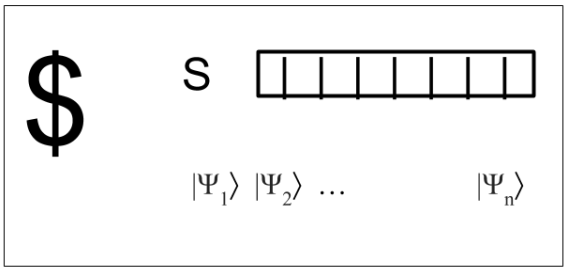
Lecture 8, Thurs Feb 9: More on Quantum Money, BB84 QKD

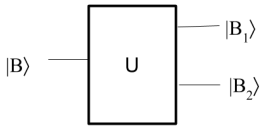
Guest Lecture by Supartha Podder



Continuation of Quantum Money

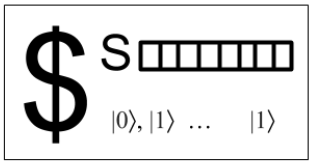
Last time we discussed how classical money is copyable and described a scheme for making money uncopyable through an application of the No-Cloning Theorem.

Let's consider a counterfeiter who wants to take a copy of a legitimate bill B and submit it for verification.



Say the counterfeiter decides to measure all qubits in the $|0\rangle, |1\rangle$ basis.

Their new bill becomes:



- s gets copied
  - (classical information)
- Puts $|0\rangle$ or $|1\rangle$ as each qubit.

So the bank will measure each qubit. The ones that should be in the $\{|0\rangle, |1\rangle\}$ basis are correct all of the time. But the ones that should be in the $\{|+\rangle, |-\rangle\}$ basis are correct in both bills only $\frac{1}{4}$ of the time.

Thus the probability that the counterfeiter succeeds (i.e., that both bills pass verification) is $(\frac{5}{8})^n$.

As we mentioned last time, it was recently shown that any such attack succeeds with probability at most $(\frac{3}{4})^n$.

Interactive Attack

There's a clever attack on Wiesner's scheme based around the assumption that verification involves giving the bank a bill, and then the bank returns the bill whether or not it passed verification.

We can start with a legitimate bill, then repeatedly go to the bank and ask them to verify it—but manipulating the qubits of the bill one at a time.

For example, if we set the first qubit to $|0\rangle$ and the bill still passes verification each and every time, then we've learned that the first qubit *should* be $|0\rangle$. Otherwise, we can successively try setting the first qubit to $|1\rangle, |+\rangle, |-\rangle$, and see which choice makes the bank consistently happy. Then, once we know, we move on to toggling the second qubit, and so on.

OK, but surely the bank wouldn't be so naïve as to return the bill even if it fails verification! We should assume instead that if verification fails (or fails often enough), then the bank alerts the police or something.

Can we come up with an attack that works even then? A recent paper by Nagaj and Sattath points out that we can!

Recall the **Elitzur Vaidman Bomb**. The general idea is that by making a succession of measurements, none of which reveals that much by itself, we can with a high probability of success learn whether a system is a certain state, without triggering a "bad event" that would happen if the system were actually measured to be in that state (such as a bomb going off). Applying a similar idea to quantum money gives us an…

Attack Based on the Elitzur Vaidman Bomb

Set $|c\rangle$ to $|0\rangle$

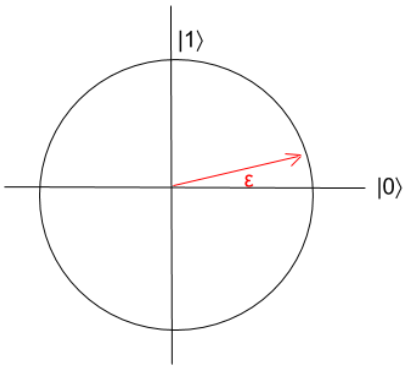
Let $|\Psi\rangle$ be the qubit of the banknote we're trying to learn

Repeat $\frac{\pi}{2\varepsilon}$ times:

Apply the rotation R_ε to $|c\rangle$

Apply a cNOT gate to $|c\rangle|\Psi\rangle$

Send the bill to the bank for verification.



Suppose $|\Psi\rangle = 0$. Then each time we apply cNOT, we get $(\cos\varepsilon|0\rangle + \sin\varepsilon|1\rangle)|0\rangle = \cos\varepsilon|00\rangle + \sin\varepsilon|11\rangle$

Most of the time $|c\rangle$ will stay at $|0\rangle$.

At each step, the probability of getting caught (i.e. failing verification) is $\sin^2\varepsilon$.

Thus Prob[getting caught at all] is upper-bounded by $\leq \frac{\pi}{2\varepsilon}\sin^2\varepsilon = O(\varepsilon)$

A similar analysis can be done if $|\Psi\rangle$ is $|1\rangle$ or $|-\rangle$: we're unlikely to get caught, *and* the $|c\rangle$ qubit keeps "snapping back" to $|0\rangle$.

But if $|\Psi\rangle = |+\rangle$, then something different happens: the $|c\rangle$ qubit gradually rotates from $|0\rangle$ to $|1\rangle$.

So when we measure at the end, we can distinguish $|+\rangle$ from the other states, because it's the only one that causes the $|c\rangle$ qubit to rotate to $|1\rangle$.

By symmetry, we can give analogous procedures to recognize the other three possible states for $|\Psi\rangle$.

So then we just iterate over all n qubits in the bill, learning them one by one, just like in the previous attack on Wiesner's scheme.

Can Wiesner's scheme be fixed to patch this vulnerability?

Yes! The bank can just give the customer a *new* bill (of the same value) after each verification, instead of the bill that was verified.

There's an additional problem with Wiesner's scheme, as we've seen it. Namely, it requires the bank to hold a huge amount of information: one secret for every bill in circulation. However, the paper (Bennett Brassard Breidbart Wiesner 82) points out how to circumvent this, by basically saying: let f be a pseudorandom function with a secret key k, so that for any serial number s, the bank can compute $f_k(s)$ for itself, rather than needing to look it up.

Of course the bank had better keep k itself secret: if it leaks out, the entire money system collapses! But assuming that k remains a secret, why is this secure?

We use a reduction argument. Suppose that the counterfeiter can copy money by some means. What does that say about f_k? If it were truly random, then the counterfeiter wouldn't have succeeded. So by checking whether the counterfeiter succeeds, we can distinguish f_k from a random function. So f_k wasn't very good at being pseudorandom!

Note that with this change, we give up on provable security, of the sort we had with Wiesner's original scheme. Now we "only" have security assuming that f_k is computationally intractable to distinguish from random. (And a recent result by Prof. Aaronson shows that some computational assumption is necessary, if we don't want the bank to have to store a giant database.)

However, even after we make the improvements above, Wiesner's scheme still has a fundamental problem, which is that to verify a bill, you need to go to the bank. And if you have to go to the bank, then arguably you might as well have used a credit card or something instead! The point of cash is supposed to be that we don't need a bank to complete a transaction. Which brings us to...

Public-Key Quantum Money

This is quantum money that *anyone* can verify using a "public key," but that can only be produced or copied using a "private key" known only to the bank.

For formal definitions see (Aaronson 2009), (Aaronson, Christiano 2012).
With this sort of scheme, you'll *always* need computational assumptions on the counterfeiter, in addition to quantum mechanics. Why? Because a counterfeiter with infinite computational power could always just try *every* possible quantum state (or an approximation thereof) on the appropriate number of qubits, until it found one that made the public verification procedure accept.

Quantum Key Distribution

Now we'll discuss something closely related to quantum money, but that doesn't require storing quantum states for long times—and that, for that reason, is actually practical today (though so far there's only a small market for it).

Key distribution is a fundamental task in cryptography. It just means causing two agents, Alice and Bob, to share a secret key (without loss of generality, a uniformly random string), when they didn't have one before.

Once Alice and Bob share a long enough key, they can then exchange secret messages, using the central technique in cryptography called the **One-Time Pad**.

Given a shared key $k \in \{0,1\}^n$

Alice has a secret message $m \in \{0,1\}^n$

Alice sends the ciphertext $c = m \oplus k$, where $\oplus$ denotes bitwise XOR

Bob decodes the message m as $m = c \oplus k$

As its name implies, the One-Time Pad can only be used once securely with a given key k, so it requires a large amount of sharing of keys. In fact, in the classical world, it's been proven that if they want to communicate securely, Alice and Bob either need initial secret information in common, or else they must make computational assumptions on the eavesdropper Eve.

The great discovery of Quantum Key Distribution was that quantum mechanics lets us get encryption with no computational assumptions! (But we do need communication channels capable of sending quantum states.)

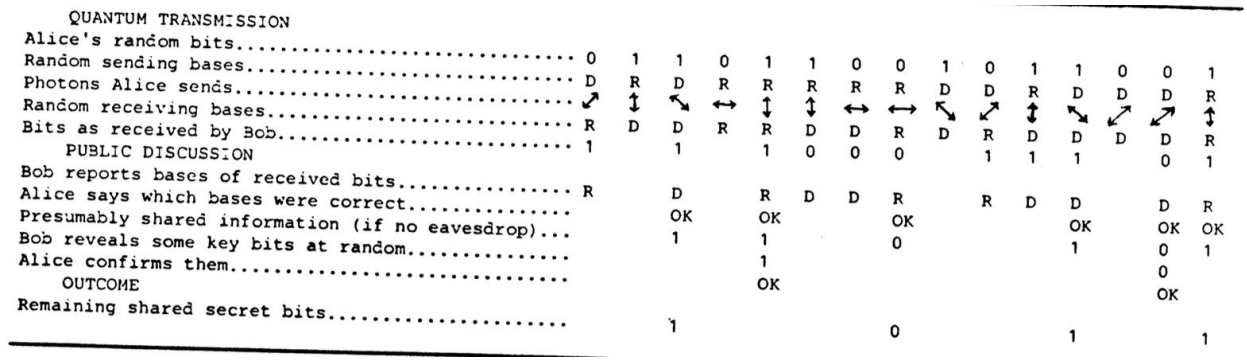
In cryptography, besides secrecy, an equally important goal is authentication. However, we're only going to deal with secrecy.

BB84

We'll describe the BB84 scheme, the first full quantum key distribution scheme. This scheme was proposed by Bennett and Brassard in 1984, though it was partly anticipated in Wiesner's paper (the same one that introduced quantum money!). It circumvents the issues we've seen in maintaining a qubit, because it only requires coherence for the time it takes for communication between Alice and Bob.

There are companies that are already doing quantum key distribution through fiber optic cables over up to about 10 miles. In addition, just this year a team from China demonstrated QKD over distances of thousands of miles, by sending photons to and from a satellite that was launched into space for that express purpose.

Here's a diagram from the original paper that shows how BB84 works.



The basic idea is that you're trying to establish some shared secret knowledge and you want to know for certain that no eavesdroppers on the channel can uncover it. You've got a channel in which to transmit quantum information, and a channel in which to transmit classical information. In both, eavesdroppers may be able to listen in (no secrecy). But in the classical channel, we'll assume you at least have *authenticity*: Bob knows that any messages really come from Alice and vice versa.

- So Alice chooses a string x of random bits $\epsilon \{0,1\}^n$
- And another string y of random bits $\epsilon \{0,1\}^n$, which she uses to decide which basis to encode each bit from x in.
- She then encodes each bit of x in the $\{|0\rangle, |1\rangle\}$ basis (in the diagram it's R), if the corresponding bit of y is 0, or the $\{|+\rangle, |-\rangle\}$ basis (D), if the corresponding bit of y is 1
- Then she sends over the qubits to Bob.
- Bob picks his own random string $y' \epsilon \{0,1\}^n$ and uses y'_i to decide in which basis

to decode the i^{th} qubit sent over (picking again between D and R)

Now Alice and Bob share which bases they picked to encode and measure the bits of x (the strings y and y'). They discard any bits of x for which they didn't pick the same basis (which will be about half the bits).

At this point we consider an eavesdropper Eve who was watching the qubits as they were were sent over. The whole magic of using qubits is that if Eve tries to measure the qubits, then she inherently changes what Bob receives! Sure, if she measures a $\{|0\rangle, |1\rangle\}$ qubit in the $\{|0\rangle, |1\rangle\}$ basis, then the qubit doesn't change. But what if she's unlucky, and measures a $\{|+\rangle, |-\rangle\}$ qubit in the $\{|0\rangle, |1\rangle\}$ basis? And eventually, she almost certainly *will* be unlucky.

In more detail: suppose Alice sent $|+\rangle$, then Eve measured $|0\rangle$ and passed that along to Bob. Then even if Bob measures in the $\{|+\rangle, |-\rangle\}$ basis (i.e., the "right" basis), he has a 50% chance of measuring $|+\rangle$ and a 50% chance of measuring $|-\rangle$. In the latter case, Alice and Bob will be able to see that the channel was tampered with.

So Alice and Bob can verify that no one listened in to their qubit transmission by making sure that some portion of their qubits that *should* match, do match. Of course, after Alice and Bob discuss those qubits over the channel, they aren't going to be secret anymore! But they've still got all the others.

If any of the qubits didn't match, then Alice and Bob deduce that Eve eavesdropped. So then they can just keep trying again and again until they can get a batch where no one listened in. At worst, Eve can prevent Alice and Bob from ever communicating by listening in constantly. But we can prevent a situation where Alice and Bob *think* their shared key is secure even though it isn't.

Again, once Alice and Bob share a secret key, they can then use some classical encryption scheme, like the One-Time Pad.