Quantum Money

$$|\$\rangle$$

Peter W. Shor
M.I.T., Cambridge, MA, U.S.A.

Joint work with:
Edward Farhi, David Gosset,
Avinatan Hassidim, Andrew Lutomirski

Outline:

- Background: Cryptography and Quantum Money
- What we did that didn't work
- What we did that we think might work

Traditional cryptography is called *symmetric cryptography*, where each pair of parties who want to communicate have a secret key, shared in advance.

German World War II Engima machine

The possibiity of doing cryptography (called *public key* cryptography) done without secret keys was raised by Ralph Merkle in 1974.

Public Key Cryptography

The first convincing truly example was Diffie and Hellman's *key exchange* protocol.

This lets two parties agree on a secret key without any pre-existing secret knowledge.

This key can then be used for a symmetric cryptosystem, or as a one-time pad.

The security of Diffie-Hellman depends on the difficulty of computing discrete logarithms.

Quantum Cryptography

Two of the first two quantum cryptographic protocols were Wiesner’s protocol for quantum money, and the BB84 protocol for key exchange.

Wiesner’s protocol for quantum money inspired BB84.

Both of these depend on the quantum no-cloning theorem.

One problem with money is that you can make copies.

Quantum states satisfy the no-cloning theorem, which says you cannot make a copy of an unknown quantum state.

One might think this will immediately let us use quantum states for money.

It's actually quite a bit harder than it sounds, but we give a proposal for creating unforgeable quantum states.

No Cloning Theorem (1982)

There is no quantum transformation taking $|\,\psi\rangle\,|\,0\rangle$ to $|\,\psi\rangle\,|\,\psi\rangle$ for an unknown state $|\,\psi\rangle$.

Why not? This transformation isn't unitary:
$|\,\phi\rangle\,|\,0\rangle$ would go to $|\,\phi\rangle\,|\,\phi\rangle$.

But

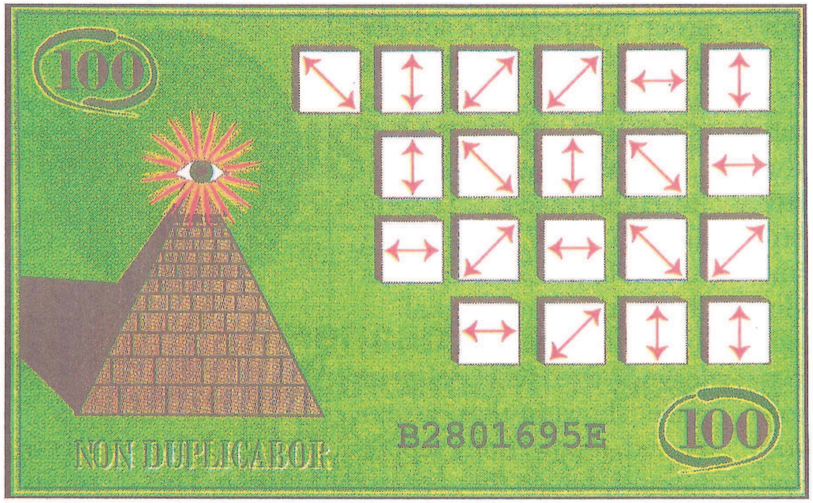
$$\alpha = \langle\phi\,|\,\psi\rangle\,\langle 0\,|\,0\rangle > \langle\phi\,|\,\psi\rangle\,\langle\phi\,|\,\psi\rangle = \alpha^2$$

unless $\alpha = 0$ or $\alpha = 1$.

Thus, angles are not preserved, and the cloning transformation is not unitary.

History of Quantum Money

Stephen Wiesner's idea for making quantum money, (circa 1970, published 1983).



In each bill, there is a sequence of quantum states in one of two complementary bases (so one of $|\updownarrow\rangle, |\leftrightarrow\rangle$ $|\nearrow\rangle, |\searrow\rangle$). By the quantum no-cloning theorem, anyone who does not know the polarizations of these states cannot copy them.

How to check the money? The mint knows the polarizations, and so can easily check it.

We want the merchant to be able to verify that the bill is legit without sending it back to the mint.

If the merchant knows the quantization axis and eigenvalue of each qubit, then the merchant can verify the money.

However, he could also make new bills exactly like the one he got.

We would like a verification procedure that does not allow the merchant to make fresh bills.

Cryptography Background and Motivation

For many years, cryptography was done with *ad hoc* cryptosystems, many of which were eventually broken.

Over the last few decades, cryptography has become much more mathematical, and theoretical computer scientists try to prove security of cryptosystems.

There are two kinds of proofs of security in cryptography: security through information and security through complexity.

Definitions

Informationally Secure	*Computationally Secure*
BB84 key exchange, one-time pad	Diffie-Hellman key exchange, RSA cryptosystem
No matter how powerful a computer an adversary has, he will not be able to break the cryptosystem, because he doesn't have access to enough information.	The security of the cryptosystem relies on the difficulty of solving some computationally hard problem

Disadvantages

Informationally Secure

Many problems cannot be solved with informationally secure cryptosystems. For example, an informationally secure cryptosystem for encryption of messages requires a key as long as the message. (This is achieved by a one-time pad.)

Computationally Secure

It is hard to prove anything about the security of computationally secure cryptosystems. For example, the only reason for believing prime factorization is hard is that nobody has been able to solve it yet.

Quantum cryptography

The BB84 protocol for quantum key distribution can be proved informationally secure, assuming the laws of quantum mechanics. This solves a task which is impossible to perform with an informationally secure protocol and classical computing.

One genesis for this research was wondering whether there were any tasks that a quantum computer might perform with computational security, but which were impossible for a digital computer to perform.

We believe we have identified one.

Task: Unforgeable States

We would like to make quantum states that

a) can be verified.

b) cannot be duplicated.

Task: Unforgeable States

That is, we would like one of the players in the protocol (we will call her the mint) to be able to make a state $|\psi_i\rangle$, and a verification protocol P_i, so that

a) $|\psi_i\rangle$ passes the test P_i.

b) The test P_i does not destroy $|\psi_i\rangle$.

c) a possible counterfeiter holding both the state $|\psi_i\rangle$ and knowing the protocol P_i cannot produce a state of two quantum systems (possibly entangled) that both pass the test P_i.

One-of-a-Kind States

In fact, in our protocol, we think that not even the mint can efficiently make another copy of the state $|\psi_i\rangle$ that pases the test P_i.

Uses for Unforgeable States: Quantum Money

The mint makes quantum states, and gets pairs $|\,\psi_i\rangle$, P_i.

The mint publishes a list of valid pairs i, P_i somewhere secure (so nobody can add an extra pair to the list).

It then hands out some $|\,\psi_i\rangle$, together with i, to a customer who wants quantum money.

Then anybody with $|\,\psi_i\rangle$ who knows i (and has a quantum computer) can check that it is a valid quantum money state; i.e., that i is on the list, and $|\,\psi_i\rangle$ passes the test P_i.

Uses for Unforgeable States: Quantum ID Cards

You could put a unforgeable quantum state into an ID card.

These ID cards could be stolen, but they could not be forged.

Of course, for both money and quantum ID cards, you need to have long-lived quantum states.

You have to be careful.

For one of the potential protocols we looked at,

- A potential counterfeiter holding just $|\,\psi_i\rangle$ could not make a copy of it (because of the no cloning theorem).
- Knowing just the verification protocol P_i, it appeared very difficult to find a state that passed it.
- But both holding $|\,\psi_i\rangle$ and knowing P_i, a counterfeiter could produce two copies of $|\,\psi_i\rangle$.

Failed Protocol

Creating the money:

1. Choose a random product state
$$|\,\psi\rangle = |\,\psi_1\rangle \otimes |\,\psi_2\rangle \otimes |\,\psi_3\rangle \otimes \ldots \otimes |\,\psi_n\rangle\,.$$
where the $|\,\psi_i\rangle$ are uniform in Haar measure.

2. Choose Cn k-clauses (i.e., local Hamiltonians on k qubits) such that $|\,\psi\rangle$ has zero energy for each of the clauses.

The quantum money state is $|\,\psi\rangle$ and the verification procedure is checking that it indeed has zero energy on all the clauses.

How to Break the Failed Protocol

In breaking this protocol, we came up with a possible new algorithmic tool for quantum computing.

Theorem (Quantum State Restoration)
Given a Hamiltonian H, and a ground state $|\psi\rangle$ of H, one can do tomography on (i.e., estimate properties of) a small number of qubits of $|\psi\rangle$ and measure local properties of it without destroying it.

Question: are there any algorithmic uses for state restoration (besides breaking quantum money)?

How does our quantum money protocol work?

We will

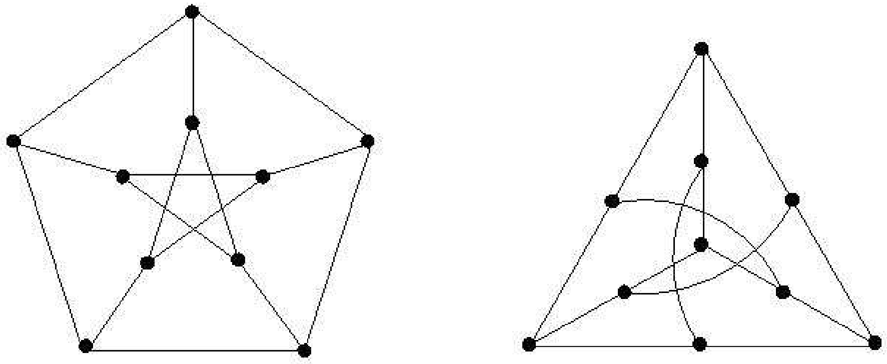
1. Give a failed protocol based on graph isomorphism. This helps motivate our current protocol.

2. Give our best current candidate for quantum money, created by using the replacing graphs with diagrams of knots.

3. Discuss a general template for building quantum money protocols.

Background on Graph Isomorphism

Two graphs are isomorphic if you can relabel the vertices of one to obtain the other.



Quantum Computing on Graph Isomorphism

Suppose we could take a graph G and create the state

$$\frac{1}{\sqrt{n!}} \sum_{\pi \in S_n} |\,\pi(G)\rangle$$

Then we could solve graph isomorphism.

How? Given graphs G_1 and G_2, we prepare the state

$$\frac{1}{\sqrt{n!}} \sum_{\pi \in S_n} |\,\pi(G_1)\rangle \otimes \frac{1}{\sqrt{n!}} \sum_{\pi \in S_n} |\,\pi(G_2)\rangle$$

If the graphs are isomorphic, these are equal. We test whether the state is a $+1$ eigenstate of the SWAP operator.

Moral from Previous Slide

Creating the equal superpositions of a graph

$$\frac{1}{\sqrt{n!}} \sum_{\pi \in S_n} |\, \pi(G)\rangle$$

seems to be hard.

It turns out that for lattices, if you could create the equal superposition of points near vectors in a lattice

$$\frac{1}{\sqrt{N}} \sum_{w : |w-v| \leq \epsilon, v \in L} |\, w\rangle$$

then you could find short vectors in the lattice. This is also a problem believed to be hard classically.

Attempt using Graph Isomorphism

Now, consider the following algorithm.

The mint starts with the equal superposition of all graphs

$$\frac{1}{2^{n(n-1)/4}} \sum_{G} |\, G\rangle\,.$$

This is easy, because you can put each edge in a superposition of present and absent.

The mint then measures some property of graphs which is invariant under permutations of the vertices (e.g., the spectrum). Suppose the spectrum is S. Then we are in the state

$$\frac{1}{\sqrt{N_S}} \sum_{G:\mathrm{Spec}(G)=S} |\, G\rangle$$

Testing this state

The quantum money is: $|\,\$_S\rangle = \frac{1}{\sqrt{N}} \sum_{G:\text{Spec}(G)=S} |\,G\rangle\,.$

To test it, we check

1. that Spec$(G) = S$,
2. that the state is invariant under the relabeling of two of the vertices.

Any state that passes these tests must be a superposition

$$\sum_G \alpha_G \sum_\pi |\,\pi(G)\rangle = \sum_G \alpha_G\,|\,\$_G\rangle$$

for some set of graphs G with Spec$(G) = S$.

Good News

We have the state:

$$|\,\$_S\rangle = \frac{1}{\sqrt{N}} \sum_{G:\text{Spec}(G)=S} |\,G\rangle$$

One thing we could do is measure this state, to get a graph with Spec$(G) = S$. But then we can't create

$$|\,\$_G\rangle = \frac{1}{\sqrt{n!}} \sum_{\pi \in S_n} |\,\pi G\rangle$$

unless we can solve graph isomorphism.

Bad News

We can solve graph isomorphism for random graphs.

If constructing the isomorphism is easy for a graph G, we can then create the state

$$|\,\$_G\rangle = \frac{1}{\sqrt{n!}}\sum_{\pi\in S_n}|\,\pi G\rangle$$

We can do this by creating the superposition over all permutations, applying the permutation, and then uncomputing the permutation.

What to do now?

To use graph isomorphism for quantum money, we need to start with an equal superposition just over hard graphs. We don't know how to do that.

The new idea: instead of graph isomorphism, use a similar problem which doesn't have the drawback that it is easy for an average case.

Are there such problems?

We looked through a lot of candidates which didn't work before identifying what we think is a good one.

We propose using knots and knot invariants.

We have to vary the protocol somewhat to make them work.

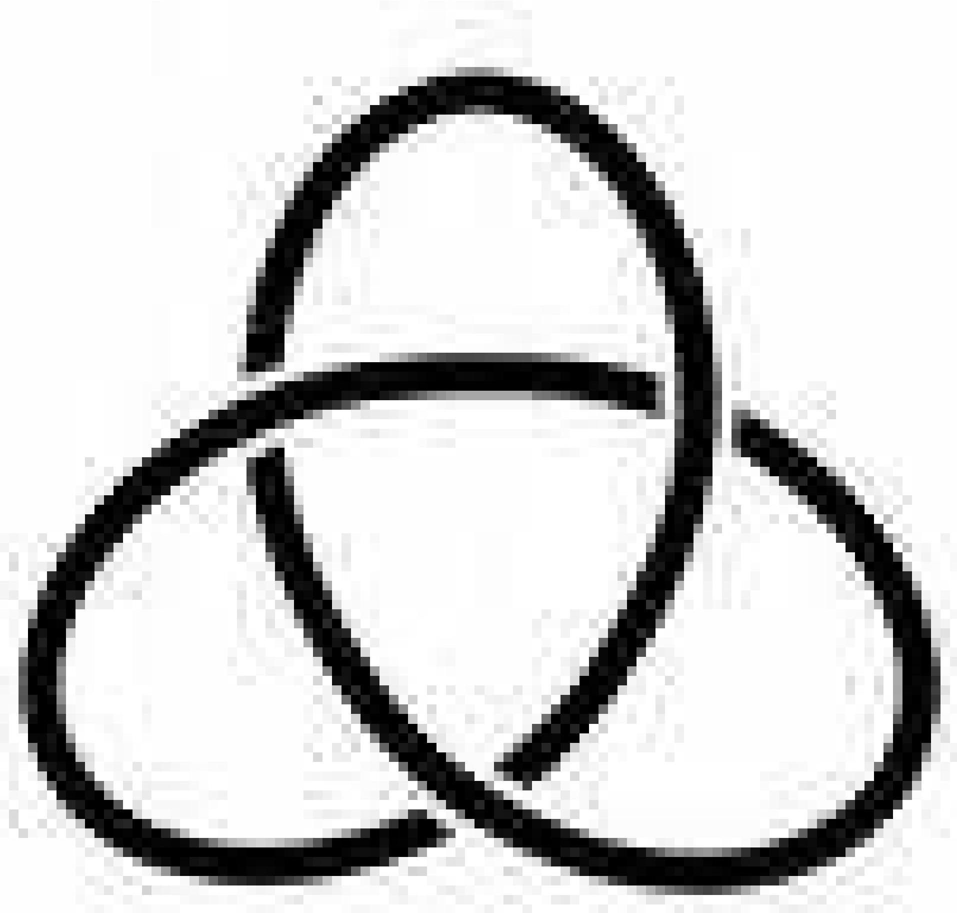
Knots

Knot diagram are similar enough to labelings of graphs that we can use them in our money scheme.

A knot diagram is a drawing of a knot in the plane.

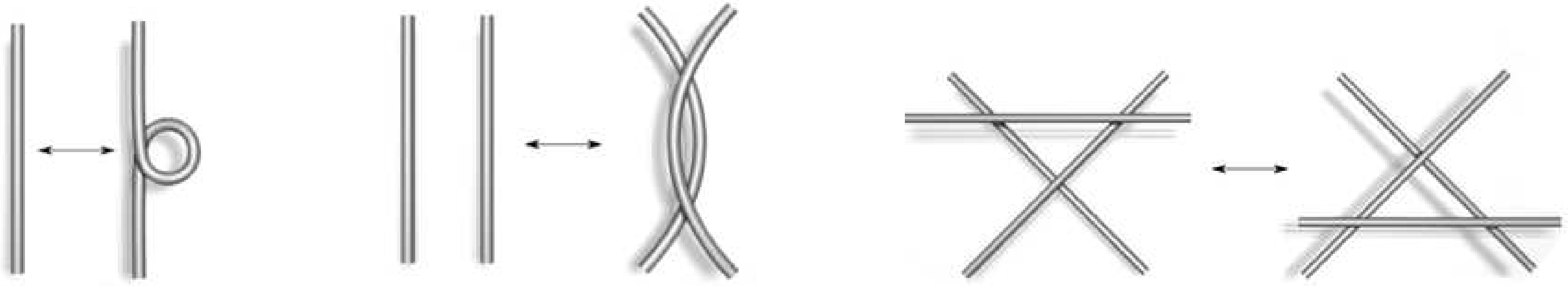
If you have a knot, then there are many different diagrams that represent the same knot. Testing whether two knots are given by the same diagram is believed to be a hard problem.

A Trefoil Knot



Reidemester Moves

If you have two knot diagrams that do give the same knot, you can move from one to the other using Reidemeister moves.



Our idea is thus to replace graph isomorphism with knot diagrams, and relabelings of vertices with Reidemeister moves.

Knot Invariants

For our template, we need some function f mapping knot diagrams into values that depend only on the knot and not the diagram (analogous to the spectrum of G for our failed attempt with graph isomorphism). These function are called *knot invariants*.

We need to choose one that is computable in polynomial time. The Alexander polynomial is the best known of these, but there are others. The Alexander polynomial maps a knot into a polynomial with integer coefficients. For the trefoil knot, it is $t^2 - t + 1$.

The Broad Outline of Our Proposal

The mint starts with the superposition of all diagrams of knots. It then measures the Alexander polynomial of these knots (or another polynomial time computable knot invariant) to get

$$\alpha_{p(t)} \sum_{A(K)=p(t)} |\, K\rangle$$

The verifier checks that the superposition given to him has the correct Alexander polynomial, and that this superposition is invariant under Reidemeister moves. If the state passes these two tests, he accepts it as valid quantum money.

But Infinity ...

There are an infinite number of diagrams for the same knot. Thus, we cannot use an equal superposition of all knot diagrams.

What we do is to take knot diagrams with between n_1 and n_2 crossings, and weight them with some probabilities p_k that depend only on the number of crossings k, so that most of the weight is at some k which is substantially less than n_2. We then have to generalize our quantum money template to work for non-uniform distributions on objects.

This can be done by using the weighting from reversible Markov chains.

Another Problem

Another difficulty we've introduced by replacing graphs with knot diagrams is that it we need to create the uniform superposition over all knot diagrams with a given number of crossings.

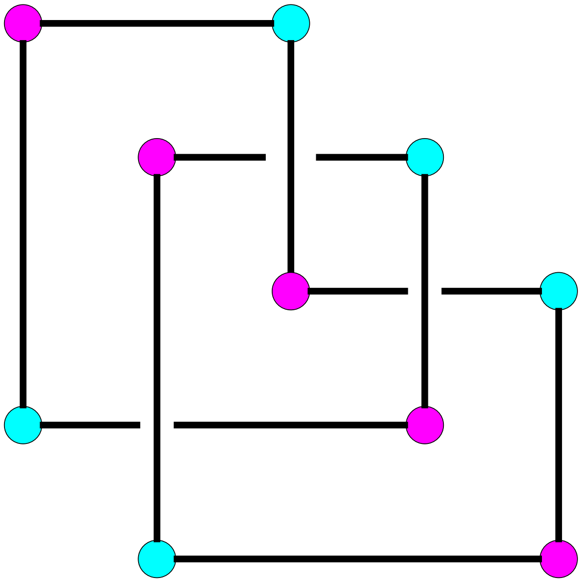
There don't seem to be any simple algorithms for this.

We can fix this by using grid diagrams of knots.

Reidemeister moves get replaced by grid moves.

Instead of weighting by the number of crossings, we weight by the size of the grid.

Grid Diagrams



For a grid diagram, you put $2n$ points on an $n \times n$ grid, two in each row and column. You connect the points in the same row and column, where vertical lines go over horizontal lines.

Grid diagrams have the advantage that it is really easy to generate the superposition of all grid diagrams, and also fairly easy to compute the Alexander polynomial of knots.

How could you break this protocol? The obvious way is to map

$$\sum_{i=1}^{N} |\, i\rangle \rightarrow \sum_{i=1}^{N} |\, G_i\rangle$$

where G_i is the ith grid diagram associated with some knot.

For this, you need an efficient 1-1 reversible mapping from i to grid diagrams of a give size associated with a given knot.

Mathematicians don't even know of an efficient algorithm to tell whether two grid diagrams are associated with the same knot. (However, they can do this in practice using knot invariants.)

A general template

Suppose we have a class of objects C; we have a small set of permutations on these objects: π_1, π_2, ..., π_k; and we have a function f so that $f(\pi_i(x)) = f(x)$ for $x \in C$.

Now, we create the uniform superposition over all objects $x \in C$, and measure the invariant f. We then get the state

$$\frac{1}{\sqrt{N}} \sum_{x:f(x)=k} |\, x\rangle$$

for some random value of the invariant k.

How to test the money in this template

The verifier has a state purporting to be in the superposition

$$\frac{1}{\sqrt{N_k}} \sum_{x: f(x)=k} |\, x\rangle \, .$$

He first measures to make sure that $f(x) = k$. He then applies the permutations π_i and sees whether the state changes. If it does not, then it passes the test.

If the Markov chain obtained from the permutations π_i is rapidly mixing, then the quantum money state is essentially the only one that passes the test.

Is this money safe?

We have a proof sketch that, if there is a scheme that will counterfeit this money for general objects x, functions f and permutations π_i satisfying the above conditions, then you can use it to solve graph isomorphism.

We would like to show that, if the functions π_i and f are given to you as oracles, then you cannot break such a money scheme (and not use the assumption that graph isomorphism is hard).

Open Problems

Can we prove that our template (with a black-box set of objects replacing knots and black-box transformations replacing Reidemeister moves) is indeed secure, without assuming that graph isomorphism is hard?

Can we use the same template to produce other protocols for quantum money?

Are there other ways to produce quantum money?

Scott Aaronson has a proposal to create quantum money from polynomials that are invariant on subspaces of $\mathbb{Z}_2^n$.