Quantum Copy-Protection and Quantum Money

Scott Aaronson*

Abstract

Forty years ago, Wiesner proposed using quantum states to create money that is physically impossible to counterfeit, something that cannot be done in the classical world. However, Wiesner's scheme required a central bank to verify the money, and the question of whether there can be unclonable quantum money that anyone can verify has remained open since. One can also ask a related question, which seems to be new: can quantum states be used as copy-protected programs, which let the user evaluate some function f, but not create more programs for f?

This paper tackles both questions using the arsenal of modern computational complexity. Our main result is that there exist quantum oracles relative to which publicly-verifiable quantum money is possible, and any family of functions that cannot be efficiently learned from its input-output behavior can be quantumly copy-protected. This provides the first formal evidence that these tasks are achievable. The technical core of our result is a "Complexity-Theoretic No-Cloning Theorem," which generalizes both the standard No-Cloning Theorem and the optimality of Grover search, and might be of independent interest. Our security argument also requires explicit constructions of quantum t-designs.

Moving beyond the oracle world, we also present an explicit candidate scheme for publicly-verifiable quantum money, based on random stabilizer states; as well as two explicit schemes for copy-protecting the family of point functions. We do not know how to base the security of these schemes on any existing cryptographic assumption. (Note that without an oracle, we can only hope for security under some computational assumption.)

*MIT. Email: aaronson@csail.mit.edu. Partly supported by the Keck Foundation. Some of this work was done while the author was at the University of Waterloo.

1 Introduction

In classical physics, any information that can be read can be copied an unlimited number of times—which is why the makers of software, music CDs, and so on have met such severe difficulties enforcing "digital rights management" on their products (see Halderman [15] for example). Quantum states, on the other hand, cannot in general be copied, since measurement is an irreversible process that destroys coherence. And this immediately raises the possibility of using quantum states as *unclonable information*, such as money or copy-protected software. The idea of using quantum states in this way actually predates quantum information as a field. In a remarkable 1970 manuscript that first discussed the idea of quantum cryptography, Wiesner [23] also proposed a scheme for "quantum money" that a central bank can prepare and verify, but that is information-theoretically impossible for anyone other than the bank to copy. Wiesner's result immediately raised a question: could there be quantum money states that *anyone* can verify—that is, for which the authentication procedure is completely public—but that are still infeasible to copy? This latter question has remained open for forty years.

However, while the quantum money problem is fascinating by itself, it also motivates a broader problem: *what sorts of "unclonable power" can be provided by a quantum state?* So for example, given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, one can ask: does there exist a quantum state $|\psi_f\rangle$ that lets its owner compute f in polynomial time, but does *not* let her efficiently prepare more states that are also useful for computing f?[1] Such a state could be interpreted as "quantumly copy-protected software." Whereas in the quantum money problem, we wanted unclonable states that could be *verified as authentic*, in the quantum copy-protection problem we want unclonable states that let us *do something useful* (namely, compute f). There are other interesting unclonable functionalities (such as identity

[1] Formally, of course, we would want a scheme that worked for a whole family of f's.

cards), but in this paper, money and copy-protected software will be more than enough to occupy us.

A first, crucial observation is that, if we insist on *information-theoretic security* (as provided, for example, by quantum key distribution), then we cannot hope for either quantum copy-protection *or* publicly-verifiable quantum money. The reason is simple: an adversary with unlimited computational power could loop through all possible quantum states $|\psi\rangle$, halting only when it found a state with the required properties.[2] Therefore, if we want these functionalities, we are going to have to make computational hardness assumptions. However, this does not by any means defeat the purpose—for remember that in the classical world, the functionalities we are talking about are flat-out impossible, regardless of what computational assumptions we make.

In our view, unclonable information remains one of the most striking potential applications of quantum mechanics to computer science. Firstly, unclonable information would solve problems of clear, longstanding, and undisputed importance in the classical world—in the case of money that cannot be counterfeited, a problem that people have been trying to solve for thousands of years. Secondly, unlike with (say) quantum cryptography, the problems being addressed here are ones for which theoretically-grounded classical alternatives simply *do not exist*, because of the copyability of classical information.[3] Thirdly, as we will see, some quantum money proposals require no multi-qubit entanglement, and might therefore be technologically feasible long before general-purpose quantum computing.[4]

Given all this, it is surprising that the questions of unclonable quantum money and software have barely been taken up over the last two decades. The goal of this paper is to revisit these questions using the arsenal of modern theoretical computer science. Our main result (stated informally) is the following:

[2] In the copy-protection case, the property of $|\psi\rangle$ that we care about is that of "being a valid quantum program for the function f." And even if an explicit description of f is not available, this property can be checked using unlimited computational power, together with polynomially many copies of $|\psi_f\rangle$ (the "store-bought" quantum program for f).

[3] Here we are leaving aside solutions that involve repeated interaction with a server: we seek solutions in which the cash, software, etc. can be placed under the complete physical control of the user.

[4] On the other hand, quantum money must be protected from decohering, and this remains the central technological obstacle to realizing it. Depending on the physical substrate, right now qubits can be stored coherently for a few seconds or at most minutes, and only in laboratory conditions. By contrast, quantum key distribution (QKD) requires only the *transmission* of coherent qubits and not their long-term *storage*—which is why QKD can be implemented even with today's technology.

Theorem 1 *There exists a quantum oracle U relative to which publicly-verifiable quantum money and quantum copy-protection of arbitrary*[5] *software are possible.*[6]

Here a "quantum oracle," as defined by Aaronson and Kuperberg [4], is just an infinite collection of unitary operations $U = \{U_n\}_{n\geq 1}$ that can be applied in a black-box fashion. Theorem 1 implies that, if quantum money and copy-protection are *not* possible, then any proof of that fact will require "quantumly non-relativizing techniques": techniques that are sensitive to the presence of a quantum oracle. Such a proof is almost certainly beyond present-day techniques.

However, we also go beyond oracle results, and propose *the first explicit candidate schemes for publicly-verifiable quantum money and for copy-protecting the family of point functions*. Here a "point function" is a Boolean function $f_s : \{0,1\}^n \rightarrow \{0,1\}$ such that $f_s(x) = 1$ if and only if x equals some secret string s. Copy-protecting point functions has an interesting application for computer security: it yields a way to distribute a password-authentication program such that, not only can one not learn the password by examining the program, one cannot even use the program to create additional programs with the ability to *recognize* the password.

Our candidate quantum money scheme is based on random stabilizer states; the problem of counterfeiting the money is closely related to noisy decoding for random linear codes over $\mathbb{GF}_2$. For copy-protecting point functions, we actually give *two* schemes: one based on random quantum circuits (as recently studied by Harrow and Low [17]), the other based on hidden subgroups of the symmetric group. The key challenge, which we leave unresolved, is to base the security of our schemes on a "standard" cryptographic assumption (for example, the existence of pseudorandom functions secure against quantum attack), as opposed to the tautological assumption that our schemes are secure!

Our results give the first complexity-theoretic evidence that quantum copy-protection and publicly-verifiable quantum money are indeed possible. On the other hand, the oracle results also help explain the *difficulty* of proving explicit schemes for these tasks secure. For as we will see, proving security in the oracle

[5] By which we mean, software that is not learnable from its input/output behavior using a polynomial-time quantum computation. Learnable software is impossible to copy-protect for trivial reasons. Our result shows that relative to an oracle, this is the only obstruction.

[6] For technical reasons, the copy-protection result currently only gives security against pirating algorithms that *more than double* the number of programs. We hope to remove this restriction in the future.

world is already highly nontrivial! Furthermore, *any security proof for an explicit scheme will need to include our oracle result as a special case*—since an attack on an explicit scheme could always proceed by treating all the relevant circuits as black boxes and ignoring their internal structure.

1.1 Techniques

In proving our oracle results, perhaps the most novel technical ingredient is what we call the "Complexity-Theoretic No-Cloning Theorem":

Theorem 2 (Complexity-Theoretic No-Cloning) *Let $|\psi\rangle$ be an n-qubit pure state. Suppose we are given the initial state $|\psi\rangle^{\otimes k}$ for some $k \geq 1$, as well as an oracle U_ψ such that $U_\psi |\psi\rangle = -|\psi\rangle$ and $U_\psi |\phi\rangle = |\phi\rangle$ for all $|\phi\rangle$ orthogonal to $|\psi\rangle$. Then for all $\ell > k$, to prepare ℓ registers $\rho_1, \ldots, \rho_\ell$ such that*

$$\sum_{i=1}^{\ell} \langle\psi| \rho_i |\psi\rangle \geq k + \delta,$$

we need

$$\Omega\left(\frac{\delta^2 \sqrt{2^n}}{\ell^2 k \log k} - \ell\right)$$

queries to U_ψ.

Intriguingly, Theorem 2 can be seen as a common generalization of the No-Cloning Theorem and the BBBV lower bound for quantum search [8].[7] It reduces to the No-Cloning Theorem if we ignore the oracle U, and it reduces to the BBBV lower bound if we ignore the initial state $|\psi\rangle$.

The proof of Theorem 2 proceeds in two steps. First we lower-bound the query complexity of cloning $|\psi\rangle$ almost *perfectly*, by using a generalization of Ambainis's quantum adversary method [5] that we design specifically for the purpose. Next we argue that, if we could even clone $|\psi\rangle$ with non-negligible fidelity, then with polynomially more queries we could also clone $|\psi\rangle$ almost perfectly, by using a recent fixed-point quantum search algorithm of Grover [14].

We regret that, due to space limitations, we are not able to include a proof of Theorem 2 in this extended abstract.

With Theorem 2 in hand, it is not hard to show the existence of a quantum oracle U relative to which a publicly-verifiable quantum money scheme exists. We simply choose n-qubit quantum banknotes uniformly at random under the Haar measure, and then "offload" all the work of preparing and recognizing the banknotes onto the oracle. Theorem 2 then implies that, even given $k = \text{poly}(n)$ valid banknotes, a would-be counterfeiter needs exponentially many queries to U to prepare a $(k+1)^{\text{st}}$ banknote. Crucially, our oracle construction is "fair," in the sense that the bank, the customers, and the counterfeiters all have access to the same oracle U, and none of them have any special knowledge about U not shared by the others. This is why we believe our result merits the informal interpretation we have given it: namely, that any impossibility proof for quantum money would have to be non-relativizing.

Showing the existence of a quantum oracle U relative to which quantum copy-protection works is a harder problem. As in the money case, we choose n-qubit "quantum programs" $|\psi_f\rangle$ uniformly at random under the Haar measure, and then define a quantum oracle U that is able both to prepare $|\psi_f\rangle$ given a description of f, and to evaluate $f(x)$ given $|\psi_f\rangle$ and x. However, a new difficulty is that some families of Boolean functions $\mathcal{F}$ *cannot* be copy-protected: namely, those for which any $f \in \mathcal{F}$ can be efficiently learned using black-box access. Thus, our proof somehow needs to explain why learnability is the only obstruction. Our solution will be to construct a polynomial-time *simulator*, which takes an algorithm (in the oracle world) for pirating a quantum program $|\psi_f\rangle$, and converts it into an algorithm (with no oracle) that learns f using only black-box access to f.

Among other things, the simulator needs the ability to "mock up" its own quantum state $|\varphi\rangle$ that can stand in for $|\psi_f\rangle$ in a simulation of the pirating algorithm, which in turn means that $|\varphi\rangle^{\otimes t}$ should be indistinguishable from t copies of a Haar-random state for some fixed $t = \text{poly}(n)$. As it turns out, precisely this problem—the construction of explicit quantum states that behave like Haar-random states—has recently become a major topic in quantum computing. So for example, Ambainis and Emerson [6] gave an explicit construction of *approximate quantum t-designs* for arbitrary t: that is, finite ensembles of pure states $(p_x, |\varphi_x\rangle)$ such that

$$\begin{aligned}(1-\varepsilon)\int_\psi (|\psi\rangle\langle\psi|)^{\otimes t}\, d\psi &\leq \sum_x p_x (|\varphi_x\rangle\langle\varphi_x|)^{\otimes t} \\ &\leq (1+\varepsilon)\int_\psi (|\psi\rangle\langle\psi|)^{\otimes t}\, d\psi\end{aligned}$$

where the integrals are with respect to the Haar measure. Our requirement is slightly different: basically, we need that no algorithm that receives t copies of $|\varphi\rangle$,

[7] Here by "quantum search," we mean search for an unknown pure state $|\psi\rangle$, which need not be a computational basis state. As far as we know, this generalization of the usual Grover problem was first studied by Farhi and Gutmann [12].

and makes T queries to an oracle that recognizes $|\varphi\rangle$, can decide whether $|\varphi\rangle$ was drawn from the explicit distribution or the Haar measure. Both for that reason, and because our construction was independent of [6], in the full version of the paper we give a self-contained proof of the following result:

Theorem 3 *Let d be a positive integer. Then there exists a collection of n-qubit pure states $\{|\varphi_x\rangle\}_{x\in\{0,1\}^{n(d+1)}}$ such that:*

(i) Given x as input, the state $|\varphi_x\rangle$ can be prepared in time polynomial in n and d.

(ii) Let E be any quantum algorithm that receives a state $|\varphi\rangle^{\otimes t}$ as input, and also makes T queries to a quantum oracle U_φ such that $U_\varphi|\varphi\rangle = -|\varphi\rangle$ and $U_\varphi|\phi\rangle = |\phi\rangle$ for all $|\phi\rangle$ orthogonal to $|\varphi\rangle$. Let $E(|\varphi\rangle)$ represent the probability that E accepts. Then provided $t + 2T \leq \min\left\{d/2, \sqrt{2^n/2}\right\}$, we have

$$\left| \begin{array}{c} \mathrm{EX}_{x\in\{0,1\}^{n(d+1)}}\left[E\left(|\varphi_x\rangle\right)\right] \\ -\,\mathrm{EX}_{|\psi\rangle\in\mu}\left[E\left(|\psi\rangle\right)\right] \end{array} \right| \leq \frac{4\left(t+2T\right)^2}{2^n}$$

where μ is the Haar measure.

For those who are curious, the explicit states in question are

$$|\varphi_x\rangle := \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{GF}(2^n)} e^{2\pi i p(x)/2^n} |x\rangle,$$

where $p : \mathbb{GF}(2^n) \to \mathbb{GF}(2^n)$ is a univariate polynomial of degree at most d that is encoded by the string $x \in \{0,1\}^{n(d+1)}$, and elements of $\mathbb{GF}(2^n)$ are freely reinterpreted as n-bit integers where relevant.

1.2 Related Work

Recall that quantum money was first studied by Wiesner [23]. In Wiesner's scheme, a central bank distributes "quantum banknotes," each consisting of a unique serial number (which is written down classically), together with n polarized photons in the states $|0\rangle$, $|1\rangle$, $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$, or $|-\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$. The bank also stores, in a secure location, a database of all the serial numbers together with classical descriptions of the associated quantum states. Whenever a banknote is returned to the bank, the note can be measured (using the secure database) to verify its authenticity. On the other hand, using the uncertainty principle, it is possible to show that, starting from k banknotes, any attempt to forge $k+1$ banknotes that all pass the authentication test can succeed with probability at most $(3/4)^n$.

Let us point out two striking advantages of Wiesner's scheme. Firstly, the scheme requires only single coherent qubits and one-qubit measurements; there is no need for any entanglement. For this reason, the scheme might be practical long before universal quantum computing. Secondly, the security of the scheme is *information-theoretic*—guaranteed by the laws of quantum physics—rather than computational.

An obvious drawback of Wiesner's scheme is its need for a giant secret database maintained by the bank. But in 1982, Bennett, Brassard, Breidbart, and Wiesner [9] (henceforth BBBW) showed how to avoid the giant database, at the cost of making the security of the quantum money computational rather than information-theoretic. In modern terms, their proposal was this. The bank fixes, once and for all, a secret random seed s. It then distributes banknotes, each of the form $|y\rangle\left|\psi_{g_s(y)}\right\rangle$, where $y \in \{0,1\}^n$ is a unique serial number for the banknote, $g_s : \{0,1\}^n \to \{0,1\}^n$ is a pseudorandom function, and $\left|\psi_{g_s(y)}\right\rangle$ is the state obtained by starting from $g_s(y)$, grouping the n bits into $n/2$ blocks of two, and mapping each 00 to $|0\rangle$, 01 to $|1\rangle$, 10 to $|+\rangle$, and 11 to $|-\rangle$.

Using its knowledge of s, the bank can verify the authenticity of any note $|y\rangle\left|\psi_{g_s(y)}\right\rangle$, by computing $g_s(y)$ and then measuring each qubit of $\left|\psi_{g_s(y)}\right\rangle$ in the appropriate basis. But suppose g_s were a truly random function. Then by the same argument as for Wiesner's original scheme, given any k banknotes, no quantum operation could forge a $(k+1)^{\text{st}}$ note with probability more than $(3/4)^{n/2}$ of passing the authentication test. This means that, if there *were* a quantum operation to forge high-quality banknotes, then that operation could be used to distinguish g_s from a truly random function. And therefore, assuming g_s is secure against polynomial-time quantum adversaries, there can be no polynomial-time quantum algorithm to forge banknotes that pass the authentication test with non-negligible probability.

However, the BBBW scheme still has a serious drawback: namely that s, which is needed for the authentication procedure, must remain a closely-guarded secret. And thus it would presumably be unwise, for example, to install the authentication devices in convenience-store cash registers. What we really want is a scheme where the procedure for *authenticating* the money is completely public, and only the procedure for minting the money is secret.

Bennett et al. [9] presented a candidate for such a publicly-verifiable quantum money scheme,[8] which was based on the hardness of factoring Blum integers.[9] Unfortunately, their scheme was insecure for two reasons. First, we now know that factoring is in quantum polynomial time! But even were we to base the scheme on some other cryptographic primitive, Bennett et al. pointed out that it could be broken by an adversary who is able to make entangled measurements on all of the qubits in a banknote. The question of whether secure quantum money with public authentication is possible has remained open for 30 years.

Concurrently with our work, there has been a recent renewal of interest in the quantum money problem. In his PhD thesis, Stebila [21] provides a lucid overview of quantum money, and explains why our Complexity-Theoretic No-Cloning Theorem implies the existence of a quantum oracle relative to which publicly-verifiable quantum money is possible.[10] Also, in unpublished work, Aaronson, Farhi et al. [2] propose a candidate scheme for publicly-verifiable quantum money that is quite different from the scheme in this paper. Specifically, the scheme of [2] is based on random quantum k-SAT instances whose ground states are matrix product states.

As far as we know, the idea of using quantum mechanics to copy-protect software is original to this work.

1.3 Organization

The rest of this extended abstract is organized as follows. Section 2 formally defines quantum money and copy-protection schemes and investigates their basic properties, and also recalls some preliminaries from cryptography and quantum information. Section 3 considers quantum money schemes: our explicit candidate proposal based on random stabilizer states in Section 3.1, and our oracle result in Section 3.2. Section 4 then discusses quantum copy-protection: the candidate schemes for copy-protecting point functions in Section 4.1, and the oracle result in Section 4.2. We conclude in Section 5 with a list of open problems. We regret that, because of space limitations, much of the paper's technical content (including the proof of the Complexity-Theoretic No-Cloning Theorem and the explicit construction of quantum t-designs) has had to be relegated to the full version.

[8] Bennett et al. described their public-key scheme in terms of "subway tokens" rather than money—since if we want to authenticate the tokens using single-qubit measurements only, then the authentication test necessarily destroys the tokens and prevents their reuse. On the other hand, supposing we could perform an entangled measurement on all n qubits in a token, it would be possible to authenticate the token while preserving its quantum coherence. For this reason, the token could be used as money.

[9] More generally, their scheme could be based on any *trapdoor collision-resistant hash function*: that is, a CRHF such that one can efficiently sample collision pairs using some hidden trapdoor information.

[10] Indeed, our original interest was in copy-protection; it was Stebila, along with M. Mosca, who pointed out to us the application to unforgeable money.

2 Preliminaries

For simplicity, in this paper we restrict ourselves to nonuniform (circuit) computation. Given two mixed states ρ and σ, the *trace distance* $\|\rho-\sigma\|_{\mathrm{tr}}$ equals the maximum, over all measurements M, of the variation distance $\|M(\rho)-M(\sigma)\|$ between the probability distributions $M(\rho), M(\sigma)$ over measurement outcomes obtained by applying M to ρ and σ respectively. We will use the following lemma of Aaronson [1]:

Lemma 4 ("Almost As Good As New Lemma") *Suppose a measurement on a mixed state ρ yields a particular outcome with probability $1-\varepsilon$. Then after the measurement, one can recover a state $\widetilde{\rho}$ such that $\|\widetilde{\rho}-\rho\|_{\mathrm{tr}} \leq \sqrt{\varepsilon}$.*

See Nielsen and Chuang [18] for other quantum information concepts used in this paper.

In what follows, we will sometimes use the assumption that there exists a pseudorandom function family secure against quantum adversaries. The following theorem helps to justify that assumption.

Theorem 5 *Suppose there exists a one-way function $A:\{0,1\}^n \rightarrow \{0,1\}^n$ that is secure against $2^{n^{\Omega(1)}}$-time quantum adversaries. Then there also exists a family $f_s:\{0,1\}^n \rightarrow \{0,1\}^n$ of pseudorandom functions, parameterized by a seed $s \in \{0,1\}^{\mathrm{poly}(n)}$, that is secure against 2^n-time quantum adversaries. (Here A and f_s are both computable in classical polynomial time.)*

Proof Sketch. Håstad et al. [20] showed that if $2^{n^{\Omega(1)}}$-secure one-way functions exist, then so do $2^{n^{\Omega(1)}}$-secure pseudorandom generators. Razborov and Rudich [19] showed that if $2^{n^{\Omega(1)}}$-secure pseudorandom generators exist, then so do 2^n-secure pseudorandom function families (with polynomial seed length). Since both of these reductions are "black-box," they go through essentially without change if the adversary is quantum. ■

Interestingly, the reduction of Goldreich, Goldwasser, and Micali [13], from $f(n)$-secure pseudorandom generators to $f(n)^{\Omega(1)}/\mathrm{poly}(n)$-secure pseudorandom functions with seed length n, does *not* go through if the adversary is quantum.[11] We leave as an open problem whether a "strong," GGM-style reduction from PRGs to PRFs can be proved in the quantum setting.

2.1 Quantum Money

Intuitively, a quantum money scheme is a scheme in which

(1) quantum banknotes can be efficiently produced by a central bank,

(2) there exists a polynomial-time quantum algorithm for authenticating the banknotes, which could be private or public, and

(3) given as input k valid banknotes, a polynomial-time counterfeiter cannot produce $k+1$ valid banknotes that have non-negligible probability of passing the authentication test.

We now give a formal definition.

Definition 6 *A quantum money scheme with key size n consists of the following:*

- *A quantum circuit B of size $O(\text{poly}(n))$ (the "bank"), which takes a string $s \in \{0,1\}^n$ (the "secret key") as input, and produces a classical string e_s (the "public key") and mixed state ρ_s (the "banknote") as output.*[12]

- *A quantum circuit A of size $O(\text{poly}(n))$ (the "authenticator"), which takes a string e and state ρ as input and either accepts or rejects.*

We say (B, A) has completeness error ε if $A(e_s, \rho_s)$ accepts with probability at least $1-\varepsilon$ for all s. We say (B, A) has soundness error δ if for all quantum circuits C of size $O(\text{poly}(n))$ (the "counterfeiter") and all $k, r = O(\text{poly}(n))$, the following holds. Assume C takes $\rho_s^{\otimes k}$ as input, and outputs a state σ_s on $k+r$ registers. For $i \in [k+r]$, let σ_s^i denote the contents of the i^{th} register, and let p_i be the probability that $A(e_s, \sigma_s^i)$ accepts, averaged over all $s \in \{0,1\}^n$. Then $\sum_{i=1}^{k+r} p_i \leq k + \delta$.

We call (B, A) public-key if C also receives e_s as input, and private-key otherwise. If (B, A) is private-key, we call it query-secure if C has access to an oracle that takes a state σ as input and simulates $A(e_s, \sigma)$ (that is, accepts with the same probability and returns the same post-measurement state $\widetilde{\sigma}$).[13]

We make a few remarks on Definition 6. First, it is obvious that no money scheme exists where the states ρ_s are classical.[14] Second, if a money scheme has completeness error ε, it follows from Lemma 4 that the authentication procedure can return a banknote $\widetilde{\rho}_s$ such that $\|\widetilde{\rho}_s - \rho_s\| \leq \sqrt{\varepsilon}$. This means that the same banknote can be verified $\Omega(1/\sqrt{\varepsilon})$ times before it needs to be replaced. In this paper, we will generally be interested in schemes with perfect completeness.

Third, we will generally want the soundness error δ to be negligible (that is, $o(1/p(n))$ for all polynomials p). If δ is negligible, then it is easy to see that, starting from $\rho_s^{\otimes k}$, no polynomial-time counterfeiter C can ever increase its "wealth" (defined as the expected number of states in C's possession that A accepts) by more than a negligible amount in expectation. Note that this is true even if the states output by C are entangled; our definition automatically accounts for this possibility.

We now discuss some examples. The BBBW scheme [9], discussed in Section 1.2, is a private-key quantum money scheme. We therefore have the following:

Theorem 7 (implicit in [9]) *If there exists a pseudorandom function family secure against quantum adversaries, then there exists a private-key quantum money scheme with perfect completeness and exponentially small soundness error.*

However, the BBBW scheme is *not* query-secure. The reason is simple: given a banknote of the form $|y\rangle \left|\psi_{g_s(y)}\right\rangle$, a counterfeiter can learn a classical description of $\left|\psi_{g_s(y)}\right\rangle$, by rotating each qubit i in turn while leaving the other $n/2-1$ qubits fixed, and repeatedly feeding the result to the authenticator A until it has ascertained the correct state of the i^{th} qubit. This works because A always measures the qubits in the correct bases, and therefore does not damage the qubits that are not being rotated. Of course, once the counterfeiter has learned a classical description of $\left|\psi_{g_s(y)}\right\rangle$,

[11] This is because the GGM reduction makes essential use of the fact that a polynomial-time adversary can examine only $\text{poly}(n)$ outputs of the function f_s—something that is manifestly false if the adversary is quantum.

[12] One can of course generalize the definition to let e_s be randomized or even a quantum state, and possibly correlated with ρ_s as well. However, we will not need the additional freedom in this paper.

[13] Note that any public-key scheme is also query-secure, since we can hardwire a description of A into C.

[14] Furthermore, no query-secure scheme can exist where the states ρ_s have $O(\log n)$ qubits. For in that case, a counterfeiter could reconstruct ρ_s in polynomial time, by first generating a tomographically complete set of states, and then sending several copies of each state to A to estimate the probability that each one is accepted.

it can then produce as many copies of $|y\rangle\left|\psi_{g_s(y)}\right\rangle$ as it likes.

In the full version of this paper, we will give a private-key quantum money scheme that *is* query-secure, assuming the existence of pseudorandom functions secure against quantum adversaries. We will also prove the following result, which is not entirely trivial:

Theorem 8 *Any quantum money scheme satisfying Definition 6 (even a private-key one) must rely on some computational assumption.*

As mentioned in Section 1.2, Wiesner's original scheme [23] avoided the need for any computational assumption, but only by having the bank maintain a giant lookup table, containing a classical description of every banknote that has ever been issued. If we want to fit Wiesner's scheme into Definition 6, one way to do it is to assume that all parties have access to a (classical) random oracle $\mathcal{O}$. For then the bank can use a secret part of the oracle string to generate the banknotes $|y\rangle|\psi_y\rangle$; and to any counterfeiter who does not know which part of the oracle the bank is using, the states $|\psi_y\rangle$ will appear to be drawn uniformly from $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}^n$. This observation gives us the following:

Theorem 9 (implicit in [23]) *Relative to a random oracle $\mathcal{O}$, there exists a private-key quantum money scheme with perfect completeness and exponentially small soundness error.*

On the other hand, Wiesner's scheme is not query-secure, for the same reason the BBBW scheme is not. In the full version of this paper, we give a private-key quantum money scheme that *is* query-secure, relative to a random oracle $\mathcal{O}$.

In Section 3.1, we will present a candidate for a public-key quantum money scheme based on random stabilizer states, while in Section 3.2, we will prove that public-key quantum money schemes exist relative to a quantum oracle.

The situation is summarized in Table 2.1.

2.2 Quantum Copy-Protection

What if we want to distribute unclonable quantum states that are *useful* for something besides just getting authenticated? This brings us to the question of *quantum software copy-protection*. Informally, given a secret Boolean function $f : \{0,1\}^n \to \{0,1\}$ drawn from a known family $\mathcal{F}$, what we want is a quantum state ρ_f that

(1) can be efficiently prepared given a classical description of f,

(2) can be used to compute $f(x)$ efficiently for any input $x \in \{0,1\}^n$, and

(3) *cannot* be efficiently used to prepare more states from which f can be computed in quantum polynomial time.

It is clear that, if the function family $\mathcal{F}$ is efficiently *learnable*—in the sense that we can output a circuit for an unknown $f \in \mathcal{F}$ in quantum polynomial time, using only oracle access to f—then there is no hope of copy-protecting f. For in that case, being able to run a program for f is tantamount to being able to copy the program. Indeed, even if we cannot learn a useful *classical* description of f by measuring ρ_f, it might still be possible to prepare additional quantum programs for f directly, by some quantum operation on ρ_f.

The quantum copy-protection problem might remind readers of the classical *code obfuscation* problem, and indeed there are similarities. Roughly speaking, we say a program P for a function $f \in \mathcal{F}$ is obfuscated if knowing P's source code is "no more useful" than being able to run P, in the sense that any property of f that is efficiently computable given P's source code, is also efficiently computable given oracle access to f. Barak et al. [7] famously showed that there exist function families $\mathcal{F}$ that are impossible to obfuscate. On the other hand, Wee [22] and others have shown that, under strong cryptographic assumptions, it *is* possible to obfuscate point functions and several related families of functions. In Section 4.1, we will give proposals for quantumly copy-protecting point functions that are somewhat reminiscent of known methods for obfuscating point functions.

However, let us point out two differences between copy-protection and obfuscation. Firstly, it is trivial to show that copy-protection is *always* impossible in the classical world, for *any* function family: one does not need anything like the elegant argument of Barak et al. [7]. Secondly, as discussed before, any function family $\mathcal{F}$ that is learnable from input/output behavior cannot be copy-protected—but for exactly the same reason, $\mathcal{F}$ *can* be obfuscated! For if we can output a program for $f \in \mathcal{F}$ using only oracle access to f, then clearly the source code of that program is no more useful than the oracle access. Thus, while unbreakable copy-protection has connections with obfuscation, fundamentally it is a new cryptographic task, one whose very possibility depends on quantum mechanics.

We now define quantum copy-protection schemes.

Money Scheme	Type	Oracle	Security	States Used	Reference
Wiesner	Private-key	Random	Unconditional	Single qubits	[23]
BBBW	Private-key	None	Assuming PRFs	Single qubits	[9]
Modified Wiesner	Query-secure	Random	Unconditional	Haar-random	Full version
Modified BBBW	Query-secure	None	Assuming PRFs	Haar-random	Full version
Quantum Oracle	Public-key	Quantum	Unconditional	Haar-random	This paper
Random Stabilizers	Public-key	None	Conjectured	Stabilizer	This paper
Matrix Product States	Public-key	None	Conjectured	MPS	[2]

Table 1. Known quantum money schemes and their properties

Definition 10 *Consider a family $\mathcal{F}$ of Boolean functions $f:\{0,1\}^n \rightarrow \{0,1\}$, where each $f \in \mathcal{F}$ is associated with a unique "description" $d_f \in \{0,1\}^m$. (Thus $|\mathcal{F}| \leq 2^m$.) A quantum copy-protection scheme for $\mathcal{F}$ consists of the following:*

- *A quantum circuit V of size $O(\mathrm{poly}(n,m))$ (the "vendor"), which takes d_f as input and produces a mixed state ρ_f as output.*
- *A quantum circuit C of size $O(\mathrm{poly}(n,m))$ (the "customer"), which takes (ρ_f, x) as input and attempts to output $f(x)$.*

We say (V,C) has correctness parameter ε if C outputs $f(x)$ with probability at least $1-\varepsilon$ given (ρ_f, x) as input, for all $f \in \mathcal{F}$ and $x \in \{0,1\}^n$.

We say (V,C) has security δ against a probability distribution $\mathcal{D}$ over $\mathcal{F} \times \{0,1\}^n$, if for all quantum circuits P and L of size $O(\mathrm{poly}(n,m))$ (the "pirate" and "freeloader" respectively) and all $k, r = O(\mathrm{poly}(n,m))$, the following holds. Assume P takes $\rho_f^{\otimes k}$ as input, and outputs a state σ_f on $k+r$ registers. For $i \in [k+r]$, let σ_f^i denote the contents of the i^{th} register. Also, suppose L takes (ρ_f, x) as input and attempts to output $f(x)$. Then if we run L on (σ_f^i, x) for all $i \in [k+r]$, the expected number of invocations that output $f(x)$, averaged over (f,x) drawn from $\mathcal{D}$, is at most $k + (1-\delta) r$.[15]

A few remarks on Definition 10. First, the security criterion might seem a bit strange. The basic motivation is that we need to ignore "trivial" pirating strategies, such as mapping the state $\rho_f^{\otimes 2}$ to

$$\frac{1}{3}\left(\rho_f \otimes \rho_f \otimes I + \rho_f \otimes I \otimes \rho_f + I \otimes \rho_f \otimes \rho_f\right),$$

which has large fidelity with ρ_f on each of the three registers. On the other hand, we also do not want to require all $k+r$ pirated programs to output the right answer *simultaneously* (with high probability and on some input x), since that criterion is too stringent even for legitimate programs with constant error. Looking at the expected number of correct answers is convenient, since by linearity of expectation, we can then ignore entanglement and classical correlations among the registers. Note that it is always possible for $(1-\varepsilon)k + r/2$ of the $k+r$ pirated programs to get the right answers on average—using a pirating strategy that outputs the legitimate programs $\rho_f^{\otimes k}$, alongside r programs that guess randomly on every input x. But ideally it should not be possible to do too much better than that.

Second, a natural question is whether the state ρ_f can be used more than once, or whether the irreversibility of measurement makes such a state "disposable." In our setting, disposable states might actually be *preferred*—since any disposable state is copy-protected by definition! (If we could copy ρ_f with high fidelity, then we could run each copy on a different input x, contrary to assumption.) However, it is not hard to see that, provided the customer buys $k = \Omega(n)$ copies of ρ_f from the quantum software store, she can evaluate f on as many inputs as she likes—indeed, all 2^n of them, if she has exponential time. For by standard amplification, $\rho_f^{\otimes k}$ can be used to evaluate f with error probability $2^{-\Omega(k)}$. So by Lemma 4, it is possible to reuse $\rho_f^{\otimes k}$ an exponential number of times, by uncomputing garbage after each measurement.

In this paper, we will typically assume that ρ_f "comes from the store" already amplified, and that both customers and would-be software pirates can therefore reuse ρ_f as many times as needed. This raises an interesting point: given an amplified state $\rho_f = \sigma^{\otimes k}$, a customer willing to tolerate slightly higher error could always split ρ_f into $\sigma^{\otimes k/2} \otimes \sigma^{\otimes k/2}$, and give one of the copies of $\sigma^{\otimes k/2}$ to a friend (rather like donating a kidney). We leave as an open question whether it is possible to amplify success probability in a way that does not allow this sort of sharing.

[15] One might also want to require a concentration inequality—e.g. that for all inputs x, the probability that at least $k + 2r/3$ of the pirated programs output $f(x)$ correctly decreases exponentially with r. This is a topic we leave to future work.

Third, call the function family $\mathcal{F}$ and distribution $\mathcal{D}$ *quantumly learnable with error* δ if there exist polynomial-size quantum circuits Q and C such that

$$\Pr_{(f,x)\in\mathcal{D}}\left[C\left(Q^f,x\right) \text{ outputs } f\left(x\right)\right] \geq 1-\delta,$$

where Q^f denotes the mixed state output by Q given oracle access to f. (Note that Q does not receive x.) The following simple proposition delimits the function families that one can hope to copy-protect.

Proposition 11 *No $(\mathcal{F},\mathcal{D})$ pair that is quantumly learnable with error δ can be quantumly copy-protected with security $\delta+2^{-n}$.*

Proof. Using an amplified state of the form $\rho_f^{\otimes \operatorname{poly}(n)}$, a pirate can *simulate* quantum oracle access to f with exponentially small error. The pirate can thereby use the learning algorithm Q^f to output as many states σ_f as he wants with the property that $\Pr_{(f,x)\in\mathcal{D}}\left[C\left(\sigma_f,x\right) \text{ outputs } f\left(x\right)\right] \geq 1-\delta-2^{-n}$. Note that by Lemma 4, each "query" to f damages $\rho_f^{\otimes \operatorname{poly}(n)}$ by only an exponentially small amount. ■

Notice that if $|\mathcal{F}| \leq \operatorname{poly}(n)$, then $\mathcal{F}$ is quantumly learnable (and indeed classically learnable), since the learning algorithm Q simply needs to hardwire inputs $x_1,\ldots,x_{|\mathcal{F}|-1}$ such that every distinct $f,f'\in\mathcal{F}$ differ on some x_i. Thus, one corollary of Proposition 11 is that we can only hope to copy-protect superpolynomially large function families.

Let us end with a simple but important fact, which shows that, as in the quantum money case, we can only hope for security under computational assumptions.

Proposition 12 *A software pirate with unlimited computational power can break any quantum copy-protection scheme.*

Proof. Let f and g be two functions in $\mathcal{F}$, and assume there exists an $x\in\{0,1\}^n$ such that $f(x)\neq g(x)$. Then letting ρ_f and ρ_g be the quantum programs for f and g respectively, the fidelity $F(\rho_f,\rho_g)$ must be at most ϵ, for some ϵ bounded away from 1 by a constant. (Otherwise ρ_f and ρ_g would lead to the same answers on x with $1-o(1)$ probability.) This implies that $F(\rho_f^{\otimes k},\rho_g^{\otimes k})\leq\epsilon^k$. So if we choose k sufficiently large (say, more than $2m$), then the set of states $\left\{\rho_f^{\otimes k}\right\}_{f\in\mathcal{F}}$ is extremely close to an orthonormal basis. Thus, as in the algorithm of Ettinger, Høyer, and Knill [11] for the nonabelian Hidden Subgroup Problem, there must be a measurement of $\rho_f^{\otimes k}$ (possibly exponentially hard to implement) that outputs f with high probability. ■

3 Quantum Money

We now consider the problem of developing public-key quantum money schemes. First, in Section 3.1, we propose an explicit candidate scheme for public-key quantum money, based on random stabilizer states. Then, in Section 3.2, we use the Complexity-Theoretic No-Cloning Theorem to construct a quantum oracle relative to which public-key quantum money schemes exist.

3.1 The Random Stabilizer Scheme

Recall that a *stabilizer state* is a pure state that can be obtained by starting from $|0\rangle^{\otimes n}$ and then applying controlled-NOT, Hadamard, and $\pi/4$-phase gates, while a *stabilizer measurement* is a measurement that can be performed using those gates together with computational basis measurements. (See Aaronson and Gottesman [3] for details.) Given a security parameter n, let $\mathcal{D}_n$ be the uniform distribution over all n-qubit stabilizer states. Also, let m,ℓ,ε be additional parameters such that $n/\varepsilon \ll m \ll 1/\varepsilon^2 \ll \ell$.

To generate a banknote, first the bank prepares ℓ stabilizer states $|C_1\rangle,\ldots,|C_\ell\rangle$, which are drawn independently from $\mathcal{D}_n$. (It is well-known that any stabilizer state can be prepared in polynomial time.) The bank temporarily remembers the classical descriptions of the $|C_i\rangle$'s, though it can erase those descriptions once the preparation procedure is finished. Next, for each $i\in[\ell]$, the bank generates m random stabilizer measurements $E_{i1},\ldots,E_{im}$ as follows. For each $j\in[m]$:

- With $1-\varepsilon$ probability, E_{ij} is a tensor product of n uniformly random Pauli operators, with a random phase. That is, $E_{ij}=(-1)^b P_1\otimes\cdots\otimes P_n$, where b is drawn uniformly from $\{0,1\}$, and each P_k is drawn uniformly from $\{I,\sigma_x,\sigma_y,\sigma_z\}$.

- With ε probability, E_{ij} is a random tensor product of Pauli operators as above, except that we condition on the event that $|C_i\rangle$ is a $+1$ eigenstate of E_{ij} (that is, $E_{ij}|C_i\rangle=|C_i\rangle$).

We can represent these ℓm measurements by a table $\mathcal{E}=(E_{ij})_{ij}$, using $(2n+1)\,\ell m$ classical bits. Finally, the bank generates an ordinary, classical digital signature $\operatorname{sig}(\mathcal{E})$ of the table $\mathcal{E}$, to prove that it and it alone could have generated $\mathcal{E}$. The bank then distributes $(|C_1\rangle,\ldots,|C_\ell\rangle,\mathcal{E},\operatorname{sig}(\mathcal{E}))$ as the quantum banknote.

To authenticate such a banknote, one does the following. First check that $\operatorname{sig}(\mathcal{E})$ is a valid digital signature for $\mathcal{E}$. Next, for each $i\in[\ell]$, choose an index $j(i)\in[m]$ uniformly at random. Let M be the two-outcome measurement that applies $E_{1j(1)}$ to $|C_1\rangle$, $E_{2j(2)}$ to $|C_2\rangle$, and so on up to $|C_\ell\rangle$, and that accepts if and only if the *majority* of these measurements return a $+1$ outcome (corresponding to $|C_i\rangle$ being a $+1$ eigenstate of $E_{ij(i)}$). Then apply M to $|C_1\rangle \otimes \cdots \otimes |C_\ell\rangle$, accept if and only if M accepts, and finally apply uncompute to get rid of garbage.

By construction, each $|C_i\rangle$ will be measured to be in a $+1$ eigenstate of $E_{ij(i)}$ with independent probability $\frac{1-\varepsilon}{2} + \varepsilon = 1/2 + \varepsilon/2$. So by a Chernoff bound, the probability that M rejects is bounded away from 1. Indeed, we can make the probability that M rejects exponentially small, by simply taking ℓ to be sufficiently larger than $1/\varepsilon^2$. By Lemma 4, this implies that when we uncompute, we recover a state that is exponentially close to $|C_1\rangle \otimes \cdots \otimes |C_\ell\rangle$ in trace distance—which in turn implies that we can reuse the quantum banknote an exponential number of times.

On the other hand, we conjecture the following:

Conjecture 13 *Given $(|C_1\rangle, \ldots, |C_\ell\rangle, \mathcal{E}, s)$, it is computationally infeasible not only to recover classical descriptions of the states $|C_1\rangle, \ldots, |C_\ell\rangle$, but even to prepare additional copies of these states—or for that matter, of any states that are accepted by the authentication procedure with non-negligible probability.*

The intuition behind Conjecture 13 is this: recovering classical descriptions of $|C_1\rangle, \ldots, |C_\ell\rangle$ given $\mathcal{E}$ can be seen as a random instance of the noisy decoding problem for linear codes, which is known to be NP-complete in the worst case (see Berlekamp et al. [10]). Furthermore, while it is conceivable that a counterfeiter could use her knowledge of $\mathcal{E}$ to copy the $|C_i\rangle$'s without learning classical descriptions of them, we have not found an efficient way to do this. Indeed, it seems possible that to a polynomial-time quantum algorithm—even one with knowledge of $\mathcal{E}$—the $|C_i\rangle$'s are actually indistinguishable from n-qubit maximally mixed states.

Note that the scheme is *not* secure if $m \leq n/\varepsilon$—since then finding an n-qubit stabilizer state $|C_i\rangle$ that is accepted by an ε fraction of the measurements $E_{i1}, \ldots, E_{im}$ is a trivial problem, solvable by Gaussian elimination. Likewise, the scheme is not secure if ε is too large (say, greater than $1/\sqrt{m}$)—since then one can recover the stabilizer group of $|C_i\rangle$, with high probability, by listing all measurements in the set $\{E_{i1}, \ldots, E_{im}\}$ that commute with suspiciously more than half of the other measurements in the set.[16] Thus, Conjecture 13 can only hold for suitable parameter ranges.

[16] We thank Peter Shor for this observation.

3.2 Oracle Result

If we allow ourselves the liberty of a quantum oracle, then we can prove the following.

Theorem 14 *There exists a quantum oracle U relative to which a public-key quantum money scheme exists. (Here all parties—the bank, authenticators, and counterfeiters—have the same access to U; no party has "inside information" about U that is not available to others.)*

By "quantum oracle," we simply mean a unitary transformation U that can be applied in a black-box fashion. (We may assume controlled-U and U^{-1} are also available; this does not particularly affect our results.) Quantum oracles were first studied in a complexity-theoretic context by Aaronson and Kuperberg [4], where they were used to exhibit an oracle separation between the classes QMA and QCMA.

In the proof of Theorem 14, the oracle U does basically what one would expect. Firstly, for each possible "secret key" $s \in \{0,1\}^n$ that could be chosen by the bank, the oracle maps the state $|0\rangle|s\rangle$ to $|0\rangle|s\rangle|e_s\rangle|\psi_s\rangle$, where e_s is a classical "public key" chosen uniformly at random from $\{0,1\}^{3n}$, and $|\psi_s\rangle$ is an n-qubit pure state chosen uniformly at random under the Haar measure. (Of course, after being chosen at random, e_s and $|\psi_s\rangle$ are then fixed for all time by the oracle. Notice that with overwhelming probability, there is no pair s, s' such that $e_s = e_{s'}$. Also, here and throughout we omit ancilla qubits set to $|0\cdots0\rangle$, when they are part of the input to U.)

Secondly, for each $s \in \{0,1\}^n$, the oracle maps the state $|1\rangle|e_s\rangle|\psi_s\rangle$ to $|1\rangle|e_s\rangle|\psi_s\rangle|1\rangle$. On the other hand, it maps $|1\rangle|e_s\rangle|\phi\rangle$ to $|1\rangle|e_s\rangle|\phi\rangle|0\rangle$ if $|\phi\rangle$ is orthogonal to $|\psi_s\rangle$, and $|1\rangle|e\rangle|\phi\rangle$ to $|1\rangle|e\rangle|\phi\rangle|0\rangle$ if $e \neq e_s$ for every s.

By feeding U inputs of the form $|0\rangle|s\rangle$, the bank can prepare and distribute an unlimited number of banknotes $|e_s\rangle|\psi_s\rangle$. By feeding U inputs of the form $|1\rangle|e_s\rangle|\psi_s\rangle$, buyers and sellers can then authenticate these banknotes. Furthermore, by the optimality of Grover's algorithm [8], it is clear that any would-be counterfeiter needs $\Omega\left(2^{n/2}\right)$ queries to U to find the secret key s, even if given the public key e_s.

So the real question is this: given e_s *together with* $|\psi_s\rangle^{\otimes k}$ for some $k = \text{poly}(n)$, can a counterfeiter, by making $\text{poly}(n)$ queries to U, prepare a state that has non-negligible overlap with $|\psi_s\rangle^{\otimes k+1}$? We observe that a negative answer follows more-or-less immediately from Theorem 2, the Complexity-Theoretic No-Cloning Theorem.

4 Quantum Copy-Protection

Having summarized our results about quantum money, we now move on to the related problem of copy-protecting quantum software.

4.1 Two Schemes for Copy-Protecting Point Functions

Recall that a point function $f_s : \{0,1\}^n \rightarrow \{0,1\}$ has the form

$$f_s(x) = \begin{cases} 1 & \text{if } x = s \\ 0 & \text{otherwise} \end{cases}$$

In this section we propose two explicit schemes for quantumly copy-protecting the family $\{f_s\}_{s\in\{0,1\}^n}$ of point functions.

The first scheme, which we are grateful to Adam Smith for suggesting, uses a pseudorandom generator $g : \{0,1\}^n \rightarrow \{0,1\}^{p(n)}$, where p is some reasonably large polynomial (say n^3). Given the secret key s, the software vendor first computes $g(s)$, then reinterprets $g(s)$ as a description of a quantum circuit $U_{g(s)}$ over some universal basis of gates, which acts on m qubits for some $m \ll n$. The vendor then outputs $|\psi_s\rangle = U_{g(s)} |0\rangle^{\otimes m}$ as its quantum program for f_s. Given $|\psi_s\rangle$, the customer can efficiently compute $f_s(x)$ for any x, by measuring the state $U_{g(x)}^{-1} |\psi_s\rangle$ in the standard basis and then checking whether the outcome is $|0\rangle^{\otimes m}$.

Harrow and Low [17] have recently shown that random quantum circuits are approximate unitary 2-designs. From this it follows that if $x \neq s$, then $|\langle\psi_x|\psi_s\rangle|$ must be exponentially small with overwhelming probability, unless g is insecure against 2^m-time classical adversaries. It is also clear that s cannot be learned by a polynomial-time measurement on $|\psi_s\rangle^{\otimes k}$ for any $k = \text{poly}(n)$, unless g is insecure against polynomial-time quantum adversaries. However, the key conjecture is the following:

Conjecture 15 *Given $|\psi_s\rangle^{\otimes k}$, no polynomial-time quantum algorithm can prepare a $(k+1)^{st}$ copy of $|\psi_s\rangle$, or indeed, any other state from which f_s can be efficiently computed.*

Our second candidate scheme is based on the Hidden Subgroup Problem over the symmetric group. Given the secret key $s \in \{0,1\}^n$, the software vendor first encodes s, in some canonical way, as a permutation $\tau_s \in S_n$ such that $\tau_s^2 = e$ is the identity. The vendor then prepares a state of the form

$$|\psi_s\rangle = \frac{|\sigma_1\rangle + |\sigma_1\tau_s\rangle}{\sqrt{2}} \otimes \cdots \otimes \frac{|\sigma_k\rangle + |\sigma_k\tau_s\rangle}{\sqrt{2}},$$

where $\sigma_1, \ldots, \sigma_k$ are permutations chosen uniformly at random from S_n. Finally, the vendor distributes $|\psi_s\rangle$ as the (amplified) quantum program for f_s. Given $|\psi_s\rangle$, the customer can compute $f_s(x)$ for any x, by mapping each state $\frac{|\sigma_i\rangle+|\sigma_i\tau_s\rangle}{\sqrt{2}}$ to

$$\frac{1}{2}\left[|0\rangle\left(|\sigma_i\rangle + |\sigma_i\tau_s\rangle\right) + |1\rangle\left(|\sigma_i\tau_x\rangle + |\sigma_i\tau_s\tau_x\rangle\right)\right],$$

then Hadamarding the first qubit and measuring it in the standard basis. If $\tau_x = \tau_s$, then outcome $|0\rangle$ will be obtained with certainty, while if $\tau_x \neq \tau_s$, then outcome $|1\rangle$ will be obtained with probability $1/2$.

On the other hand, recovering τ_s given $|\psi_s\rangle$ is clearly at least as hard as the Hidden Subgroup Problem (HSP) over the symmetric group, at least for subgroups $H \leq S_n$ of order 2. Solving this special case of HSP would lead to a polynomial-time quantum algorithm for the Rigid Graph Isomorphism problem. Furthermore, Hallgren et al. [16] have shown that any quantum algorithm for recovering τ_s would require entangled measurements on $\Omega(n \log n)$ coset states; such an algorithm seems beyond present-day techniques.

Again, though, the conjecture we need is a stronger one:

Conjecture 16 *Given $|\psi_s\rangle$, no polynomial-time quantum algorithm can prepare an additional coset state $\frac{|\sigma_{k+1}\rangle+|\sigma_{k+1}\tau_s\rangle}{\sqrt{2}}$, or indeed, any other state from which f_s can be efficiently computed.*

The copying problem clearly reduces to HSP, but we do not know of a reduction in the other direction.

4.2 Oracle Result

Our main result about quantum copy-protection is the following:

Theorem 17 *There exists a quantum oracle U, relative to which any family $\mathcal{F}$ of efficiently computable functions that is not quantumly learnable can be quantumly copy-protected (with security δ, against pirates mapping k programs to $k+r$ with $(1-2\delta)r > k$).*

By a function family $\mathcal{F}$ being "quantumly learnable," we mean that given quantum oracle access to any function $f \in \mathcal{F}$, one can in polynomial time prepare a state $|\varphi_f\rangle$ from which f can then be computed in polynomial time without further help from the oracle. As discussed before, it is clear that no learnable family of functions can be copy-protected. Theorem 17 says that this is the *only* relativizing obstruction to quantum copy-protection.

In the remainder of this section, we explain the essential steps in the proof of Theorem 17, in the special case where we only need to protect against pirating algorithms that *more than double* the number of programs.

The oracle U does the following. Given as input a state of the form $|0\rangle |d_f\rangle$, where d_f is a classical description of a Boolean function $f \in \mathcal{F}$, the oracle outputs $|0\rangle |d_f\rangle |K_f\rangle |\psi_f\rangle$, where K_f is a random classical codeword specifying f, and $|\psi_f\rangle$ is a $2n$-qubit "code state" chosen uniformly at random under the Haar measure for each f. Given as input a state of the form $|1\rangle |K_f\rangle |\psi_f\rangle |x\rangle$, for some $x \in \{0,1\}^n$, the oracle outputs $|1\rangle |K_f\rangle |\psi_f\rangle |x\rangle |f(x)\rangle$. Given as input a state of the form $|1\rangle |K_f\rangle |\phi\rangle |x\rangle$, for any $|\phi\rangle$ orthogonal to $|\psi_f\rangle$, the oracle outputs $|1\rangle |K_f\rangle |\phi\rangle |x\rangle |0\rangle$.

It is clear that, for any function $f \in \mathcal{F}$, the software vendor can create and distribute states of the form $|K_f\rangle |\psi_f\rangle$, from which $f(x)$ can be efficiently computed for any input x. Furthermore, by the optimality of Grover's algorithm, a software pirate has little hope of using the oracle U to find d_f, given only K_f. As in the quantum money case, the real question is this: given the state $|K_f\rangle |\psi_f\rangle^{\otimes k}$ for some $k = \mathrm{poly}(n)$, can a quantum pirate produce $\ell > k$ programs for f using only $\mathrm{poly}(n)$ queries to U?

The Complexity-Theoretic No-Cloning Theorem suggests that the answer should be no. However, we now have to handle a new difficulty that did not arise in the money case. The new difficulty is that for certain function families $\mathcal{F}$—namely, the learnable families—we know that it *is* possible to pirate $|\psi_f\rangle$ efficiently, by using $|\psi_f\rangle$ to simulate an oracle for f, and then learning a new quantum program for f just from f's input/output behavior. Thus, our proof will need to show that learnability is the only obstacle to copy-protection. Or taking the contrapositive, we need to construct a *simulator*, which takes as input a polynomial-time algorithm for pirating $|\psi_f\rangle$, and converts it into a polynomial-time algorithm that learns a quantum program for f using only oracle access to f (and no oracle access to U).

How should the simulator work? For simplicity, let us restrict ourselves to simulators that use the pirating algorithm as a black box in constructing the learning algorithm. Intuitively, what the simulator ought to do is

(1) "mock up" its own stand-in $|K\rangle |\varphi\rangle^{\otimes k}$ for the state $|K_f\rangle |\psi_f\rangle^{\otimes k}$,

(2) run the pirating algorithm on $|K\rangle |\varphi\rangle^{\otimes k}$, using the simulator's own oracle access to f to simulate the pirating algorithm's oracle calls to U on inputs of the form $|1\rangle |K_f\rangle |\psi_f\rangle |x\rangle$, and then

(3) use the output of the pirating algorithm to get an oracle-free quantum program for f.

The idea behind step (3) is as follows: we know that at least *some* of the programs output by the pirating algorithm must not make essential use of the oracle U. For the oracle can only be usefully accessed via the "pseudorandom" state $|\varphi\rangle$—and by the Complexity-Theoretic No-Cloning Theorem, the simulator cannot have produced any additional copies of $|\varphi\rangle$.

However, already at step (1) of the above plan, we encounter a problem: in the oracle world, the states $|\psi_f\rangle$ were chosen uniformly at random under the Haar measure. In polynomial time, with no oracle access, how does one "mock up" a $2n$-qubit state $|\varphi\rangle$ such that $|\varphi\rangle^{\otimes k}$ behaves indistinguishably from k copies of a uniform random state? This is the question that we answer in the full version using Theorem 3, which gives an explicit quantum t-design for arbitrary $t = \mathrm{poly}(n)$ with the properties we need.

Let us now explain how the pieces are put together. Assume that $(1-2\delta) r > k$. Suppose we are given a pirating algorithm that takes $|\psi_f\rangle^{\otimes k}$ as input (for a given $f \in \mathcal{F}$), makes T queries to the quantum oracle U, and outputs $k+r$ possibly-entangled quantum programs $\sigma_1^U, \ldots, \sigma_{k+r}^U$ such that

$$\sum_{i=1}^{k+r} \Pr_{(f,x)\in\mathcal{D}} \left[L^U\left(\sigma_i^U, x\right) \text{ outputs } f(x) \right] \geq k + (1-\delta) r.$$

From this pirating algorithm, we want to obtain a polynomial-time algorithm that uses oracle access to f to learn an (oracle-free) quantum program for f. Here is how it works:

(1) The simulator chooses some $t = \mathrm{poly}(k, T, n)$. It then chooses a $2n$-qubit state $|\varphi\rangle$ uniformly at random from a quantum t-design, in the sense of Theorem 3. The simulator also chooses a random string K.

(2) The simulator creates a *simulated oracle* $\widetilde{U}$, which maps $|1\rangle |K\rangle |\varphi\rangle |x\rangle$ to $|1\rangle |K\rangle |\varphi\rangle |x\rangle |f(x)\rangle$ and $|1\rangle |K\rangle |\phi\rangle |x\rangle$ to $|1\rangle |K\rangle |\phi\rangle |x\rangle |0\rangle$ for every $|\phi\rangle$ orthogonal to $|\varphi\rangle$. (As a technicality, $\widetilde{U}$ does nothing on inputs of the form $|0\rangle |d_f\rangle$.[17]) Note that $\widetilde{U}$

[17] For simplicity, we are assuming it is exponentially hard for anyone but the software vendor to guess the classical description d_f for even a *single* function $f \in \mathcal{F}$—in which case, no one but the software vendor ever has anything to gain by querying U on inputs of the form $|0\rangle |d\rangle$. With slightly more work, one can remove this assumption, and even assume d_f has some standard form such as a description of a circuit for f.

can be implemented in polynomial time, using the simulator's oracle access to f.

(3) The simulator runs the pirating algorithm, except with $|K\rangle |\varphi\rangle^{\otimes k}$ in place of $|K_f\rangle |\psi_f\rangle^{\otimes k}$ for the input, and queries to $\widetilde{U}$ in place of queries to U. The simulator outputs $|\Phi\rangle$, the output of the pirating algorithm, as its candidate for an oracle-free quantum program for f.

(4) Let $\sigma_1, \ldots, \sigma_{k+r}$ be the (possibly-entangled) registers of $|\Phi\rangle$ corresponding to the $k+r$ pirated programs. Then given an input $x \in \{0,1\}^n$ and freeloading algorithm L, one computes $f(x)$ as follows. Choose $i \in [k+r]$ uniformly at random; then run $L^{\widetilde{U}}(\sigma_i, x)$ with $\widetilde{U}$ replaced by the identity transformation, and return L's output as the guess for $f(x)$.

We claim that step (4) outputs $f(x)$ with probability non-negligibly greater than $1/2$. Notice that one can amplify the success probability by repeating steps (1)-(3) $t' = \operatorname{poly}(n)$ times to obtain the state $|\Phi\rangle^{\otimes t'}$, then repeating step (4) on each copy of $|\Phi\rangle$ and outputting the majority answer.

The argument goes as follows. By Theorem 2 (the Complexity-Theoretic No-Cloning Theorem), it is impossible to use the original pirating algorithm to produce $k+1$ copies of the Haar-random state $|\psi_f\rangle$. Indeed, there cannot even be a single input $x \in \{0,1\}^n$ such that given x, one can use the output of the pirating algorithm (together with $\operatorname{poly}(n)$ additional queries to U) to prepare $k+1$ copies of $|\psi_f\rangle$. For then, by simply *guessing* x and then using amplitude amplification, one could prepare $k+1$ copies of $|\psi_f\rangle$ using only $O\left(\sqrt{2^n} \operatorname{poly}(n)\right)$ queries to U, whereas Theorem 2 implies that $\Omega\left(2^n / \operatorname{poly}(n)\right)$ queries are needed. (This is why we stipulated that $|\psi_f\rangle$ has $2n$ qubits rather than n.)

By Theorem 3, it follows that the output $|\Phi\rangle$ cannot be used to prepare $k+1$ copies of $|\varphi\rangle$ in the simulated case either—for otherwise, we would be able to distinguish the real case from the simulated one.

As a consequence, when we run $L^{\widetilde{U}}(\sigma_i, x)$ for each $i \in [k+r]$, at least r of the $k+r$ invocations must be unaffected when $\widetilde{U}$ is replaced by the identity transformation. For if an invocation *is* affected, then by the BBBV lower bound [8], it must at some point have fed $\widetilde{U}$ an input state that has $\Omega(1/\operatorname{poly}(n))$ fidelity with some state of the form $|1\rangle |K\rangle |\varphi\rangle \sum_y \alpha_y |y\rangle$. For those are the only states on which $\widetilde{U}$ behaves differently from the identity transformation. Thus, we can prepare a "clock state" of the form $\frac{1}{\sqrt{T}} \sum_{t=1}^{T} |t\rangle$, and use that state to determine how many steps t of L to apply to σ_i. We can then apply $\operatorname{poly}(n)$ steps of amplitude amplification to the joint state of the clock register and the σ_i register, searching for a marked item of the form $|t\rangle \otimes |\varphi\rangle$ for any t. This will produce, in the σ_i register, a state having $1-\varepsilon$ fidelity with $|\varphi\rangle$. But we already decided that this can be done for at most k registers.

In summary, the expression

$$\sum_{i=1}^{k+r} \Pr_{(f,x)\in\mathcal{D}} \left[L^{\widetilde{U}}(\sigma_i, x) \text{ outputs } f(x) \right]$$

can decrease by at most (say) $k + 2^{-n}$ when $\widetilde{U}$ is replaced by I. Since $(1-\delta) r > k + \delta r$, this means the sum is at least

$$\begin{aligned} k + (1-\delta) r - \left(k + 2^{-n}\right) &= (1-\delta) r - 2^{-n} \\ &\geq k + \delta r - 2^{-n}. \end{aligned}$$

So for $i \in [k+r]$ chosen randomly, $L^I(\sigma_i, x)$ outputs the correct value of $f(x)$ with probability bounded above $1/2$, as claimed.

5 Open Problems

Can we find more explicit candidate schemes for public-key quantum money—and better yet, prove such a scheme secure under a standard assumption?

Can we find candidate schemes for quantumly copy-protecting richer families of functions than just point functions? What about trapdoor inversion functions?

Can we prove a scheme for copy-protecting point functions (such as those in Section 4.1) secure under a standard assumption?

Can we improve Theorem 17 to remove the restriction on r?

Can a public-key (or at least query-secure) quantum money scheme exist, that does not require multi-qubit entanglement in the banknotes? What about a scheme for copy-protecting point functions that does not require multi-qubit entanglement in the programs?

Can we show that public-key quantum money schemes exist relative to a *classical* oracle, rather than a quantum oracle? What about nontrivial copy-protection schemes?

Is there a way to amplify a quantum program "unsplittably"—i.e., such that one cannot efficiently decompose the amplified program into two somewhat-less-amplified programs, as $\rho^{\otimes k}$ can be decomposed into $\rho^{\otimes k/2} \otimes \rho^{\otimes k/2}$?

Can we improve the parameters of the Complexity-Theoretic No-Cloning Theorem?

Can the Goldreich-Goldwasser-Micali reduction [13] from PRGs to PRFs be adapted to work in the presence of quantum adversaries?

Can we find a function family which is quantumly obfuscatable, but is not (or is not known to be) classically obfuscatable?

Can we give constructions for unclonable quantum ID cards or quantum proofs? How do these functionalities relate to money and copy-protection?

What can we say about *information-theoretically secure* quantum copy-protection, in the regime where the number of copies of the quantum program is assumed to be small?

6 Acknowledgments

I am grateful to Michele Mosca and Douglas Stebila for making the connection between the Complexity-Theoretic No-Cloning Theorem and quantum money; Adam Smith for pointing me to Wiesner's article and suggesting the copy-protection scheme based on random quantum circuits; Andris Ambainis and Luca Trevisan for discussions about t-designs; Zeev Dvir for Theorem 5; the anonymous reviewers for their comments; and Anne Broadbent, Ran Canetti, Aram Harrow, Avinatan Hassidim, Peter Shor, and Salil Vadhan for helpful conversations.

References

[1] S. Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1:1–28, 2005. quant-ph/0402095. Conference version in Proceedings of CCC'2004.

[2] S. Aaronson, E. Farhi, D. Gosset, A. Hassidim, A. Lutomirski, and P. W. Shor. Quantum money using matrix product states. In preparation, 2009.

[3] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70(052328), 2004. quant-ph/0406196.

[4] S. Aaronson and G. Kuperberg. Quantum versus classical proofs and advice. *Theory of Computing*, 3(7):129–157, 2007. Previous version in Proceedings of CCC 2007. quant-ph/0604056.

[5] A. Ambainis. Quantum lower bounds by quantum arguments. *J. Comput. Sys. Sci.*, 64:750–767, 2002. Earlier version in ACM STOC 2000. quant-ph/0002066.

[6] A. Ambainis and J. Emerson. Quantum t-designs: t-wise independence in the quantum world. In *Proc. IEEE Conference on Computational Complexity*, 2007. quant-ph/0701126.

[7] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. Vadhan, and K. Yang. On the (im)possibility of obfuscating programs. In *Proceedings of CRYPTO*, pages 1–18, 2001. ECCC TR01-057.

[8] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. quant-ph/9701001.

[9] C. H. Bennett, G. Brassard, S. Breidbart, and S. Wiesner. Quantum cryptography, or unforgeable subway tokens. pages 267–275. Plenum Press, 1982.

[10] E. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Info. Theory*, 24:384–386, 1978.

[11] M. Ettinger, P. Høyer, and E. Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Inform. Proc. Lett.*, 91(1):43–48, 2004. quant-ph/0401083.

[12] E. Farhi and S. Gutmann. Analog analogue of a digital quantum computation. *Phys. Rev. A*, 57:2403–2406, 1998. quant-ph/9612026.

[13] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1984.

[14] L. K. Grover. A different kind of quantum search. quant-ph/0503205, 2005.

[15] J. A. Halderman. Evaluating new copy-prevention techniques for audio CDs. In *Digital Rights Management Workshop*, pages 101–117, 2002.

[16] S. Hallgren, C. Moore, M. Rötteler, A. Russell, and P. Sen. Limitations of quantum coset states for graph isomorphism. In *Proc. ACM STOC*, pages 604–617, 2006.

[17] A. Harrow and R. Low. Random quantum circuits are approximate 2-designs. To appear in Comm. Math. Phys. arXiv:0802.1919, 2008.

[18] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[19] A. A. Razborov and S. Rudich. Natural proofs. *J. Comput. Sys. Sci.*, 55(1):24–35, 1997.

[20] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396.

[21] D. Stebila. *Classical Authenticated Key Exchange and Quantum Cryptography*. PhD thesis, University of Waterloo, 2009. hdl.handle.net/10012/4295. See especially Chapter 8.

[22] H. Wee. On obfuscating point functions. In *Proc. ACM STOC*, pages 523–532, 2005.

[23] S. Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, 1983. Original manuscript written circa 1970.