Secure Software Leasing Without Assumptions

Anne Broadbent[1] Stacey Jeffery[2] Sébastien Lord[1] Supartha Podder[1] Aarthi Sundaram[3]

[1]University of Ottawa [2]QuSoft and CWI [3]Microsoft — QCrypt 2021 — arXiv:2101.12739

Main Result

From any total authentication scheme, we construct a secure software leasing scheme for point functions which is provably secure against unbounded adversaries in the plain model.

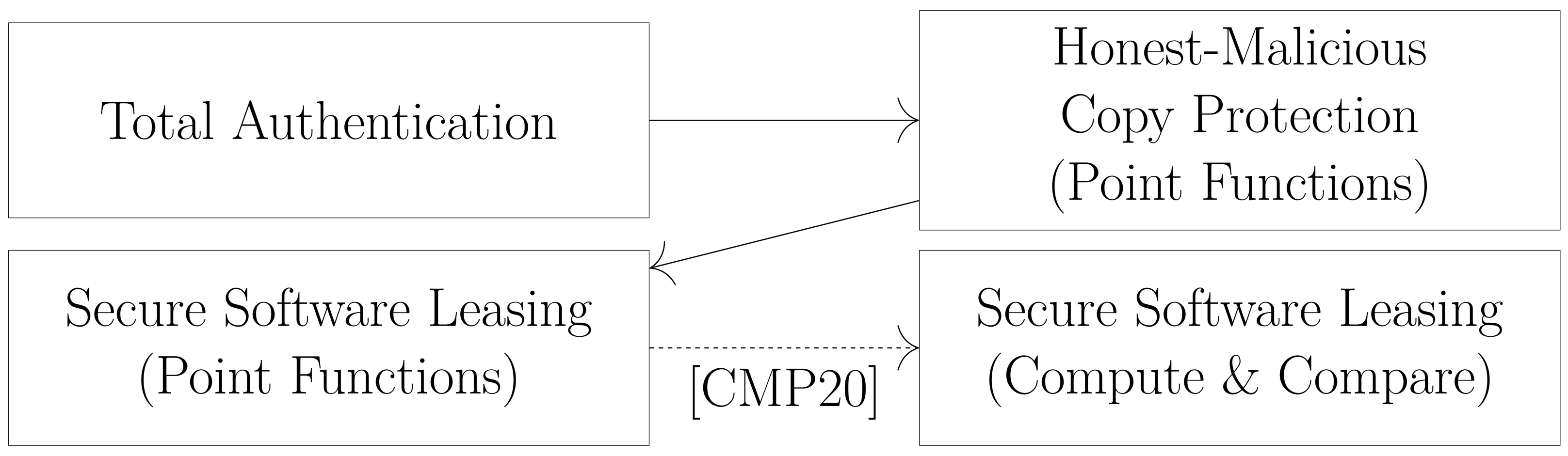

Figure 1: Notions considered in this work. Prior work yields the last implication.

1. Introduction: A Hierarchy of Uncloneability

The no-cloning theorem tells us that it is impossible to create perfect copies of arbitrary quantum states. One can use this fact to construct a hierarchy of uncloneable cryptographic primitives. At the base of this hierarchy, we have primitives which achieve a notion of **authenticity** that is uncloneable. This includes quantum money schemes. One level up, we have schemes to make **information** uncloneable, such as uncloneable encryption and tamper-evident encryption.

There has been recent interest in achieving uncloneable **functionalities**, the next level in this hierarchy. These are instantiated either as copy-protection schemes [Aar09] or as secure software leasing schemes [ALP21].

2. What is Secure Software Leasing (SSL)?

An SSL scheme is a set of procedures to **encode a circuit** C, taken from a suitable family $\mathcal{C}$, **as a quantum state** ρ_C which can then be used to evaluate C on an input x. Formally, we have

$$\rho_C = \mathsf{Lease}(C) \quad \text{and} \quad C(x) = \mathsf{Eval}(\rho_C, x). \tag{1}$$

An SSL scheme also includes a procedure which allows the original creator of a program state to **verify the return of a program sate**:

$$\text{Accept/Reject} \leftarrow \mathsf{Verify}(\sigma, C). \tag{2}$$

An SSL scheme should also satisfy a security guarantee which essentially states that a user can no longer evaluate C once they have returned ρ_C to the vendor.

3. The Secure Software Leasing Security Game

1. The referee samples C and gives $\rho_C = \mathsf{Lease}(C)$ to the adversary.
2. The adversary returns a state σ to the referee.
3. The referee runs $\mathsf{Verify}(\sigma, C)$. The adversary loses if the state is rejected.
4. The referee samples x, depending on C, and gives it to the adversary.
5. The adversary outputs a y and wins if and only if $C(x) = y$.

Definition. A scheme is ϵ secure if no adversary wins this game with probability greater than $p_{\text{trivial}} + \epsilon$ for a trivially possible probability p_{trivial}.

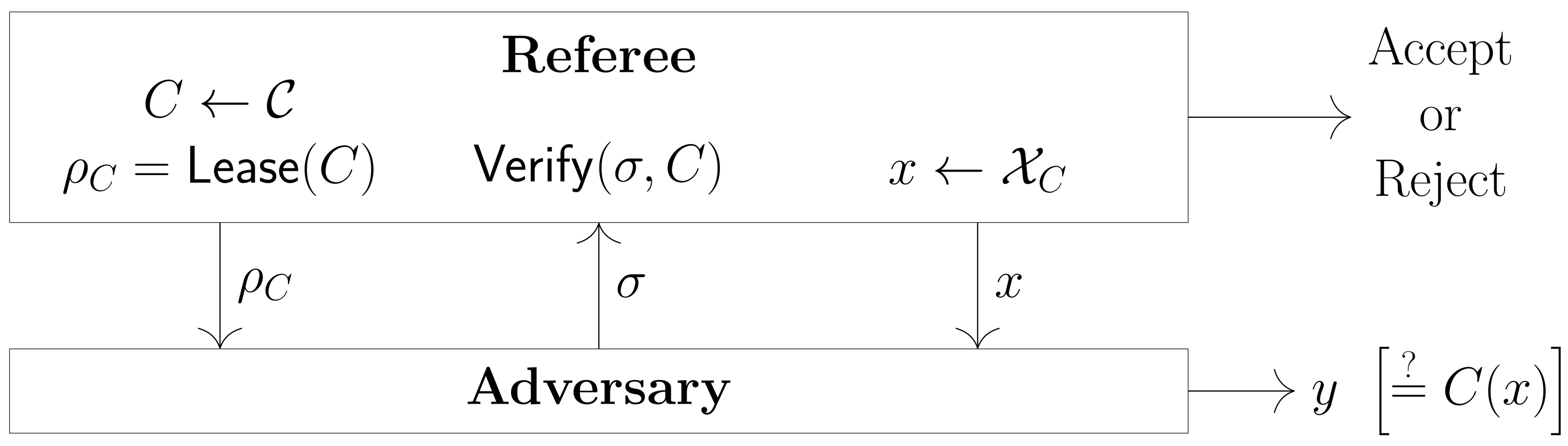

Figure 2: The secure software leasing security game.

4. Point Functions and Total Authentication

For any $p \in \{0,1\}^n$, the point function $f_p : \{0,1\}^n \to \{0,1\}$ is defined by

$$f_p(q) = 1 \iff p = q. \tag{3}$$

A total authentication scheme [GYZ17] $\{\mathsf{Auth}_k, \mathsf{Verf}_k\}_k$ is a pair of keyed procedures such that

$$\mathsf{Verf}_k \circ \mathsf{Auth}_k(\rho) = \rho \otimes |\text{Accept}\rangle\langle\text{Accept}|\,. \tag{4}$$

Note that the Verf_k procedure could also produce a $|\text{Reject}\rangle$ state.

The security guarantee is that, conditioned on acceptance, any eavesdropper between Auth_k and Verf_k essentially did not interact with the encoded state.

Our Construction

Let $\{\mathsf{Auth}_k, \mathsf{Verf}_k\}_k$ be a total authentication scheme and $|\psi\rangle$ a fixed state.

Lease

Input: A circuit C_p for the point function f_p.

- Output $\mathsf{Auth}_p(|\psi\rangle\langle\psi|)$.

Eval

Input: A state σ and a string x.

- Run $\mathsf{Verf}_x(\sigma)$.
- If Verf_x accepts, output 1. Else, 0.

The Verify procedure checks if $\mathsf{Eval}(\sigma, x) = C_p(x)$ for a suitably sampled x.

5. From Copy-Protection to Secure Software Leasing

A copy protection scheme is essentially an SSL scheme with a different security guarantee. There is no Verify procedure. Instead, we guarantee that no pirate given a single program state can create two states which can be used to evaluate the underlying circuit. This is tested by challenging two evaluators to compute the circuit on given inputs.

We consider a variation called **honest-malicious copy-protection** where one of the evaluators must use the honest evaluation procedure.

Theorem. Under mild conditions, any honest-malicious copy-protection scheme yields a secure software leasing scheme.

Theorem. Our construction offers honest-malicious copy protection.

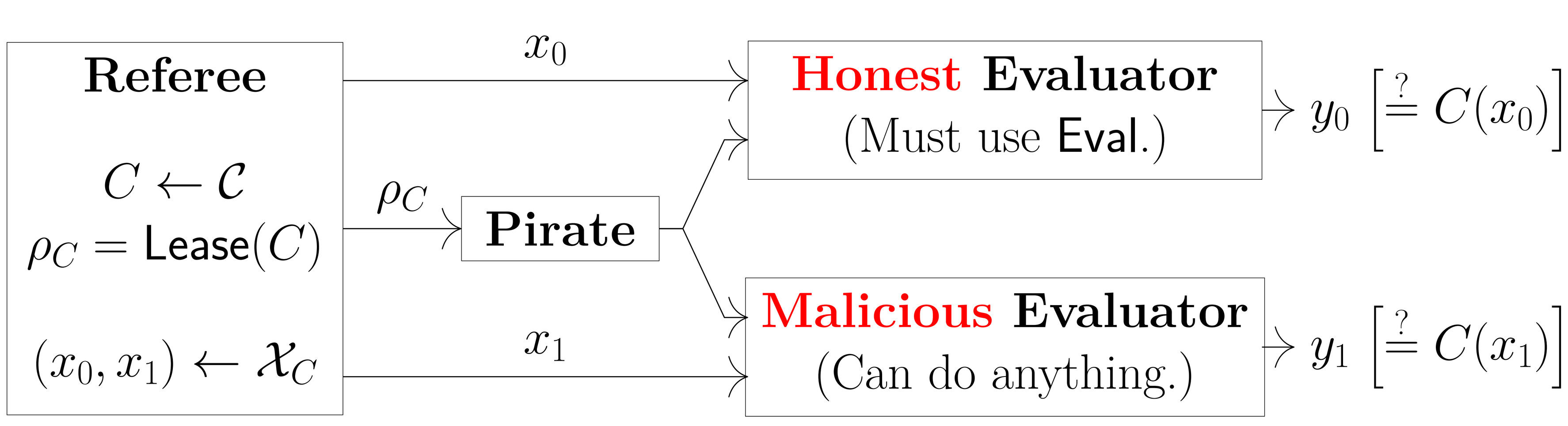

Figure 3: The honest-malicious copy-protection security game.

References

[Aar09] Aaronson. Quantum copy-protection and quantum money. 2009.

[ALP21] Ananth and La Placa. Secure software leasing. 2021.

[CMP20] Coladangelo, Majenz, and Poremba. Quantum copy-protection of compute-and-compare programs in the quantum random oracle model. 2020.

[GYZ17] Garg, Yuen, and Zhandry. New security notions and feasibility results for authentication of quantum data. 2017.

Acknowledgements

We thank Christian Majenz and Martti Karvonen for related discussions. This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-17-1-0083, Canada's NFRF and NSERC, an Ontario ERA, and the University of Ottawa's Research Chairs program. SJ is a CIFAR Fellow in the Quantum Information Science program.