Lecture 8

Private-Key Quantum Money

Lecturer: Scott Aaronson *Scribe: Oded Regev*

We now start on our final application of the complexity of states and unitaries—one that, on its face, seems about as remote as possible from quantum gravity, though it will turn out not to be completely unrelated.

In the late 1960s, long before anyone even dreamed of quantum computation, Stephen Wiesner, then a graduate student, wrote a paper proposing to use quantum physics to construct unforgeable money. The paper was repeatedly rejected until it appeared 14 years later in *SIGACT News* [Wie83].

Because of the copyability of classical information, all existing approaches to electronic cash have to rely either on a trusted third party like a bank or credit card agency, or sometimes on the entire Internet (in the case of Bitcoin's block chain). To get around this, Wiesner's basic idea was to exploit what we now call the No-Cloning Theorem (from Lecture 1), stating that there's no way to duplicate an unknown quantum state. Why couldn't we use this phenomenon to create "quantum banknotes," which could be held and traded, yet which would be physically impossible to copy?

In implementing quantum money, the obvious engineering difficulty is that the users need to transport quantum states around (ideally, at room temperature and in their wallets), while keeping them coherent for more than a tiny fraction of a second! It's mainly this issue that's so far prevented the practical realization of quantum money—in contrast to *quantum key distribution*, a closely-related idea (also originating in Wiesner's 1969 paper) that's already seen some modest commercial deployment. In any case, from now on we'll ignore implementation issues and concentrate on the theory of quantum money.

For simplicity, we'll assume throughout that quantum money comes in only one denomination.

At a high level, quantum money needs to satisfy two requirements: it needs to be

(1) unclonable (obviously), but also

(2) verifiable—that is, users should be able to verify that a quantum money state presented to them is a valid state and not some junk.

8.1 The Wiesner and BBBW Schemes

Wiesner's proposal is now as follows. Each quantum banknote consists of two parts. The first part is an n-bit classical string s, called the *serial number*, chosen independently and uniformly for each note. Just like with today's classical money, the serial number serves to identify each banknote uniquely.

The second part is an n-qubit quantum state of the form

$$|0\rangle|-\rangle|1\rangle|1\rangle|+\rangle\cdots|0\rangle$$

which is chosen uniformly from among all 4^n possible n-qubit tensor products of $|0\rangle$, $|1\rangle$, $|+\rangle$, and $|-\rangle$. The bank that generates the notes stores a database of pairs, $(s, f(s))$ where $f(s)$ is a classical description of the quantum state $|\psi_s\rangle$ generated for note s (note that $f(s)$ is $2n$ bits long).

Clearly, if a user takes a note $(s, |\psi\rangle)$ back to the bank, the bank can verify the note's veracity. Indeed, the bank simply needs to check in its database that s is a valid serial number, and then if it is, measure each qubit of $|\psi_s\rangle$ in the appropriate basis ($\{|0\rangle, |1\rangle\}$ or $\{|+\rangle, |-\rangle\}$) to make sure that it's in the state corresponding to $f(s)$. A user who wants to verify a note needs to go to the bank (or send the note to the bank over a quantum channel).

What about counterfeiting? Assume the counterfeiter is given a legitimate banknote, $(s, |\psi_s\rangle)$. Can he generate from it two notes that both pass the bank's verification with high probability?

The naïve strategy to do so would simply be to measure each qubit in (say) the standard basis and then copy the result. This strategy succeeds with probability $(5/8)^n$. (Why? Qubits that happen to be in the standard basis are copied successfully; this happens with probability 1/2. The remaining qubits are damaged by the measurement, so that the result has probability 1/4 of passing both measurements.)

A less trivial strategy generates two entangled notes such that both pass verification with probability $(3/4)^n$. This turns out to be tight! Answering a question posed by Aaronson, this was proven in 2012 by Molina, Vidick, and Watrous [MVW99], and independently by Pastawski et al. [PYJ+11], using semidefinite formulations of the problem faced by the counterfeiter. (For some reason, from 1969 until 2012, people had been satisfied with handwavy, qualitative security analyses of Wiesner's scheme.)

Strictly speaking, the results of [MVW99, PYJ+11] don't constitute a full security proof, since (for example) they don't consider a counterfeiter who starts with *multiple* legitimate banknotes, $(s_1, |\psi_{s_1}\rangle), \ldots, (s_m, |\psi_{s_m}\rangle)$. However, in recent unpublished work, Aaronson gives a general security reduction that implies, as a special case, the full security of Wiesner's scheme.

The obvious disadvantage of Wiesner's scheme is the need to bring a banknote back to the bank in order to verify it. However, even if we set that aside, a second disadvantage is that in order to verify, the bank (and all its branches) need to maintain a database of all banknotes ever produced. In 1982, Bennett, Brassard, Breidbart, and Wiesner (BBBW) [BBBW82] suggested using a standard cryptographic trick to get rid of the giant database. Namely, they proposed replacing the random function $f(s)$ by a cryptographic *pseudorandom function*—or more precisely, by a function $f_k : \{0,1\}^n \longrightarrow \{0,1\}^{2n}$ chosen from a pseudorandom function family (PRF) $\{f_k\}_k$. That way, the bank need only store a single secret key k. Given a banknote $(s, |f_k(s)\rangle)$, the bank then verifies the note by first computing $f_k(s)$ and then measuring each qubit of $|f_k(s)\rangle$ in the appropriate basis, as before.

Intuitively, by doing this we shouldn't be compromising the security of the scheme, since a pseudorandom function can't be efficiently distinguished from a truly random function anyway. More formally, we argue as follows: suppose a counterfeiter could successfully attack the BBBW scheme. Then we could distinguish the pseudorandom function f_k from a truly random function, by running the attack with the given function and then checking whether the attack succeeds in counterfeiting. Given a pseudorandom function, the attack should succeed by assumption, whereas given a truly random function, the attack *can't* succeed because of the security of Wiesner's original scheme.

8.2 Formal Underpinnings

To make the above discussion rigorous, we must define what exactly we mean by quantum money, and what properties we need it to satisfy. This was done by Aaronson [Aar09] in 2009.

Definition 8.2.1 ([Aar09])**.** A *private-key quantum money scheme* consists of two polynomial-time quantum algorithms:

- Bank(k) generates a banknote $\$_k$ corresponding to a key k. ($\$_k$ can be a mixed state corresponding, e.g., to a mixture over serial numbers.)

- Ver(k, ρ) verifies that ρ is a valid note for the key k.

We say that the scheme has *completeness error* ε if valid banknotes $\$_k$ generated by Bank(k) are accepted by Ver($k, \$_k$) with probability at least $1 - \varepsilon$. We say that the scheme has *soundness error* δ if for all polynomial-time counterfeiters C outputting some number r of registers, and all polynomials q,

$$\mathbf{P}[\mathrm{Count}(k, C(\$_k^q)) > q] < \delta \ ,$$

where Count is the procedure that runs Ver($k, \cdot$) on each of the r registers output by C and counts the number of times it accepts. (So informally, we're asking that given q banknotes, C cannot generate more than q banknotes, except with probability δ.)

Let's also define a "mini-scheme" to be a scheme as above but with soundness restricted to $q = 1$ and $r = 2$, i.e., there's only one bill in circulation and the counterfeiter is asked to produce two valid bills.

Theorem 8.2.2 (Aaronson 2013, unpublished)**.** *We can build a private-key quantum money scheme with negligible completeness and soundness errors given*

- *a secure mini-scheme, and*

- *a pseudorandom function family (PRF)* $\{f_k\}_k$ *secure against quantum attacks.*

Proof Sketch. The construction is as follows. Given a mini-scheme M that generates a quantum state $\$_k$, we define a money scheme M' whose bills are of the form

$$\$'_{k,s} := |s\rangle\langle s| \otimes \$_{f_k(s)},$$

where s is chosen uniformly at random. For verification, the bank applies Ver($f_k(s)$).

We remark that when we apply this construction to the "mini-Wiesner" scheme (i.e., Wiesner's scheme without the serial number), we get the BBBW scheme. To get Wiesner's original scheme, we should replace f_k by a truly random function f.

The proof of completeness is easy, while soundness (or security) uses a hybrid argument. Intriguingly, the proof of soundness uses the PRF assumption twice: first one argues that breaking M' implies either an attack on the PRF (i.e., that it can be distinguished from a uniform function) or an attack on M' as a mini-scheme; second, one argues that an attack on M' as a mini-scheme implies either an attack on the PRF, or an attack on M (as a mini-scheme). □

What should we use as the PRF family? There are many constructions of PRF families in cryptography based on various cryptographic assumptions. There's also a general result, which says that the existence of a one-way function (the most basic of all cryptographic primitives, defined in Lecture 6, without which there's basically no crypto) is already enough to imply the existence of PRFs. This implication is proved in two steps:

$$\text{OWF} \overset{HILL}{\Longrightarrow} \text{PRG} \overset{GGM}{\Longrightarrow} \text{PRF}\ ,$$

The first step—that OWFs imply pseudorandom generators (PRGs), stretching n random bits into $n^{O(1)}$ pseudorandom bits—is due to Håstad et al. [HILL99]. The second step—that PRGs imply PRFs, stretching n random bits into 2^n pseudorandom bits—is due to Goldreich, Goldwasser, and Micali [GGM86]. Both steps are nontrivial, the first extremely so.

In our case, of course, we need a PRF family secure against *quantum* attacks. Fortunately, as discussed in Lecture 6, there are many known candidate OWFs that are quantum-secure as far as anyone knows. Also, one can show, by adapting the HILL and GGM arguments, that quantum-secure PRFs follow from quantum-secure OWFs. The first step, that of HILL, applies as-is to the quantum setting. The GGM step, however, breaks in the quantum setting, because it makes crucial use of "the set of all inputs queried by the attacker," and the fact that this set is of at most polynomial size—something that's no longer true when the attacker can query all 2^n input positions in superposition. Luckily, though, in 2012, Zhandry [Zha12] found a different proof of the GGM security reduction that does apply to quantum attackers.[1]

8.3 Security/Storage Tradeoff

To summarize, we've seen that Wiesner's scheme is unconditionally secure but requires a huge database, and that it can be transformed as above to the BBBW scheme, which is only computationally secure but which doesn't require a huge database. We also remarked that quantum-secure PRFs, as needed in the BBBW scheme, are known to follow from quantum-secure OWFs.

Given the discussion above, one might wonder whether there are quantum money schemes that are (1) information-theoretically secure, and (2) don't require lots of storage. It turns out that such schemes are impossible.

Theorem 8.3.1 (Aaronson 2013, unpublished)**.** *Any scheme in which the bank stores only n bits can be broken by an exponential-time quantum counterfeiter, using only* $\text{poly}(n)$ *legitimate money states, and $O(n)$ verification queries to the bank. (Alternatively, we can avoid the verification queries, and instead get an output that passes verification with probability* $\Omega(1/n)$*.)*

Proof. We fix the notation $k^* \in \{0,1\}^n$ for the actual secret stored by the bank. "We" (that is, the counterfeiter) are given the state $\$_{k^*}^{\otimes m}$ for $m = n^{O(1)}$, and we need to achieve a non-negligible probability of producing $m+1$ or more quantum money states that all pass verification.

A naïve strategy would be brute-force search for the key k^*, but that doesn't work for obvious reasons. Namely, we have only $\text{poly}(n)$ legitimate bills at our disposal (and the bank is not *that* naive to issue us a brand-new bill if ours doesn't pass verification!). And we can make only $\text{poly}(n)$ queries to the bank.

[1] On the other hand, for applications to quantum money, one can assume *classical* queries to the pseudorandom function f_k—since the set of banknotes that the attacker possesses is a determinate, classical set—and thus one doesn't strictly speaking need the power of Zhandry's result.

So instead, we'll recurse on the set of "potential candidate secrets S," and the crucial statement will say that at any given iteration, S shrinks by at least a constant factor.

A word of warning: we do *not* promise to find the actual value k^* (e.g. if $\$_{k^*} = \$_{k'}$ for some other k', we can never distinguish between the two). We only claim that we can come up with a "linotype" $S \ni k^*$ that's good enough for forging purposes.

Initially, we don't possess any knowledge of k^*, so we simply set $S := \{0,1\}^n$.

Now let S be our current guess. We generate (not surprisingly) the uniform mixture over the corresponding bills:

$$\sigma_S = \frac{1}{|S|} \sum_{k \in S} |\$_k\rangle\langle\$_k|$$

and submit it to the bank.

If σ_S is accepted, then S is our linotype; we can use it to print new bills.

The more interesting case is when $\mathrm{Ver}(k^*, \sigma_S)$ rejects with high probability. Then, having unlimited computational power, we can construct for ourselves the set $U \ni k^*$ of all those k for which $\mathrm{Ver}(k, \sigma_S)$ rejects with high probability. Next, we use $\mathrm{poly}(n)$ copies of the legitimate state $\$_{k^*}$ to find (we'll later explain how to do it) *some* key $k' \in U$ for which $\mathrm{Ver}(k', \$_{k^*})$ accepts with high probability. And lastly, we use this key k' to go over all individual states $\$_k$ in the mixture σ_S, and weed out all those k for which $\mathrm{Ver}(k', \$_k)$ rejects with high probability. Note that k^* survives (which is the only property we need to maintain). Meanwhile, the fact that at least a constant fraction of entries in S is removed follows from going through the definitions and from a simple application of Markov's inequality.

So all that's left is to explain how to find k' using only $\mathrm{poly}(n)$ legitimate bills.

For that, we use two more ideas: error amplification and the "Almost As Good As New Lemma" (Lemma 1.3.1 from Lecture 1). Using error amplification and $\mathrm{poly}(n)$ copies of the legitimate bill, we can assume the following without loss of generality:

- $\mathrm{Ver}(k^*, \$_{k^*})$ accepts with the overwhelming probability $1 - \exp(-Cn)$, whereas
- for any "bad" key k (defined as a key for which the original acceptance probability is less than 0.8), $\mathrm{Ver}(k, \$_{k^*})$ accepts with exponentially small probability $\exp(-Cn)$.

Now consider doing a brute-force search through U, say in lexicographic order, repairing our state $\$_{k^*}$ using the Almost As Good As New Lemma as we go along. Then the hope is that our state will still be in a usable shape by the time we reach k^*—since any "bad" keys k that we encounter along the way will damage the state by at most an exponentially small amount.

Unfortunately, the above doesn't *quite* work; there's still a technical problem that we need to deal with. Namely: what if, while searching U, we encounter keys that are "merely pretty good, but not excellent"? In that case, the danger is that applying $\mathrm{Ver}(k, \cdot)$ will damage the state substantially, but still without causing us to accept k.

One might hope to solve this by adjusting the definitions of "good" and "bad" keys: after all, if k was good enough that it caused even an amplified verifier to have a non-negligible probability of accepting, then presumably the original, unamplified acceptance probability was at least (say) 0.7, if not 0.8. The trouble is that, no matter how we set the threshold, there could be keys that fall right on the border, with a high enough acceptance probability that they damage our state, but not high enough that we reliably accept them.

To solve this problem, we use the following lemma, which builds on work by Aaronson [Aar06], with an important correction by Harrow and Montanaro [HM16]. □

Lemma 8.3.2 ("Secret Acceptor Lemma")**.** *Let $M_1, \ldots, M_N$ be known 2-outcome POVMs, and let ρ be an unknown state. Suppose we're promised that there exists an $i^* \in [N]$ such that*

$$\mathbf{P}[M_{i^*}(\rho) \text{ accepts}] \geq p.$$

Then given $\rho^{\otimes r}$, where $r = O\left(\frac{\log^4 N}{\varepsilon^2}\right)$, there's a measurement strategy to find an $i \in [N]$ such that

$$\mathbf{P}[M_i(\rho) \text{ accepts}] \geq p - \varepsilon,$$

with success probability at least $1 - \frac{1}{N}$.

Proof Sketch. The key ingredient is a result called the "Quantum OR Bound," which states the following:

- Let $M_1, \ldots, M_N$ be known 2-outcome POVMs, and let ρ be an unknown state. Then given $\rho^{\otimes r}$, where $r = O(\log N)$, there exists a measurement strategy that accepts with overwhelming probability if there exists an i such that $\mathbf{P}[M_i(\rho) \text{ accepts}] \geq 2/3$, and that rejects with overwhelming probability if $\mathbf{P}[M_i(\rho) \text{ accepts}] \leq 1/3$ for all i.

To prove the Quantum OR Bound, the obvious approach would be to apply an amplified version of M_i to $\rho^{\otimes r}$, for each i in succession. The trouble is that there might be i's for which the amplified measurement accepts with probability that's neither very close to 0 nor very close to 1—in which case, the measurement could damage the state (compromising it for future measurements), yet without leading to an overwhelming probability of acceptance. Intuitively, the solution should involve arguing that, if this happens, then at any rate we have $\mathbf{P}[M_i(\rho) \text{ accepts}] > 1/3$, and therefore it's safe to accept. Or in other words: if we accept on the i^{th} measurement, then either $M_i(\rho)$ accepts with high probability, or else our state was damaged by previous measurements—but in the latter case, some of those previous measurements (we don't know which ones) must have accepted ρ with a suitably high probability.

The trouble is that, if $M_1, \ldots, M_N$ were chosen in a diabolical way, then applying them in order could slowly "drag" the state far away from the initial state $\rho^{\otimes r}$, without any of the M_i's accepting with high probability (even though the later measurements would have accepted $\rho^{\otimes r}$ itself). In 2006, Aaronson [Aar06] encountered this problem in the context of proving the containment $\mathsf{QMA/qpoly} \subseteq \mathsf{PSPACE/poly}$, which we mentioned as Theorem 5.6.6 in Lecture 5. In [Aar06], Aaronson claimed that one could get around the problem by simply choosing the M_i's in *random* order, rather than in the predetermined (and possibly adversarial) order $M_1, \ldots, M_N$.

Unfortunately, very recently Harrow and Montanaro [HM16] found a bug in Aaronson's proof of this claim. As they point out, it remains plausible that Aaronson's random measurement strategy is sound—but if so, then a new proof is needed (something Harrow and Montanaro leave as a fascinating open problem).

In the meantime, though, Harrow and Montanaro recover the Quantum OR Bound itself, by giving a different measurement strategy for which they *are* able to prove soundness. We refer the reader to their paper for details, but basically, their strategy works by first placing a control qubit in the $|+\rangle$ state, and then applying the amplified measurements to $\rho^{\otimes r}$ conditioned on the control qubit being $|1\rangle$, and periodically measuring the control qubit in the $\{|+\rangle, |-\rangle\}$ basis to check whether the measurements have damaged the state by much. We accept if *either* some measurement accepts, or the state has been found to be damaged.

Now, once we have the Quantum OR Bound, the proof of Lemma 8.3.2 follows by a simple application of binary search (though this increases the number of copies of ρ needed by a further $\operatorname{polylog}(N)$ factor). More precisely: assume for simplicity that N is a power of 2. Then we first apply the Quantum OR Bound, with suitably-chosen probability cutoffs, to the set of measurements $M_1, \ldots, M_{N/2}$. We then apply the OR Bound separately (i.e., with fresh copies of ρ) to the set $M_{N/2+1}, \ldots, M_N$. With overwhelming probability, at least one of these applications will accept, in which case we learn a subset of half the M_i's containing a measurement that accepts ρ with high probability—without loss of generality, suppose it's $M_1, \ldots, M_{N/2}$. We then recurse, dividing $M_1, \ldots, M_{N/2}$ into two subsets of size $N/4$, and so on for $\log N$ iterations until we've isolated an i^* such that M_{i^*} accepts with high probability. At each iteration we use fresh copies of ρ.

There's just one remaining technical issue: namely, at each iteration, we start with a promise that the region we're searching contains a measurement M_i that accepts ρ with probability at least p. We then reduce to a smaller region, which contains a measurement that accepts ρ with probability at least $p - \delta$, for some fudge factor $\delta > 0$. Thus, if δ was constant, then we could only continue for a constant number of iterations. The solution is simply to set $\delta := \frac{\varepsilon}{\log N}$, where ε is the final error bound in the lemma statement. This blows up the number of required copies of ρ, but only (one can check) to

$$O\left(\frac{\log^4 N}{\varepsilon^2}\right).$$

□

8.3.1 Open Problems

As usual, we can now ask: is there any collapse of complexity classes, such as $\mathsf{P} = \mathsf{PSPACE}$, that would make the counterfeiting algorithm of Theorem 8.3.1 *computationally* efficient, rather than merely efficient in the number of legitimate bills and queries to the bank? We've seen that, if quantum-secure OWFs exist, then the algorithm can't be made efficient—but just like in the case of firewalls and the HH Decoding Task, we don't have a converse to that result, showing that if counterfeiting is hard then some "standard" cryptographic assumption must fail.

Meanwhile, the proof of Lemma 8.3.2 raises a fascinating open problem about quantum state tomography, and we can't resist a brief digression about it.

Let ρ be an unknown state, and let $E_1, \ldots, E_N$ be some list of known two-outcome POVMs. Suppose we wanted, not merely to find an i^* for which $\operatorname{Tr}(E_{i^*}\rho)$ is large, but to estimate *all* $\operatorname{Tr}(E_i\rho)$'s to within some additive error $\varepsilon > 0$, with success probability at least (say) 0.99. What resources are needed for this?

The task is trivial, of course, if we have $O(N \log N)$ fresh copies of ρ: in that case, we just apply every E_i to its own $\log N$ copies.

But suppose instead that we're given only $\operatorname{polylog} N$ copies of ρ. Even in that case, it's not hard to see that the task is achievable when the $\operatorname{Tr}(E_i\rho)$'s are promised to be bounded away from $1/2$ (e.g., either $\operatorname{Tr}(E_i\rho) > 2/3$ or $\operatorname{Tr}(E_i\rho) < 1/3$ for all i). For then we can simply apply the amplified E_i's in succession, and use the Almost As Good As New Lemma to upper-bound the damage to $\rho^{\operatorname{polylog} N}$.

We now raise the following open question:

Question 8.3.3 ("The Batch Tomography Problem")**.** Is the above estimation task achievable with only polylog N copies of ρ, and *without* a promise on the probabilities?

Note that this problem doesn't specify any dependence on the Hilbert space dimension d, since it's conceivable that a measurement procedure exists with no dimension dependence whatsoever (as happened, for example, for the Quantum OR Bound and Lemma 8.3.2). But a nontrivial dimension-dependent result (i.e., one that didn't simply depend on full tomography of ρ) would also be of interest.

8.4 Interactive Attacks

To return to quantum money, we've seen that there's an inherent tradeoff between Wiesner-style and BBBW-style private-key quantum money schemes. Whichever we choose, though, there's an immediate further problem. Namely, a counterfeiter might be able to use repeated queries to the bank—a so-called *interactive attack*—to learn the quantum state of a bill.

Indeed, this could be seen as a security flaw in our original definition of private-key quantum money. Namely, we never specified what happens *after* the customer (or, we might as well assume, a counterfeiter) submits a bill for verification. Does the counterfeiter get back the damaged (that is, post-measured) bill if it's accepted, or does the bank issue a brand-new bill? And what happens if the bill doesn't pass verification? The possibilities here vary from the bank being exceedingly naïve, and giving the bill back to the customer even in that case, to being exceedingly strict and calling the police immediately. As we'll see in Lecture 9, we can get schemes that are provably secure even with a maximally naïve, bill-returning bank, but proving this requires work.

As for the original Wiesner and BBBW schemes, it turns out that both can be fully broken by an interactive attack. To see this, let's first consider the "naïve bank" scenario: the bank returns the bill even if it didn't pass the verification. In that scenario, a simple attack on Wiesner's scheme was independently observed by Aaronson [Aar09] and by Lutomirski [Lut10].

The attack works as follows: the counterfeiter starts with a single legitimate bill,

$$|\$\rangle = |\theta_1\rangle|\theta_2\rangle \cdots |\theta_n\rangle.$$

The counterfeiter then repeatedly submits this bill to the bank for verification, swapping out the first qubit $|\theta_1\rangle$ for $|0\rangle$, $|1\rangle$, $|+\rangle$, and $|-\rangle$ in sequence. By observing which choice of $|\theta_1\rangle$ causes the bank to accept, after $O(\log n)$ repetitions, the counterfeiter knows the correct value of $|\theta_1\rangle$ with only (say) $1/n^2$ probability of error. Furthermore, since the bank's measurements of $|\theta_2\rangle, \ldots, |\theta_n\rangle$ are in the correct bases, none of these qubits are damaged at all by the verification process. Next the counterfeiter repeatedly submits $|\$\rangle$ to the bank, with the now-known correct value of $|\theta_1\rangle$, but substituting $|0\rangle$, $|1\rangle$, $|+\rangle$, and $|-\rangle$ in sequence for $|\theta_2\rangle$, and so on until all n qubits have been learned.

8.4.1 The Elitzur-Vaidman Bomb Attack

Of course, the above counterfeiting strategy fails if the bank adopts the simple countermeasure of calling the police whenever a bill fails verification (or perhaps, whenever too many bad bills get submitted with the same serial number)!

In 2014, however, Nagaj and Sattath [NS14] cleverly adapted the attack on Wiesner's scheme, so that it works even if the bank calls the police after a single failed verification. Their construction is based on the *Elitzur-Vaidman bomb* [EV93], a quantum effect that's related to Grover's algorithm but extremely counterintuitive and interesting in its own right—so let's now have a digression about the Elitzur-Vaidman bomb.

Suppose someone has a package P that might or might not be a bomb. You can send to the person a bit $b \in \{0,1\}$. You always get the bit back. But if you send $b = 1$ ("I believe you have a bomb in your package") and P is a bomb, then it explodes, killing everyone. The task is to learn whether or not P is a bomb without actually setting it off.

This is clearly impossible classically. For the only way to gain *any* information about P is eventually to dare the question $b = 1$, with all its consequences.

Not so quantumly! We can model the package P as an unknown 1-qubit operator, on a qubit $|b\rangle$ that you send. If there's no bomb, then P simply acts as the identity on $|b\rangle$. If there *is* a bomb, then P measures $|b\rangle$ in the $\{|0\rangle, |1\rangle\}$ basis. If P observes $|0\rangle$, then it returns the qubit to you, while if P observes $|1\rangle$, then it sets off the bomb.

Now let

$$R_\varepsilon = \begin{pmatrix} \cos\varepsilon & -\sin\varepsilon \\ \sin\varepsilon & \cos\varepsilon \end{pmatrix}$$

be a unitary that rotates $|b\rangle$ by a small angle $\varepsilon > 0$.

Then you simply need to use the following algorithm:

- **Initialize** $|b\rangle := |0\rangle$
- **Repeat** $\frac{\pi}{2\varepsilon}$ **times:**
  - Apply R_ε to $|b\rangle$
  - Send $|b\rangle$ to P
- **Measure** $|b\rangle$ **in the** $\{|0\rangle, |1\rangle\}$ **basis**
- **If the result is** $|0\rangle$ **then output "bomb"; if** $|1\rangle$ **then output "no bomb"**

Let's analyze this algorithm. If there's no bomb, then the invocations of P do nothing, so $|b\rangle$ simply gets rotated by an ε angle $\frac{\pi}{2\varepsilon}$ times, evolving from $|0\rangle$ to $|1\rangle$. If, on the other hand, there *is* a bomb, then you repeatedly submit $|b\rangle = \cos\varepsilon|0\rangle + \sin\varepsilon|1\rangle$ to P, whereupon P measures $|b\rangle$ in the standard basis, observing $|1\rangle$ (and hence setting off the bomb) with probability $\sin^2\varepsilon \approx \varepsilon^2$. Thus, across all $\frac{\pi}{2\varepsilon}$ invocations, the bomb gets set off with total probability of order $\varepsilon^2/\varepsilon = \varepsilon$, which of course can be made arbitrarily small by choosing ε small enough. Furthermore, assuming the bomb is *not* set off, the qubit $|b\rangle$ ends up in the state $|0\rangle$, and hence measuring $|b\rangle$ reveals (non-destructively!) that the bomb was present. Of course, what made the algorithm work was the ability of quantum measurements to convert an ε amplitude into an ε^2 probability.

Nagaj and Sattath [NS14] realized that one can use a similar effect to attack Wiesner's scheme even in the case of a suspicious bank. Recall the attack of Aaronson [Aar09] and Lutomirski [Lut10], which learned a Wiesner banknote one qubit $|\theta_i\rangle$ at a time. We want to adapt this attack so that the counterfeiter still learns $|\theta_i\rangle$, with high probability, without ever submitting a banknote that fails verification. In particular, that means: without ever submitting a banknote whose i^{th} qubit is more than some small ε away from $|\theta_i\rangle$ itself. (Just like in the earlier attack, we can assume that the qubits $|\theta_j\rangle$ for $j \neq i$ are all in the correct states, as we vary $|\theta_i\rangle$ in an attempt to learn a classical description of that qubit.)

The procedure by Nagaj et al., like the previous procedure, works qubit by qubit: this is okay since we get our bill back if it has been validated, and if we fail then we have more pressing issues to worry about than learning the remaining qubits. So let's assume $n = 1$: that is, we want to learn a single qubit $|\theta\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$.

The algorithm is as follows: we (the counterfeiter) prepare a control qubit in the state $|0\rangle$. We then repeat the following, $\frac{\pi}{2\varepsilon}$ times:

- Apply a 1-qubit unitary that rotates the control qubit by ε counterclockwise.
- Apply a CNOT gate from the control qubit to the money qubit $|\theta\rangle$.
- Send the money qubit to the bank to be measured.

To see why this algorithm works, there are four cases to consider:

- If $|\theta\rangle = |+\rangle$, then applying a CNOT to $|\theta\rangle$ does nothing. So the control qubit just rotates by an ε angle $\frac{\pi}{2\varepsilon}$ times, going from $|0\rangle$ to $|1\rangle$. In no case does the bank's measurement yield the outcome $|-\rangle$.
- If $|\theta\rangle = |0\rangle$, then applying a CNOT to $|\theta\rangle$ produces the state

  $$\cos(\varepsilon)|00\rangle + \sin(\varepsilon)|11\rangle.$$

  The bank's measurement then yields the outcome $|0\rangle$ with probability $\cos^2(\varepsilon) \approx 1 - \varepsilon^2$. Assuming $|0\rangle$ is observed, the state collapses back down to $|00\rangle$, and the cycle repeats. By the union bound, the probability that the bank *ever* observes the outcome $|1\rangle$ is at most

  $$\frac{\pi}{2\varepsilon} \sin^2(\varepsilon) = O(\varepsilon).$$

- If $|\theta\rangle = |1\rangle$, then the situation is completely analogous to the case $|\theta\rangle = |0\rangle$.
- If $|\theta\rangle = |-\rangle$, then applying a CNOT to $|\alpha\rangle$ produces the state

  $$(\cos(\varepsilon)|0\rangle - \sin(\varepsilon)|1\rangle) \otimes |-\rangle.$$

  So at the next iteration, the control qubit gets rotated back to $|0\rangle$ (and nothing happens); then it gets rotated back to $\cos(\varepsilon)|0\rangle - \sin(\varepsilon)|1\rangle$, and so on, cycling through those two states. In no case does the bank's measurement yield the outcome $|+\rangle$.

To summarize, if $|\theta\rangle = |+\rangle$ then the control qubit gets rotated to $|1\rangle$, while if $|\theta\rangle$ is any of the other three possibilities, then the control qubit remains in the state $|0\rangle$ (or close to $|0\rangle$) with overwhelming probability. So at the end, measuring the control qubit in the $\{|0\rangle, |1\rangle\}$ basis lets us distinguish $|\theta\rangle = |+\rangle$ from the other three cases. By symmetry, we can repeat a similar procedure for the other three states, and thereby learn $|\theta\rangle$, using $O(1/\varepsilon)$ trial verifications, with at most $O(\varepsilon)$ probability of getting caught. So by repeating the algorithm for each θ_i, we learn the entire bill with $O(n/\varepsilon)$ trial verifications and at most $O(n\varepsilon)$ probability of getting caught.

While we explained these attacks for Wiesner's scheme, the same attacks work with only minor modifications for the BBBW scheme—or indeed, for *any* scheme where the banknotes consist of unentangled qubits, and a banknote is verified by projectively measuring each qubit separately.

However, having explained these attacks, we should also mention an extremely simple fix for them. The fix is just that, rather than *ever* returning a bill to a customer after verification, the bank destroys the bill, and hands back a new bill (with a new serial number) of the same denomination! If this is done, then the original security definition that we gave for private-key quantum money really does make sense.

8.5 Quantum Money and Black Holes

We end this lecture with the following observation. As we saw above, quantum-secure OWFs are enough to obtain a private-key quantum money scheme with small secret. It can be shown that if, moreover, the OWFs are injective, they give us something additional: namely, an "injective" private-key quantum money scheme, one where distinct serial numbers $k \neq k'$ map to nearly-orthogonal banknotes, $|\langle \$_k | \$_{k'} \rangle| < \varepsilon$. Now, we saw in Lecture 6 that injective, quantum-secure OWFs are also enough to imply that the HH Decoding Task is hard.

But is there any *direct* relationship between private-key quantum money and the HH Decoding Task, not mediated by OWFs? It turns out that there is. To see this, in Aaronson's first hardness result for the HH Decoding Task (i.e., the first construction in Theorem 6.5.3), we simply need to replace the injective OWF output $|f(x)\rangle$ by the output $|\$_k\rangle$ of an injective private-key quantum money scheme, yielding the state

$$|\psi\rangle_{RBH} = \frac{1}{\sqrt{2^{n+1}}} \sum_{k \in \{0,1\}^n} \left(|k\, 0^{p(n)-n}, 0\rangle_R |0\rangle_B + |\$_k, 1\rangle_R |1\rangle) |k\rangle_H \right).$$

Then the same argument as in the proof of Theorem 6.5.3 shows that the ability to decode entanglement between R and B in this state would imply the ability to pass from $|\$_k\rangle$ to $|k\rangle$—and *therefore*, the ability to break the private-key quantum money scheme. So we get the following consequence, which perhaps exemplifies the strange connections in this course better than any other single statement:

Theorem 8.5.1. *Suppose there exists a secure, private-key, injective quantum money scheme with small keys. Then the HH Decoding Task is hard.*

(Or in words: "the ability to decode the Hawking radiation from a black hole implies the ability to counterfeit quantum money.")

Since injective OWFs imply injective private-key quantum money with small keys, but the reverse isn't known, Theorem 8.5.1 is stronger than Theorem 6.5.3, in the sense that it bases the hardness of HH decoding on a weaker assumption.

Lecture 9

Public-Key Quantum Money

Lecturer: Scott Aaronson *Scribe: Alexander Razborov*

As the last lecture made clear, one interesting question is *whether there's any private-key quantum money scheme where the banknote can be returned to the customer after verification, yet that's still secure against interactive attacks.* We'll return to that question later in this lecture.

But as long as we're exploring, we might as well be even more ambitious, and ask for what Aaronson [Aar09] called a *public-key* quantum money scheme. By analogy to public-key cryptography, this is a quantum money scheme where absolutely anyone can verify a banknote, not just the bank that printed it—i.e., where verifying a banknote no longer requires taking it to the bank at all. And yet, despite openly publishing a verification procedure, the bank still somehow ensures that no one can use that procedure to *copy* a bill efficiently.

Following Aaronson [Aar09], and Aaronson and Christiano [AC12], we now give a more precise definition of public-key quantum money, in the style of Definition 8.2.1.

Definition 9.0.2. A *public-key quantum money scheme* consists of three polynomial-time algorithms:

- KeyGen(0^n), which outputs a pair $k = (k_{\text{pub}}, k_{\text{private}})$ (this is probabilistic, of course);
- Bank(k) $= \$_k$ (exactly as in Definition 8.2.1);
- Ver(k_{pub}, ρ) (same as in Definition 8.2.1, except that now the verifier has access only to the public part of the key).

Completeness and soundness errors are defined exactly as before, replacing k with k_{pub} in appropriate places. Completeness is defined in the worst case (with respect to the choice of k), whereas while defining the soundness, we average over the internal randomness of KeyGen.

9.1 Public-Key Mini-Schemes

Just like we had a notion of private-key mini-schemes, we have a corresponding notion of public-key mini-schemes.

Definition 9.1.1. A *public-key mini-scheme* consists of two polynomial-time algorithms:

- Bank(0^n), which probabilistically outputs a pair $\$ = (s, \rho_s)$ (where s is a classical serial number);
- Ver($\$$), which accepts or rejects a claimed banknote.

The scheme has completeness error ϵ if

$$\mathbf{P}[\mathrm{Ver}(\$) \text{ accepts}] \geq 1 - \epsilon.$$

It has soundness error δ if for all polynomial-time counterfeiters mapping $\$ = (s, \rho_s)$ into two possibly entangled states σ_1 and σ_2, we have

$$\mathbf{P}[\mathrm{Ver}(s, \sigma_1),\ \mathrm{Ver}(s, \sigma_2) \text{ both accept}] < \delta.$$

Just like in the private-key case, to go from a mini-scheme to a full money scheme, one needs an additional ingredient. In this case, the ingredient turns out to be a conventional *digital signature scheme* secure against quantum attacks. We won't give a rigorous definition, but roughly: a digital signature scheme is a way of signing messages, using a private key k_{private}, in such a way that

- the signatures can be verified efficiently using a public key k_{public}, and yet
- agents who only know k_{public} and not k_{private} can't efficiently generate *any* yet-unseen message together with a valid signature for that message, even after having seen valid signatures for any polynomial number of messages of their choice.

Now, given a public-key mini-scheme plus a quantum-secure signature scheme, the construction that produces a full public-key money scheme was first proposed by Lutomirski et al. in 2010 [LAF$^+$10], with the security analysis given by Aaronson and Christiano in 2012 [AC12].

The idea is simply to turn a mini-scheme banknote, (s, ρ_s), into a full money scheme banknote by digitally signing the serial number s, and then tacking on the signature as an additional part of the serial number:

$$\$_k = (s, \mathrm{sign}(k, s), \rho_s).$$

To check the resulting banknote, one first checks the digital signature $\mathrm{sign}(k, s)$ to make sure the note $\$_k$ really came from the bank rather than an impostor. One then checks (s, ρ_s) using the verification procedure for the underlying mini-scheme. Intuitively, then, to print more money, a counterfeiter *either* needs to produce new bills with new serial numbers and new signatures, thereby breaking the signature scheme, *or else* produce new bills with already-seen serial numbers, thereby breaking the mini-scheme. This intuition can be made rigorous:

Theorem 9.1.2 (Aaronson-Christiano [AC12]). *If there exists a counterfeiter against the public-key quantum money scheme above, then there also exists a counterfeiter against either the mini-scheme or the signature scheme.*

The proof of Theorem 9.1.2 is a relatively standard cryptographic hybrid argument, and is therefore omitted.

Now, we have many good candidates for quantum-secure signature schemes (indeed, it follows from a result of Rompel [Rom90] that such schemes can be constructed from any quantum-secure one-way function). Thus, the problem of public-key quantum money can be reduced to the problem of constructing a secure public-key mini-scheme.

9.2 Candidates

How can we do that? Many concrete mini-schemes were proposed over the past decade, but alas, the majority of them have since been broken, and those that remain seem extremely hard to analyze. To give three examples:

- Aaronson [Aar09], in 2009, proposed a scheme based on stabilizer states. This scheme was subsequently broken by Lutomirski et al. [LAF+10], by using a nontrivial algorithm for finding planted cliques in random graphs.
- There were several schemes based on random instances of the QMA-complete Local Hamiltonians problem (see Lecture 4). All these schemes were then broken by Farhi et al. [FGH+10] in 2010, using a new technique that they called *single-copy tomography.* We'll have more to say about single-copy tomography later.
- In 2012, Farhi et al. [FGH+12] proposed another approach based on knot theory. In this approach, a quantum banknote is a superposition over oriented link diagrams sharing a given value v of the Alexander polynomial (a certain efficiently-computable knot invariant), with the serial number encoding v. As of 2016, this scheme remains unbroken, but it remains unclear how to say anything else about its security (for example, by giving a reduction).

9.3 Public-Key Quantum Money Relative to Oracles

Given the apparent difficulty of constructing a secure public-key mini-scheme, perhaps we should start with an easier question: *is there even an oracle A such that, if the bank, customers, and counterfeiters all have access to A, then public-key quantum money is possible?* If (as we'd hope) the answer turns out to be yes, *then* we can work on replacing A by an explicit function.

In 2009, Aaronson [Aar09] showed that there's at least a *quantum* oracle (as defined in Lecture 5) relative to which public-key quantum money is possible.

Theorem 9.3.1 (Aaronson 2009 [Aar09])**.** *There exists a quantum oracle U relative to which a public-key mini-scheme exists.*

Proof Sketch. We start by fixing a mapping from n-bit serial numbers s to n-qubit pure states $|\psi_s\rangle$, where each $|\psi_s\rangle$ is chosen independently from the Haar measure. Then every valid banknote will have the form $(s, |\psi_s\rangle)$, and verifying a claimed a banknote (s, ρ) will consist of projecting ρ onto the subspace spanned by $|\psi_s\rangle$.

Now, the quantum oracle U is simply a reflection about the subspace spanned by valid banknotes: that is,

$$U = I - 2 \sum_{s \in \{0,1\}^n} |s\rangle\langle s| \otimes |\psi_s\rangle\langle\psi_s|.$$

Given this U, verifying a banknote is trivial: we just delegate the verification to the oracle. (Technically, U also includes a hidden component, accessible only to someone who knows the bank's secret key k, which maps $|s\rangle|0 \cdots 0\rangle$ to $|s\rangle|\psi_s\rangle$, and which the bank uses to print new bills.)

How do we prove this scheme secure? On the one hand, it follows immediately from the No-Cloning Theorem that a counterfeiter who had only $(s, |\psi_s\rangle)$, and who lacked access to U, would not be able to print additional bills. On the other hand, if a counterfeiter *only* had access to U, and lacked any legitimate banknote, it's not hard to see that the counterfeiter could produce a valid banknote $(s, |\psi_s\rangle)$ using $O(2^{n/2})$ queries to U—but, on the other hand, that $\Omega(2^{n/2})$ queries are also necessary, by the optimality of Grover's algorithm (see Lecture 5).

The interesting part is to show that, even if the counterfeiter starts with a valid banknote $(s, |\psi_s\rangle)$, the counterfeiter still needs $\Omega(2^{n/2})$ queries to U to produce a *second* valid banknote—i.e., to produce any state that has nontrivial overlap with $|\psi_s\rangle \otimes |\psi_s\rangle$. This is a result that Aaronson [Aar09] calls the *Complexity-Theoretic No-Cloning Theorem* (since it combines the No-Cloning Theorem with query complexity):

- Given a Haar-random n-qubit pure state $|\psi\rangle$, together with a quantum oracle $U_\psi = I - 2|\psi\rangle\langle\psi|$ that reflects about $|\psi\rangle$, one still needs $\Omega(2^{n/2})$ queries to U_ψ to produce any state that has $\Omega(1)$ fidelity with $|\psi\rangle \otimes |\psi\rangle$.

We won't prove the Complexity-Theoretic No-Cloning Theorem here, but will just sketch the main idea. One considers all pairs of n-qubit pure states $|\psi\rangle, |\phi\rangle$ such that (say) $|\langle\psi|\phi\rangle| = 1/2$. One then notices that a successful cloning procedure would map these pairs to $|\psi\rangle^{\otimes 2}, |\phi\rangle^{\otimes 2}$ that satisfy

$$\left|\langle\psi|^{\otimes 2}|\phi\rangle^{\otimes 2}\right| = |\langle\psi|\phi\rangle|^2 = \frac{1}{4}.$$

In other words, for all these pairs, the procedure needs to *decrease the inner product* by $1/4 = \Omega(1)$. Given any *specific* pair $|\psi\rangle, |\phi\rangle$, it's not hard to design a single query, to the oracles U_ψ, U_ϕ respectively, that would decrease the inner product by $\Omega(1)$. However, by using Ambainis's adversary method [Amb02], one can show that no query can do this for *most* $|\psi\rangle, |\phi\rangle$ pairs, if the pairs are chosen Haar-randomly subject to $|\langle\psi|\phi\rangle| = 1/2$. Rather, any query decreases the *expected* squared fidelity,

$$\mathbb{E}_{|\psi\rangle,|\phi\rangle}\left[|\psi|\phi|^2\right],$$

by at most $O(2^{-n/2})$, where the expectation is taken over all such pairs. This then implies that, to decrease the squared fidelity by $\Omega(1)$ for all or even most pairs, we need to make $\Omega(2^{n/2})$ queries to U—i.e., just as many queries as if we were doing pure Grover search for $|\psi\rangle$, rather than starting with one copy of $|\psi\rangle$ and then merely needing to make a second one. □

9.4 The Hidden Subspace Scheme

Because it delegates all the work to the oracle U, Theorem 9.3.1 gives us little insight about how to construct secure public-key mini-schemes in the "real," unrelativized world. As the next step toward that goal, one naturally wonders whether public-key quantum money can be shown possible relative to a *classical* oracle A (that is, a Boolean function accessed in quantum superposition). This question was answered by the work of Aaronson and Christiano [AC12] in 2012.

Theorem 9.4.1 (Aaronson-Christiano 2012 [AC12])**.** *There exists a classical oracle relative to which public-key quantum money is possible.*

Proof Sketch. The construction is based on "hidden subspaces." In particular, let $S \leq \mathbb{F}_2^n$ be a subspace of the vector space $\mathbb{F}_2^n$, which is chosen uniformly at random subject to $\dim(S) = n/2$.

(Thus, S has exactly $2^{n/2}$ elements.) Let

$$S^{\perp} = \{x \mid x \cdot s \equiv 0 \,(\text{mod}2) \,\forall s \in S\}$$

be S's dual subspace, which also has $\dim(S^{\perp}) = n/2$ and $|S| = 2^{n/2}$.

Then in our public-key mini-scheme, each banknote will have the form $(d_S, |S\rangle)$, where

- d_S is an obfuscated classical description of the subspace S and its dual subspace (which we take to be the serial number), and
- $$|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{x \in S} |x\rangle$$

  is a uniform superposition over S.

When we feed it the serial number d_S as a "password," the classical oracle A gives us access to the characteristic functions $\chi_S, \chi_{S^{\perp}}$ of S and $S^{\perp}$ respectively. Using these, we can easily realize a projection onto $|S\rangle$. To do so, we just do the following to our money state (which is supposed to be $|S\rangle$):

- Apply χ_S, to check that the state has support only on S elements.
- Hadamard all n qubits, to map $|S\rangle$ to $|S^{\perp}\rangle$.
- Apply $\chi_{S^{\perp}}$, to check that the transformed state has support only on $S^{\perp}$ elements.
- Hadamard all n qubits again, to return the state to $|S\rangle$.

As a technical point, the oracle also contains a one-way mapping $(s_1, \ldots, s_{n/2}) \to d_S$, which lets the bank find the obfuscated serial number d_S for a given bill $|S\rangle$ that it prepares, given a basis for S, but *doesn't* let a user learn the basis for S given d_S, which would enable counterfeiting.

The key claim is this: *any algorithm that breaks this mini-scheme—that is, maps $|d_S\rangle|S\rangle$ to $|d_S\rangle|S\rangle^{\otimes 2}$—must make $\Omega(2^{n/4})$ queries to the oracles χ_S and $\chi_{S^{\perp}}$.*

This claim, whose proof we omit, is a somewhat stronger version of the Complexity-Theoretic No-Cloning Theorem from the proof of Theorem 9.3.1. It's stronger because, in its quest to prepare $|S\rangle^{\otimes 2}$, the counterfeiting algorithm now has access not only to a copy of $|S\rangle$ and a unitary transformation that implements $I - 2|S\rangle\langle S|$, but also the oracles χ_S and $\chi_{S^{\perp}}$. Nevertheless, again by using Ambainis's quantum adversary method, it's possible to show that there's no faster way to prepare a new copy of $|S\rangle$, than simply by doing Grover search on χ_S or $\chi_{S^{\perp}}$ to search for a basis for S or $S^{\perp}$—i.e., the same approach one would use if one were preparing $|S\rangle$ "de novo," with no copy of $|S\rangle$ to begin with. And we know that Grover search for a nonzero element of S or $S^{\perp}$ requires $\Omega(2^{n/4})$ queries to S or $S^{\perp}$ respectively.

There are two further ideas needed to complete the security proof. First, the argument based on the Complexity-Theoretic No-Cloning Theorem implies that $\Omega(2^{n/4})$ oracle queries are needed to prepare a state that has *very* high fidelity with $|S\rangle^{\otimes 2}$. But what about preparing a state that merely has non-negligible (1/poly) fidelity with $|S\rangle^{\otimes 2}$, which of course would already be enough to break the mini-scheme, according to our security definition? To rule that out, Aaronson and Christiano give a way to "amplify" weak counterfeiters into strong ones, with only a polynomial overhead in query complexity, in any money scheme where the verification consists of a projective measurement onto a 1-dimensional subspace. This, in turn, relies on recent variants of amplitude amplification that converge monotonically toward the target state (in this case, $|S\rangle^{\otimes 2}$), rather than "overshooting" it if too many queries are made (see, for example, Tulsi, Grover, and Patel [TGP06]).

The second issue is that the Complexity-Theoretic No-Cloning Theorem only rules out a counterfeiter that maps $|S\rangle$ to $|S\rangle^{\otimes 2}$ for *all* subspaces S with $\dim(S) = n/2$, or at least the vast majority of them. But again, what about a counterfeiter that only succeeds on a 1/poly fraction of subspaces? To deal with this, Aaronson and Christiano show that the hidden subspace scheme has a "random self-reducibility" property: any counterfeiter C that works on a non-negligible fraction of S's can be converted into a counterfeiter that works on *any* S. This is proven by giving an explicit random self-reduction: starting with a state $|S\rangle$, one can apply a linear transformation that maps $|S\rangle$ to some random *other* subspace state $|T\rangle$, while also replacing the oracles χ_S or $\chi_{S^\perp}$ by "simulated oracles" χ_T or $\chi_{T^\perp}$, which use χ_S and $\chi_{S^\perp}$ to recognize elements of T and $T^\perp$ respectively. □

Now, once that we have a public-key quantum money scheme that's secure relative to a classical oracle, Aaronson and Christiano [AC12] observed that we get something else essentially for free: namely, a *private*-key quantum money scheme where bills are returned after verification, yet that's secure against interactive attack. This solves one of the main problems left open by Lecture 8. Let's now explain the connection.

Theorem 9.4.2 (Aaronson-Christiano 2012 [AC12])**.** *There exists a private-key quantum money scheme secure against interactive attack (with no computational assumptions needed, though with the bank maintaining a huge database).*

Proof. Each banknote has the form $|z\rangle|S_z\rangle$, where $|z\rangle$ is a classical serial number, and $|S_z\rangle$ is an equal superposition over S_z, a subspace of $\mathbb{F}_2^n$ of dimension $n/2$. The bank, in a giant database, stores the serial number z of every bill in circulation, as well as a basis for S_z. When the user submits $|z\rangle|S_z\rangle$ to the bank for verification, the bank can use its knowledge of a basis for S_z to decide membership in both S_z and $S_z^\perp$, and thereby implement a projection onto $|S_z\rangle$. On the other hand, suppose it were possible to map $|z\rangle|S_z\rangle$ to $|z\rangle|S_z\rangle^{\otimes 2}$, using $\text{poly}(n)$ queries to the bank (with the submitted bills returned to the user on successful or even failed verifications). Then by using the oracle A to simulate the queries to the bank, we could also map $|d_S\rangle|S\rangle$ to $|d_S\rangle|S\rangle^{\otimes 2}$ in the scheme of Theorem 9.4.1, and thereby break that scheme. But this would contradict Theorem 9.4.1. It follows that mapping $|z\rangle|S_z\rangle$ to $|z\rangle|S_z\rangle^{\otimes 2}$ requires $\Omega(2^{n/2})$ queries to the bank. □

9.5 Instantiation

In light of Theorem 9.4.1, an obvious question is whether it's possible to "instantiate" the hidden subspace construction, thus getting rid of the oracle A. This would require providing an "obfuscated" description of the subspaces S and $S^\perp$—one that let the user efficiently apply χ_S and $\chi_{S^\perp}$ in order to verify a banknote, yet that didn't reveal a basis for S.

In their paper, Aaronson and Christiano [AC12] offered one suggestion for this, based on low-degree polynomials. More concretely: assume that the serial number of a banknote $|S\rangle$ encodes two sets of (say) degree-3 polynomials,

$$p_1, \ldots, p_m : \mathbb{F}_2^n \to \mathbb{F}_2 \text{ and } q_1, \ldots, q_m : \mathbb{F}_2^n \to \mathbb{F}_2,$$

for some $m = O(n)$, such that $p_1, \ldots, p_m$ simultaneously vanish on S (and only on S), while $q_1, \ldots, q_m$ simultaneously vanish on $S^\perp$ (and only on $S^\perp$). Given S, it's easy to generate such polynomials efficiently—for example, by choosing random degree-3 polynomials that vanish on some canonical subspace S_0 (say, that spanned by the first $n/2$ basis vectors), and then acting on the polynomials with a linear transformation that maps S_0 to S. For additional security, we could also add noise: an ϵ fraction of the polynomials can be chosen completely at random.

Clearly, given these polynomials, we can use them to compute the characteristic functions χ_S and $\chi_{S^\perp}$, with no oracle needed. But given only the polynomials, there's at least no *obvious* way to recover a basis for S, so one might hope that copying $|S\rangle$ would be intractable as well.

To give evidence for that, Aaronson and Christiano [AC12] gave the following security reduction.

Theorem 9.5.1 ([AC12]). *Suppose the hidden subspace mini-scheme can be broken, with the functions χ_S and $\chi_{S^\perp}$ implemented by sets of degree-3 polynomials $p_1, \ldots, p_m : \mathbb{F}_2^n \to \mathbb{F}_2$ and $q_1, \ldots, q_m : \mathbb{F}_2^n \to \mathbb{F}_2$ respectively. Then there exists a polynomial-time quantum algorithm that, given the p_i's and q_i's, finds a basis for S with success probability $\Omega(2^{-n/2})$.*

Proof. The argument is surprisingly simple. Suppose a counterfeiting algorithm C exists as in the hypothesis, which maps $|S\rangle$ to $|S\rangle^{\otimes 2}$ using $p_1, \ldots, p_m$ and $q_1, \ldots, q_m$. Then consider the following algorithm to find a basis for S with $\sim 2^{-n/2}$ success probability:

(1) Prepare a uniform superposition over all n-bit strings.

(2) Compute $p_1, \ldots, p_m$ on the elements of the superposition, and condition on getting the all-0 outcome—thereby collapsing the superposition down to $|S\rangle$ with success probability $2^{-n/2}$.

(3) If the measurement succeeds, then run C repeatedly, to map $|S\rangle$ to $|S\rangle^{\otimes m}$ for some $m = \Theta(n)$.

(4) Measure each of the copies of $|S\rangle$ in the computational basis to learn a basis for S.

□

An algorithm that finds a basis for S with exponentially-small success probability, $\Omega(2^{-n/2})$, might not sound impressive or unlikely. However, notice that if one tried to find a basis for S by random guessing, one would succeed with probability only $2^{-\Omega(n^2)}$.

9.6 Attacks

Unfortunately, some recent developments have changed the picture substantially.

First, in 2014, Pena et al. [PFP15] proved that the *noiseless* version of the low-degree polynomial scheme can be broken, if the scheme is defined not over $\mathbb{F}_2$ but over a finite field of odd characteristic. Their proof used Gröbner basis methods, and it actually yielded something stronger than a break of the money scheme: namely, a polynomial-time classical algorithm to recover a basis for S given $p_1, \ldots, p_m$ (the dual polynomials, $q_1, \ldots, q_m$, aren't even needed). In the $\mathbb{F}_2$ case, Pena et al. conjectured, based on numerical evidence, that their Gröbner basis techniques would at least yield a quasipolynomial-time algorithm.

In any case, the noisy version of the low-degree polynomial scheme wasn't affected by these results, since the Gröbner basis techniques break down in the presence of noise.

However, in a recent dramatic development, Christiano and others have fully broken the noisy low-degree polynomial scheme. They did this by giving a quantum reduction from the problem of breaking the noisy scheme to the problem of breaking the noiseless scheme, which Pena et al. already solved. (Thus, unlike Pena et al.'s result, this part of the attack is specific to quantum money.)

The reduction is extremely simple in retrospect. Given the polynomials $p_1, \ldots, p_m$ and $q_1, \ldots, q_m$, and given a money state $|S\rangle$, our goal is to use measurements on $|S\rangle$ to decide which p_i's and q_i's are "genuine" (that is, vanish everywhere on S), and which ones are "noisy" (that is, vanish on only a $\sim 1/2$ fraction of points in S).

Given a polynomial (say) p_j, we can of course evaluate p_j on $|S\rangle$ and then measure the result. If p_j is genuine, then we'll observe the outcome 0 with certainty, while if p_j is noisy, we'll observe the outcomes 0 and 1 with roughly equal probability. The problem with this approach is that, if p_j is noisy, then the measurement will destroy the state $|S\rangle$, making it useless for future measurements.

Here, however, we can use the remarkable idea of *single-copy tomography*, which Farhi et al. [FGH+10] introduced in 2010, in the context of breaking other public-key quantum money schemes. The idea is this: after we make a measurement that corrupts $|S\rangle$ to some other state $|\psi\rangle$, we then use amplitude amplification (see Lecture 5) to restore $|\psi\rangle$ back to $|S\rangle$. Now, amplitude amplification requires the ability to reflect about both the initial state $|\psi\rangle$ and the target state $|S\rangle$. But we can do this by using the polynomials $p_1, \ldots, p_m$ and $q_1, \ldots, q_m$! In particular, to reflect about $|S\rangle$, we negate the amplitude of each basis state iff a sufficient fraction of the p_i's and q_i's evaluate to 0, while to reflect about $|\psi\rangle$, we negate the amplitude of each basis state iff a sufficient fraction of the p_i's and q_i's evaluate to 0, *and* the polynomial p_j that we just measured evaluates to whichever measurement outcome we observed (0 or 1). This approach lets us proceed through the p_i's and q_i's, identifying which ones are noisy, without permanent damage to the state $|S\rangle$.

Taking the contrapositive of Theorem 9.5.1, a remarkable corollary of the new attack is that, given a noisy set of low-degree polynomials $p_1, \ldots, p_m$ and $q_1, \ldots, q_m$, there's a polynomial-time quantum algorithm that recovers a basis for S with success probability $\Omega(2^{-n/2})$.

Again, a naïve algorithm would succeed with probability only $2^{-\Omega(n^2)}$ (with some cleverness, one can improve this to 2^{-cn} for some large constant c). We don't know any quantum algorithm for this problem that succeeds with probability $\Omega(1)$, nor do we know a classical algorithm that succeeds with probability $\Omega(2^{-n/2})$. Thus, the algorithm that emerges from this attack seems to be a genuinely new quantum algorithm, which is specialized to the regime of extremely small success probabilities. The most surprising part is that this algorithm came from an attack on a quantum money scheme; we wouldn't know how to motivate it otherwise.

9.7 Future Directions

Returning to quantum money, though, what are the prospects for evading this attack on the low-degree polynomial scheme? All that's needed is some new, more secure way to instantiate the black-box functions χ_S and $\chi_{S^\perp}$ in Aaronson and Christiano's hidden subspace scheme. As a sort of stopgap measure, one could instantiate χ_S and $\chi_{S^\perp}$ using a recent idea in cryptography known as *indistinguishability obfuscation*, or i.o. I.o. has the peculiar property that, when we use it to obfuscate a given function f, we can generally prove that, if there's *any* secure way to obfuscate f, and if moreover i.o. itself is secure, then i.o. yields a secure obfuscated implementation of f. In this case, the upshot would be that, *if* the characteristic functions of the subspaces S and $S^\perp$ can be obfuscated at all, then i.o. is one way to obfuscate them! Ironically, this gives us a strong argument in favor of using i.o. even while giving us little evidence that *any* secure implementation is possible.

Other ideas include using implementing χ_S and $\chi_{S^\perp}$, or something like them, using lattice-based cryptography, or returning to other ideas for quantum money, such as the Farhi et al. [FGH+12] knot-based scheme, and trying to give them more rigorous foundations. In the meantime, there are many other wonderful open problems about public-key quantum money. Here's a particular favorite: is public-key quantum money possible relative to a *random* oracle?

Besides quantum money, another fascinating challenge is *copy-protected quantum software*: that is, quantum states $|\psi_f\rangle$ that would be useful for evaluating a Boolean function f, but not for preparing additional states that could also be used to evaluate f. In forthcoming work, Aaronson and Christiano show that copy-protected quantum software is "generically" possible relative to a classical oracle. But it remains to give a secure "real-world" implementation of quantum copy-protection for general functions f, or even a good candidate for one.