

- *Infinite-Term Lessor Security*: In this setting, the user is allowed to keep the leased state forever[3]. Informally, we require that the lessee cannot produce two *authenticated*[4] states both of which compute C. Formally speaking, any (malicious) QPT user $\mathcal{A}$ cannot output a (possibly entangled) bipartite state σ^* such that both $\sigma_1^* = \mathrm{tr}_2[\sigma^*]$ and $\sigma_2^* = \mathrm{tr}_1[\sigma^*]$ can be used to compute C with Run.
- *Finite-Term Lessor Security*: In this weaker setting, the leased state is associated with a fixed term, after which the lessee is obligated to return it. We require that the lessee cannot simultaneously return the original state and produce another *authenticated* state with the same functionality. Formally speaking, we require that any (malicious) QPT user $\mathcal{A}$ cannot output a (possibly entangled) bipartite state σ^* such that $\sigma_1^* := \mathrm{tr}_2[\sigma^*]$ passes the lessor's verification ($\mathsf{Check}(\mathsf{sk}, \sigma_1^*) = 1$) and such that the remaining state $\sigma_2^* := \mathrm{tr}_1[\sigma^*]$ can also be used to evaluate C with $\mathsf{Run}(\sigma_2^*, x) = C(x)$.

Subsequent Work. Our notion has already been adopted by two subsequent papers: the first paper constructs SSL unconditionally in the quantum random oracle model [9] and the second paper constructs finite-term SSL for a different class of functions from weaker assumptions [10]. Another related paper [3] shows how to construct a weaker notion of copy-protection, called copy-detection, for a different class of functions. They also construct copy-protection relative to classical oracles.

Construction of Infinite-Term SSL. We present a construction of SSL for a restricted class of unlearnable circuits: in particular, a subclass $\mathcal{C}$ of evasive circuits (which are circuits for which finding an accepting input is computationally hard). Ours is the first provably secure construction for the problem of software anti-piracy in the standard model (i.e., without oracles). We prove the following:

Theorem 1 (Infinite-Term SSL; Informal)**.** *Assuming post-quantum indistinguishability obfuscators exist and learning with errors is hard against sub-exponential quantum polynomial-time adversaries, there exists an infinite-term secure SSL scheme for $\mathcal{C}$.*

Our main insight is to leverage the uncloneability of publicly-verifiable quantum states[5], implicit in the work of [13], along with existing cryptographic primitives.

We design the scheme such that the leased state ρ_C, produced using a classical circuit C, has the following properties: (1) any QPT adversary producing a state ρ^* such that $\mathrm{Tr}_1[\rho^*] = \rho_C$ and $\mathrm{Tr}_2[\rho^*] = \rho_C$ would violate the uncloneability of publicly verifiable quantum states and, (2) any QPT adversary producing ρ^* such that $\mathrm{Tr}_1[\rho^*] = \rho_C$ and $\mathrm{Tr}_2[\rho^*] = \rho'_C$, with $\rho'_C \neq \rho_C$ and computing C, would violate the security of cryptographic primitives. Specifically, we use two primitives: the first primitive ensures that if the adversary produces a different state ρ'_C then it should have known an accepting input for C, and the second primitive ensures that given just ρ_C, it is computationally infeasible to produce an accepting input for C. Both these primitives together guarantee that the adversary could not have produced a different state ρ'_C.

[3] Although the lessor will technically be the owner of the leased state.

[4] We call any state ρ to be an authenticated state (of C) if for all $x \in \{0,1\}^n$, $\mathsf{Run}(\rho_C, x) = C(x)$.

[5] These are states that can be verified, against a verification key, by anyone.

Impossibility of SSL and Quantum Copy-Protection. We show that, under cryptographic assumptions, there exists a class of unlearnable circuits such that no SSL exists for this class. This also rules out the existence of quantum copy-protection for arbitrary unlearnable functions, thus resolving an important open problem in quantum cryptography [2, 1].

We identify a class of classical circuits $\mathcal{C}$ that we call *de-quantumizable*: given *any* efficient quantum implementation of $C \in \mathcal{C}$, we can efficiently "de-quantumize" it to obtain a classical circuit $C' \in \mathcal{C}$ that has the same functionality as C. Note that any learnable class is also de-quantumizable. Quantum copy-protection and secure software leasing are only meaningful for unlearnable functions, so we add the non-trivial requirement that de-quantumizable circuit classes should be unlearnable. No de-quantumizable class can be securely software leased.

We show how the outline of Barak et al.'s impossibility of (classical) virtual-black-box obfuscation [6] can be combined with non-black-box techniques introduced in seemingly different contexts – constructing quantum zero-knowledge protocols [7, 5] – to prove the following:

Theorem 2 (Informal)**.** *Assuming the quantum hardness of learning with errors (QLWE), and asssuming the existence of quantum fully homomorphic encryption [11, 8] (QFHE), there exists a de-quantumizable class of circuits.*

The circuits in this class are designed such that non-black-box access (e.g., in the form of a leased state ρ_C) allows a user to recover a secret, which is then sufficient to recover the functionality of C. Black-box access alone does not reveal this secret, because in order to access it, one must be able to homomorphically run the functionality of C. Such a homomorphic evaluation is possible with an efficient implementation of C (such as ρ_C), but not with black-box access.

A circuit in the class can be used to retrieve the public evaluation key to a QFHE scheme, which in turn is used to homomorphically evaluate the circuit itself. This dependence is inconvenient, since the size of the evaluation key may scale with the size of the evaluated circuit, and the circuit scales with the length of the evaluation key. To get rid of this dependence, the circuits in our class return the evaluation key in small, individual blocks that can be independently computed. We argue that any classical-key quantum fully-homomorphic encryption scheme has public keys that can be "decomposed" in this way using garbled circuits [12]. In particular, existing leveled QFHE schemes [11, 8] can be employed without an additional assumption on their circular security.

Impossibility of Quantum Virtual-Black-Box Obfuscation. The existence of de-quantumizable circuits not only rules out copy-protection for arbitrary circuits, but also quantum virtual black-box (VBB) obfuscation of classical circuits, solving an open question from Alagic and Fefferman [4]. In fact, in order to show this (weaker) impossibility, the circuit class does not have to be fully de-quantumizable: instead of being able to recover a secret (leading to a circuit description of C), it suffices for an adversary to distinguish between two different circuits in the class. A direct proof for the impossibility of quantum VBB is similar to the impossibility of SSL, but follows the techniques from Barak et al.'s proof [6] for impossibility of classical obfuscation more closely.

References

[1] Scott Aaronson. Shtetl-Optimized. Ask Me Anything: Apocalypse Edition. https://www.scottaaronson.com/blog/?p=4684#comment-1834174. Comment #283, Posted: 03-24-2020, Accessed: 03-25-2020.

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

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

[4] Gorjan Alagic and Bill Fefferman. On quantum obfuscation. *arXiv preprint arXiv:1602.01771*, 2016.

[5] Prabhanjan Ananth and Rolando L. La Placa. Secure quantum extraction protocols. In *TCC*, 2020.

[6] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2001.

[7] Nir Bitansky and Omri Shmueli. Post-quantum zero knowledge in constant rounds. In *STOC*, 2020.

[8] Zvika Brakerski. Quantum fhe (almost) as secure as classical. In *Annual International Cryptology Conference*, pages 67–95. Springer, 2018.

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

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

[11] Urmila Mahadev. Classical homomorphic encryption for quantum circuits. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 332–338. IEEE, 2018.

[12] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS*, pages 162–167, 1986.

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