Quantum Money

CSC2429: Advanced Topics in Quantum Theory
University of Toronto
December 2020

Hao Zhang, Logan Murphy

1 Introduction

One of the most interesting properties of quantum states is that they cannot be perfectly replicated; this is a foundational result in quantum information theory known as the no-cloning theorem. An active area of research has developed with the goal of leveraging this property for the creation of *quantum money*, i.e. money which somehow contains or is represented by a quantum state, and hence cannot be counterfeited. This concept was first proposed by Wiesner [Wie83], and later expanded upon by Aaronson [Aar09]. As technology for quantum computing matures, research in quantum money becomes increasingly relevant. Not only does quantum money provide a practical application of quantum computing, it also provides an environment for exploring concepts of quantum information processing.

The remainder of this report will proceed as follows. First, we will discuss some of the motivations and suggested applications of quantum money schemes, as well as a related area of research spurred by investigations into quantum money. Second, we will provide an overview of some of the first seminal results in quantum money research. Third, we will discuss some examples of recent work investigating public-key quantum money schemes.

2 Applications and Related Areas

The most obvious application of quantum money is in the name: creating money that is computationally inefficient to replicate. Note that an adversary with infinite computing power can break any quantum money scheme by simply trying to authenticate all possible serial numbers and quantum states. While Wiesner's construction requires users to go to the bank every time they want to verify a banknote, so-called *public-key* quantum money would allow anyone to perform authentication. This unclonable property can also be used to protect important documents such as identity cards, passports, and birth certificates. However, a major challenge currently preventing the implementation of any quantum money scheme is the limitations of our current technology for maintaining quantum states. Quantum money would be useless if the states used decohered within a matter of seconds.

While quantum money schemes may not be practical in the foreseeable future, concepts from these areas of research can contribute to quantum information processing. They provide a model to explore the implications of the uncertainty principle and no-cloning theorem. Therefore, research in this fields can yield insight into many other areas of theoretical quantum computing. Results from quantum money research have also generated a related area of quantum copy-protection, which we will describe in more detail.

2.1 Quantum Copy-Protection

Modern computer programs can be easily pirated, since classical bits can easily be replicated. However, as mentioned, arbitrary qubit states have the special property of being unclonable. Aaronson proposes to use this property to create software which cannot be replicated [Aar09]. To accomplish this, he proposes the idea of quantum copy-protection of *point functions*: Boolean functions that return "1" if the input is equal to a secret string, and "0" otherwise. Point functions can be used to produce password-authentication programs that can not be duplicated. Such a quantum

copy-protection scheme would consist of:

1. A $poly(n)$ sized quantum circuit V, which takes d_f (a classical description of f) as input and outputs a mixed state,ρ_f
2. A $poly(n)$ sized quantum circuit, C, which takes ρ_f and x as input and attempts to output $f(x)$. For correctness, it outputs $f(x)$ with high probability.

A software developer can use V to make multiple copies of ρ_f and sell them along with their software. The software would then require the user to evaluate $f(x)$ with C. To prevent pirating, we would require that it is computationally inefficient to make additional states that can be used to calculate $f(x)$, and to learn f from its input-output behavior.

3 Early Constructions

In this section, we will discuss some of the first major accomplishments in quantum money research. First, we will describe Wiesner's original proposal. Next, we will discuss some of Scott Aaronson's early contributions to the field, which included defining some concepts that have been at the core of current research activities.

3.1 Wiesner's Quantum Money Construction

The first quantum money construction was proposed by Wiesner is based on conjugate coding [Wie83]. Conjugate coding takes advantage of the fact that making a quantum measurement on a qubit will collapse the value of the qubit to the value measured. This means that measuring in the wrong basis can effectively destroy any information previously stored within the qubit.

The scheme uses n qubits per banknote. Creating a banknote is as follows:

1. Randomly sample bitstrings $M = \{0,1\}^n$ and $N = \{0,1\}^n$. Let M_i and N_i be the i^{th} bit of the respective string
2. For the i^{th} qubit in the banknote, encode N_i it in the standard basis if $M_i = 0$ and encode N_i in the $|+\rangle$, $|-\rangle$ basis if $M_i = 1$
3. Stamp a serial number, S, on the banknote and store the tuple (S, M, N) in a secure location

To verify whether a banknote is valid, the bank would look up S to get M and N, measure each qubit in the basis described by M and accept if it measures the string N. A later revision of this scheme suggested to use a random function generator, g, and a secret key k, to automatically map S to M and N, ($g_k(S) = MN$)[BZBW15]. This would allow the bank to store the single secret key k instead of a tuple for every banknote.

A brute force attempt to counterfeit one of these banknotes would require the attacker to measure each qubit in the correct basis, then prepare a second banknote with the same encodings. However,

the attacker doesn't know M, so for each qubit, they have a 50% chance of choosing the wrong basis. Furthermore, for each basis they choose incorrectly, the bank has a 50% chance of measuring the wrong bit. Therefore, the probability of a successful brute force attack is $(\frac{3}{4})^n$. However, this scheme is not query-secure, meaning that if the bank doesn't confiscate the banknote after every verification, then it can be broken. One such attack is detailed in [NSBU16]. The procedure and analysis of this attack can be found in Appendix C.

3.2 Aaronson's Formal Definition of Quantum Money

The first formal definition of quantum money was proposed by Aarsonson[Aar09]:

A quantum money scheme with key size n consists of:

1. A $poly(n)$ sized quantum circuit B with classical string, s as input and output a classical string e_s and mixed state ρ_s as output.
2. A $poly(n)$ sized quantum circuit A which takes a string e and state ρ as input and either accepts or rejects

A quantum money scheme is what Aaronson calls "private-key" if an adversarial quantum circuit, C, given $\rho_s^{\otimes k}$ can not produce more than k states that can be accepted by A with high probability. It is considered "public-key" if C also received e_s as input.

The circuit B can be considered as the bank in this case, with s being a secret key, e_s being the public key and ρ_s being the banknote. Circuit A can be considered as the authenticator which would be publicly available to everyone. The completeness error, ε, is the probability that A accepts given a valid pair produced by B. Aaronson also formally defines a soundness error, δ, which essentially measures the maximum probability of a counterfeiter succeeding.

The soundness error, δ, is defined as the following: For all counterfeiter circuits, C with input $\rho_s^{\otimes k}$ and outputs state σ_s on $k+r$ registers, where σ_i represents the i^{th} register, $\sum_i P(A(e_s, \sigma_s^i)\ accepts) \leq k + \delta$. Intuitively, δ measures the maximum probability of a counterfeiter succeeding.

3.3 Aaronson's First Public-Key Quantum Money Construction

The general idea of public-key quantum money is to create banknotes that can be authenticated with a publicly known algorithm which can't be counterfeited by adversaries. The first public-key quantum money scheme was proposed by Aaronson[Aar09]. While this scheme has already been broken, it does a good job of illustrating the concept of public-key quantum mondecoheredey.

A stabilizer state is a pure state that can be obtained by applying CNOT, H and $\frac{\pi}{4}$-phase gates to $|0\rangle^n$. To generate a banknote,

1. Prepare l stabilizer states $|C_1\rangle ... |C_l\rangle$
2. For each state, produce m stabilizer measurements, $E_{i1} ... E_{im}$.
3. With probability ε, E_{ij} is generated such that $|C_i\rangle$ is a $+1$ eigenstate of E_{ij} such that $E_{ij}|C\rangle = |C\rangle$. Otherwise, produce E_{ij} as a tensor produce of n random Pauli operators with a random phase: $E_{ij} = (-1)^b P_1 \otimes ... \otimes P_n$.

Each banknote will consist of the stabilizer states, the measurements, and a signature of a classical description of the measurements. To authenticate a banknote, randomly perform a measurement for each state and accept if the majority of measurements return +1. With probability $1-\varepsilon$, E_{ij} was produced randomly so there's a 50% chance of returning +1. With probability ε, $|C_i\rangle$ is a +1 eigenstate of E_{ij} so it always returns +1. Therefore, the probability of a measurement returning +1 is $\frac{1}{2}+\frac{\varepsilon}{2}$.
This scheme is predicated on the conjecture that it is hard to produce additional states $|C\rangle$ that have a non-negligible probability of returning +1. Unfortunately, this quantum money scheme has already been broken[LAF+09].

3.4 A Construction Relative to a Quantum Oracle

While Aaronson's first construction was broken, he did provide promising evidence of the possibility of a public-key quantum money scheme by proving that such a scheme exists relative to a quantum oracle. A quantum oracle is a logical construct that can evaluate a quantum circuit in $O(1)$ time. For example, if U were a quantum oracle, then $U(|x\rangle|0\rangle) = |x\rangle|f(x)\rangle$. Essentially, the theorem states that given the defined quantum oracle, it is possible to implement a publicly-key money scheme.

The oracle used in the proof, U is defined as follows:

U maps the state $|0\rangle|s\rangle$ to $|0\rangle|s\rangle|e_s\rangle|\psi_s\rangle$, where $s \in \{0,1\}^n$ is a secret key, $e_s \in \{0,1\}^{3n}$ is a public key and $|\psi_s\rangle$ is a n-qubit pure state. Both are uniformly randomly chosen, but fixed given input s. This allows the bank to use U to produce banknotes, $|\psi_s\rangle$ and serial numbers e_s.

U also maps $|1\rangle|e_s\rangle|\psi_s\rangle$ to $|1\rangle|e_s\rangle|\psi_s\rangle|1\rangle$ and $|1\rangle|e_s\rangle|\phi\rangle$ to $|1\rangle|e\rangle|\phi\rangle|0\rangle$ for any $|\phi\rangle$ orthogonal to $|\psi_s\rangle$ or if $e \neq e_s$ This allows users to validate banknotes by inputting $|1\rangle|e_s\rangle|\psi_s\rangle$ and measuring the last qubit of the output.

This proof is predicated on what Aaronson calls the "Complexity-Theoretic No-Cloning Theorem". It is used to show that in order to counterfeit a banknote or copy-protection state, the attacker would need to make an exponential number of queries to the quantum oracle. Therefore, it can not be efficiently accomplished.

4 Current Research Areas

Currently, the central research goal of quantum money is to provide a construction of public-key quantum money, as laid out by Aaronson, along with a security proof. Unfortunately, this has yet to be achieved. In this section, we detail how certain problems and mathematical concepts are used to construct modern public-key quantum money schemes. While each scheme is predicated on the hardness assumption of a problem, the same can be said for classical public-key encryption schemes. For example, the Decisional Diffie-Hellman Assumption has no proof but is still widely used in many classical encryption schemes. Therefore, the lack of a security proof for the following schemes should not affect their credibility.

In this section, we will discuss two significant proposals for public-key quantum money. First, we will discuss Zhandry's recent proposal of his *quantum lightning* scheme. Next, we will provide an

overview of work done by Shor et al on public-key quantum money based on problems in knot theory. Finally, we will give a brief overview of an upcoming paper by Shor on quantum-money from lattices, based on publicly available lectures that he has given about this project. This is by no means an exhaustive selection of current research in quantum money. Another notable accomplishment not investigated here was Daniel Kane's quantum money scheme based on modular forms [Kan18].

4.1 Quantum Lightning

4.1.1 What is Quantum Lightning?

The idea of using quantum lightning to construct public-key money was proposed by Zhandry in 2019. Quantum lightning is the concept of generating random states, analogous to the randomness of where real lightning strikes. Furthermore, each state generated would have a serial number, such that an adversary trying to counterfeit a banknote would be practically unable to prepare a second state with the same serial number.

A quantum lightning scheme consists of a setup procedure, $SetupQL$, which takes a security parameter, λ, as input and samples 2 poly-time quantum algorithms: a $STORM$ algorithm and a VER algorithm.

$STORM$ takes a security parameter, λ, input and samples a quantum state called a bolt: $|\lightning\rangle$

VER takes a quantum states, $|\lightning\rangle$ as input and either accepts or rejects. If it accepts, then it also outputs a serial number, s, corresponding to that bolt.

To ensure correctness, VER must accept every bolt generated by $STORM$, and must consistently output the same serial number when run on the same $|\lightning\rangle$ multiple times.
To ensure security, an adversary should not be able to counterfeit a bolt. In more formal terms, given the $STORM$ and VER algorithms, an adversary should not be able to produce two, potentially entangled, bolts, $|\lightning_0\rangle$, $|\lightning_1\rangle$, such that VER outputs the same serial number for both with non-negligible probability.

4.1.2 Public-Key Signature Scheme

Given a secure and correct quantum lightning scheme, it may seem intuitive that a public-key money scheme follows: use the bolts as banknotes and VER as the authentication algorithm. However, VER accepts all bolts generated by $STORM$, so an adversary could produce a counterfeit banknote by simply running $STORM$.

To rectify this, we need a public-key signature scheme secure against quantum chosen-message attacks. In [AC12], Aaronson describes one such scheme relative to an oracle. The signature scheme consists of the functions: $KeyGen$, $Sign$, and $Auth$. $KeyGen$ takes random bits as input and outputs a public-private key pair. $Sign$ takes a key and bitstring as inputs, and outputs a signature of that bitstring. $Auth$ takes a key, bitstring and signature as input, and either accepts or rejects. For correctness, any signature signed with one key can be authenticated with its paired key.

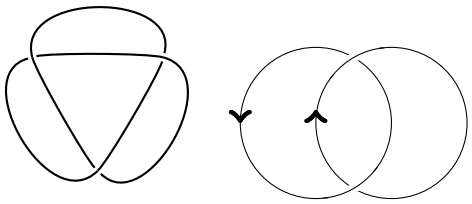

Figure 1: A knot diagram (left) and an oriented link diagram (right)

4.1.3 Public-Key Quantum Money Construction

The public-key money scheme is as follows:
Create a public-private key pair with $KeyGen$, k_{priv} and k_{pub}, for the bank.

To create a banknote, run $STORM$ to obtain $|\lightning\rangle$ and run VER to obtain s. Then, run $\sigma = Sign(k_{priv}, s)$. The quantum money state is $|\lightning\rangle$ and its serial number is (s, σ)

To verify a banknote, accept if both $Auth(k_{pub}, \sigma, s)$ accepts and $VER(|\lightning\rangle) = s$.

Intuitively, this is secure since in order to counterfeit a banknote, an adversary would either have to forge the bank's signature on a random bolt that it generated, or generate a second bolt with the same serial number, which would break the security of the quantum lightning scheme.

4.1.4 Construction Of Quantum Lightning

While quantum lightning seems promising, its construction, proposed by Zhandry, is quite complicated. This report will detail the basic mathematical concepts that Zhandry uses.

Consider a collision resistant hash function, H, on a superposition on all possible inputs, $|sup_x\rangle$: $H(|sup_x\rangle|0\rangle) = |sup_x\rangle|sup_y\rangle$. By measuring $|sup_y\rangle$, we can obtain $|y\rangle$ and a superposition over all the inputs, $|\psi_y\rangle$, corresponding to $|y\rangle$. We can interpret y as the serial number and $|\psi_y\rangle$ as the quantum state. To verify, apply H to the quantum state and output the measurement. However, an adversary can easily forge two bolts with the same serial number, y, by outputting 2 identical input states, $|x\rangle$. To prevent this, Zhandry uses a class of hash functions called "non-collapse-binding" hash functions, which allows for a mechanism to distinguish between states $|x\rangle$ and $|\psi_y\rangle$.

4.2 Quantum Money From Knots

We will now provide an overview of the public-key quantum money scheme proposed by Peter Shor et al in 2012, based on presumably intractable computational problems arising in knot theory [FGH$^+$12a]. The concepts used in Shor's construction bear some similarities to those used in quantum lightning.

4.2.1 Knots, Links, Diagrams and Polynomials

To avoid having to use too many concepts from topology, we can simply think of a *knot* as an embedding of a closed loop in three dimensional space. We will now define some of the terminology we will need for the quantum money scheme. Our definitions will be minimal, but for more information, the reader should consult [Ada94]. A *knot diagram* is obtained by projecting a knot into two dimensions, and by distinguishing, at those points where the knot crosses over itself, the near and far parts of the knot. A *link* is made from two or more knots which are intertwined. An *oriented link* is a link in which each component is associated with one of the two possible directions along its loop. Such links can also be visualized using diagrams, as in Figure 1.

Two links are equivalent if they can be deformed from one into another without having to cut them. If two links are equivalent, the diagram of one can be transformed exactly into the diagram of the other using a sequence of *Reidemeister moves*. A Reidemeister move is one of three types of simple, local manipulations on a link diagram (again, see [Ada94]) for a description of the moves. An *invariant* is a function of links (or knots) which has equivalent values for equivalent links. To show that a function is a knot invariant, you simply need to show that it is preserved under the Reidemeister moves. *Polynomials of links* (or knots) are a powerful class of invariants, analogous to characteristic polynomials in linear algebra.

One of the most well-known knot polynomials is the Alexander polynomial. We denote the Alexander polynomial in a variable x of a link diagram L by $\Delta_L(x)$. A simple algorithm to compute the Alexander polynomial of a particular link diagram is provided in Appendix A. The usefulness of this polynomial in this scheme, as we will see, is that equivalent knots always have the same Alexander polynomial. However, note that the converse is not true; if two knots have the same Alexander polynomial, they are not necessarily equivalent.

Note further that the algorithm for computing the Alexander polynomial of a knot can be converted into a polynomial-time quantum algorithm which can operate on *superpositions* of links and results in *superpositions* of polynomials.

The scheme in [FGH+12a] uses a particular encoding of link diagrams which we should discuss. A *planar grid diagram* of a link is a $d \times d$ grid in which we inscribe d X's and d O's. The diagram must have exactly one X and one O in each cell, and only one (or zero) symbols per cell. In each row, we make a horizontal arrow from the O to the X (either pointing left or right) and in each column a vertical arrow from the X to the O (up or down). Vertical arrows always "pass over" horizontal arrows. An illustration of a planar grid diagram is shown in Figure 2.

We omit the result here, but it is a theorem that every oriented link can be represented by a planar grid diagram, and conversely every grid diagram determines an oriented link. We can consequently derive a set of *grid moves* on planar grid diagrams which allow us to define the equivalence class of grid diagrams representing a particular oriented link (and thus its equivalent links).

A grid diagram is completely determined by two permutations σ_X, σ_O from the symmetric group S_d, where the X's and O's are found in the cells $(i, \sigma_X(i))$ and $(i, \sigma_O(i)$. Of course, d must be at least 2, and we require the permutations be *disjoint*, in the sense that for any i, we have $\sigma_X(i) \neq \sigma_O(i))$. Conversely, any two disjoint permutations on d elements specifies a d-dimensional planar grid diagram. The number of disjoint pairs of permutations on d elements is $(d!) \cdot (!d)$, where $!d$, called the subfactorial of d, is the number of permutations on d elements without fixed points. The subfactorial of d is the nearest integer value to the quotient $\frac{d!}{e}$, where e is euler's constant.

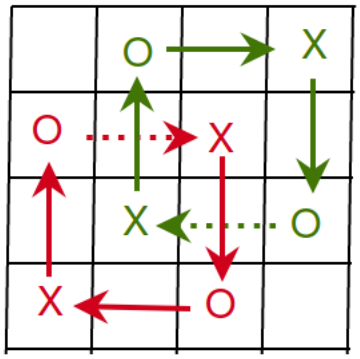

Figure 2: A basic grid diagram corresponding to an oriented link of two (trivial) knots.

4.2.2 Minting Knot-Money

We will now describe the proposed procedure to generate pieces of quantum money.

Fix an integer security parameter $\mathcal{D}$. Define a Hilbert space whose basis vectors represent (presumably via a binary encoding) planar grid diagrams $|G\rangle = |\sigma_X\rangle \otimes |\sigma_O\rangle$, where each grid diagram G is determined by a pair of disjoint permutations σ_X and σ_O. The dimensionality of each G, denoted $d(G)$, is the size of of the permutations which determine G. Assume the largest possible grid diagrams in the space are of dimensionality $\mathcal{D}$. Recall that each G determines an oriented link.

In [FGH$^+$12b], the authors specify a probability distribution which will act as a weight on a superposition of grid diagrams. The specifics of the distribution are important for the security of the protocol, but we will omit these details for now, and refer to this distribution as $\rho(G)$, where G is a grid diagram. The definition of ρ also takes into account the security parameter $\mathcal{D}$, which should be large. Following a procedure built on previous results in [GR02] for generating quantum states from certain classes of probability distributions, we can construct the initial state

$$|init\rangle = \frac{1}{\sqrt{N}} \sum_G \rho(G)|G\rangle$$

(At some point until now, we need to check that each σ_X and σ_O are in fact disjoint, which can be checked efficiently. They are disjoint with probability approximately $\frac{1}{e}$, so repeat the procedure until they are). Here, N is an appropriate normalizing constant. Intuitively, $|init\rangle$ is a weighted superposition of all possible grid diagrams G (within our dimensionality bounds).

It is now possible to take the superposition of all grid diagrams given in $|init\rangle$ and compute into an ancilla register a superposition of Alexander polynomials, following the algorithm described above. Suppose we measure this ancilla register and read the encoding of the coefficients of the Alexander polynomial as the number p. By measuring, we collapse the entire system into the quantum state

$$|\$_p\rangle = \frac{1}{\sqrt{N}} \sum_{G \mid \Delta_G = p} \rho(G)|G\rangle$$

where Δ_G is the sequence of coefficients of the Alexander polynomial of the grid diagram G. Again, we assume an appropriate N to normalize the state. In other words, $|\$_p\rangle$ is a weighted superposition

of grid diagrams G with the same Alexander polynomial determined by p. The quantum money produced by the mint is thus the pair $(|\$_p\rangle, p)$. The "public key" in this scheme is the serial number p, which we assume is published by the mint. This is analogous to the use of a hash function in quantum lightning, where p is the output of the hash function and $|\$_p\rangle$ is the superposition of inputs.

4.2.3 Verifying Knot-Money

There is an efficient algorithm to verify that a pair $(|\psi\rangle, p)$ is a genuine piece of quantum money without damaging the state. The verification algorithm basically proceeds as follows.

1. First, verify that $|\psi\rangle$ is actually a superposition of encodings of valid grid diagrams. This is simple enough to do efficiently. If it is verified, proceed, otherwise the money is obviously not genuine.

2. Compute and measure the (quantum) Alexander polynomial of $|\psi\rangle$. If pair is a genuine piece of quantum money, the measurement will yield p, since every grid diagram in the superposition will have the same Alexander polynomial. If the result is p, proceed, otherwise, reject.

3. The final component of the verification algorithm is a Markov Chain which verifies that $|\psi\rangle$ is the correctly weighted superposition of grid diagrams, with respect to the minting process. We refer to the original work [FGH$^+$12b] for the details of the Markov Chain. In brief, the procedure iteratively verifies that the superposition of grid diagrams is stable under the actions of valid grid moves. A short, but slightly more detailed summary of its behaviour is provided in Appendix B.

4.2.4 Security of Knot Money

It is conjectured (but not certain) that it is very difficult to forge money which can pass the above verification procedure. In particular, it is presumed to be very difficult to forge money which will satisfy the third step. Suppose we want to fool the verifier into accepting a state as genuine. We can assume we know a number p which corresponds to a genuine piece of quantum money, because in a public-key scheme these numbers are presumed to be published by the mint. We can even assume we know a link diagram G whose Alexander polynomial is p. To reliably fool the verifier, we would need now need generate a superposition of all grid diagrams G' equivalent to G' (or sufficiently many given arbitrarily large $\mathcal{D}$). This is assumed to be a very, very difficult problem for diagrams of even mediocre complexity. The knot recognition problem, of checking whether two diagrams G and G' represent the same link, is not even known to be in NP. Furthermore, even if we are provided with an oracle which tells us whether two diagrams G and G' are equivalent, it is not known how to construct a sequence of moves which transform G to G', which would be required to fool step 3 of the verification procedure. Since its publication, this scheme has not demonstrated any feasible security vulnerabilities. If the reader is able to fool the verification procedure, they should immediately notify the nearest available topologist, because this would imply the most remarkable breakthrough in knot theory in at least the last half century.

4.3 Quantum Money From Lattices

In this section, we will discuss the ideas behind Peter Shor's current work on quantum money from lattices. At the time of writing (December of 2020), this work is not yet published, so we cannot provide many details of the scheme, nor can we make a firm judgement on its security. However, in the last few months Shor has given several lectures on this project which are publicly available. Thus, in this section, we will be summarizing the contents from these talks, which the reader can find at [Sho20a] [Sho20c] [Sho20b]. Of course, these notes can only be a high-level approximation to Shor's work, but we believe we have summarized the concepts behind it. Any misinterpretations of his lectures are our own fault. We hope that this high-level summary will provide the reader with background and intuition so that they will be better able to assess Shor's work when it is published.

Preliminaries

In this context, *lattice* means a discrete subspace of the vector space $\mathbb{R}^n$. Given a set of basis vectors $B \subset \mathbb{R}^n$, we can define the lattice generated by B, denoted $L(B)$, as the set of all possible integer linear combinations of the basis vectors. Formally,

$$L(B) = \left\{ \sum_{i=1}^{n} \alpha_i b_i \mid \alpha_i \in \mathbb{Z}, b_i \in B \right\}$$

Given a lattice L, define the dual lattice $L^\perp$ as the set of vectors which are orthogonal to each of the lattice vectors in L.

One of the lattice problems which has been most closely studied by the cryptographic community is the Closest Vector Problem (CVP). Given a basis B and a vector $\mathbf{v}$, find a point on the lattice which is closest to $\mathbf{v}$ with respect to some norm. The particular case of the CVP where $\mathbf{v} = \mathbf{0}$ is called the Shortest Vector Problem (SVP).

A slight generalization of the CVP is called the *Bounded Distance Decoding* problem (BDD). Given a vector $\mathbf{x}$ which is close to a lattice point $\mathbf{v}$, the goal is to find $\mathbf{v}$. This problem has a polynomial-time algorithm. Here, "close" is defined as within $\alpha\lambda_1$, where α is a scalar and λ_1 is the length of the shortest vector in the lattice.

Given a ball $\mathcal{B}$ around a point $\mathbf{x}$, lattice points $\mathbf{v}$ in $\mathcal{B}$ can be sampled with probability proportional to a Gaussian distribution. If the standard deviation of that distribution is exponentially larger than the shortest basis for the lattice, one can use a quantum computer to create a superposition of lattice points in $\mathcal{B}$ in polynomial time. This superposition has the form

$$\frac{1}{N} \sum_{\mathbf{v} \in L} \exp\Big(\frac{-(\mathbf{v}-\mathbf{x})^2}{4\sigma^2}\Big) |\mathbf{v}\rangle$$

where N is a normalizing constant.

Eldar and Shor [ES16] have constructed a Quantum Fourier Transform operating on lattice vectors. More specifically, they operate on vectors in a subclass of lattices based on hypercubes, but this

is not relevant for our discussion. The function takes a Gaussian superposition of vectors around the origin of a lattice, and outputs a Gaussian superposition of vectors around each of the points in the dual lattice. The magnitude of the input Gaussian superposition is inversely proportional to Gaussian superposition around each of the points in the dual lattice. For instance, if one imagines the original superposition as a large ball around the origin, the output will be a very small ball around each of the lattice points in the dual lattice.

Creating Lattice Money

The money itself will be a Gaussian superposition of lattice vectors, forming a small ball around a dual lattice vector $\mathbf{w}$.

First, create a large Gaussian superposition around the origin L. Perform the quantum analog of the BDD algorithm, and the resulting measurement will yield a small Gaussian superposition around a vector $\mathbf{w}$ in $L^{\perp}$.

Shor argues that this quantum money is unable be forged, as follows. If we suppose there is a procedure to copy such a money state, we would effectively be left with two Gaussian distributions around the same lattice vector $\mathbf{w}$. Sampling from each of these two distributions yield two lattice vectors both close to $\mathbf{w}$. Their difference, also a lattice vector, should be close to 0 (since the Gaussian distribution is small), so this hypothetical procedure would effectively solve SVP.

In reality, this scheme just described is the "naive" version of quantum lattice money. Shor has pointed out that there doesn't seem to be a way to distinguish between a state representing a single lattice vector near $\mathbf{w}$ and a Gaussian superposition of such vectors. Hence, a counterfeiting agent could measure simply measure one lattice vector from a Gaussian superposition and forge that state. As a solution, Shor proposes using two copies of the Gaussian superposition around the same vector $\mathbf{w}$ instead of one in each piece of quantum lattice money, although the details for this argument are not yet available. Again, the presumed difficulty of counterfeiting this kind of lattice money relies on the difficulty of SVP using randomly sampled vectors from each copy. Shor qualifies that while a blind, brute-force style attack isn't feasible, they have yet to rigorously argue that no clever attack strategies exist.

5 Conclusion

This report has detailed the evolution of the concept of quantum money from its inception to modern day research. While promising evidence has been presented by Aaronson of the existence of a secure public-key quantum money scheme, the actual construction of such a scheme remains an open problem. The attacks breaking Wiesner's and Aaronson's early schemes illustrate the difficulty of this problem. The current central research question of quantum money is to develop a security proof for a public-key quantum money scheme. Currently, all proposed schemes are based on problems assumed to be hard. However, the same can be said for classical public-key cryptography as well, so the lack of a security proof does not discount a scheme's credibility. Furthermore, motivation of the research in this area extends past practical applications, as it also yields insight into the limits of quantum information processing.

Appendix

A
Computing the Alexander Polynomial

Take any oriented link diagram with α crossings, and observe that it divides the plane into $\alpha + 2$ regions. This can be derived from Euler's characteristic formula $V - E + F = 2$. Let us denote these regions as $r_1, r_2, ..r_{\alpha+2}$.

A diagram L will determine a matrix M of size $\alpha \times (\alpha + 2)$, as follows. Due to the orientations of the links, each of the α crossings will be in one of two forms, which we can call a left-hand crossing or a right-hand crossing, depicted in Figure 3. At each crossing, associate the four regions divided by that crossing with the variables A, B, C and D according to the ordering depicted in Figure 3, and record the equation

$$xA - xB + C - D = 0$$

with the variables A, B, C, D each replaced with the associated r_i. This system of equations can be represented in matrix form by

$$M \begin{bmatrix} r_1 \\ r_2 \\ \cdots \\ r_{\alpha+2} \end{bmatrix} = 0$$

where M is an $\alpha \times \alpha + 2$ integer matrix, and we will have $M_{ij} \in \{0,\ \pm 1,\ \pm x, (\pm 1 \pm x)\}$. Turn M into a square matrix M^* of dimension α by deleting two columns corresponding to any two regions which are adjacent in the original link diagram. Now, $\det(M^*)$ will be some polynomial in x. The Alexander polynomial $\Delta_L(x)$ is obtained by dividing $\det(M^*)$ by a factor $\pm x^k$ such that the term of lowest degree is a positive constant.

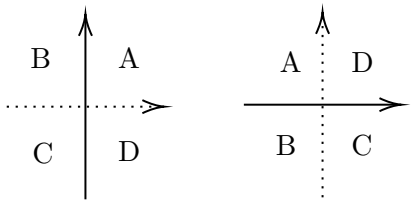

Figure 3: Pattern for labelling regions

B
Verifying Knot Money (Addendum)

1. Define a unitary operator U acting on a grid state $|G\rangle$ and an ancilla register as

$$U(G \otimes |0\rangle) = |G\rangle \frac{1}{\sqrt{\rho(G)}} \sum_{i=1}^{\rho(G)} |i\rangle$$

Take the claimed money state $|\psi\rangle$ and produce the state

$$\begin{aligned} |\psi'\rangle &= U(|\psi\rangle|0\rangle) \\ &= \frac{1}{\sqrt{N}} \sum_{G \mid \Delta(G)=p} \sum_{i=1}^{\rho(G)} |G\rangle \otimes |i\rangle \end{aligned}$$

2. Let S be the set of possible grid diagram moves, for grids of dimensionality at most $\mathcal{D}$. For each $s \in S$, write a permutation matrix P_s encoding the behaviour under s of all grid diagrams of dimension at most D. Since permutation matrices are always unitary, we can generate a unitary operator

$$V = \sum_{s \in S} P_s \otimes |s\rangle\langle s|$$

The crux of the verification process is the fact that equivalent grid diagrams are preserved under the grid moves. Take the state ψ' and generate the state

$$|\psi''\rangle = |\psi'\rangle \otimes \sum_{s \in S} \frac{1}{|\sqrt{s}|} |s\rangle$$

The action of $|\Psi\rangle$ under V is shown to be

$$\begin{aligned} V|\Psi\rangle &= \sum_{s \in S} \frac{1}{\sqrt{|S|}} \big(P_s |\psi'\rangle\big) \otimes |s\rangle \\ &= \sum_{s \in S} \frac{1}{\sqrt{|S|}} |\psi'\rangle \otimes |s\rangle \qquad \text{(By invariance under grid moves)} \end{aligned}$$

Now, measure the projector

$$\mathcal{P} = I \otimes I \otimes \Big(\sum_{s,s' \in S} \frac{1}{S} |s\rangle\langle s'| \Big)$$

If $|\Psi\rangle$ is a component of genuine money, the measurement will be $+1$ with certainty, the state $|\psi''\rangle$ is invariant under every $s \in S$.

3. Repeat step (b) some $r = poly(D)$ times. If we always get a $+1$ eigenvector from the projector $\mathcal{P}$, we can accept the money.

C
Additional Detail on an Attack on Weisner's Quantum Money

To create a counterfeit banknote, for each qubit in a banknote, do the following:

1. Start with a probe qubit $|0\rangle$
2. Rotate it by ϵ, then apply a CNOT gate to the i^{th} qubit of the banknote with the probe qubit as the control bit. Have it verified by the bank.
3. Repeat step 2 $\frac{\pi}{2\epsilon}$ times for the i^{th} qubit.
4. Measure the probe qubit in the standard basis. If it's $|0\rangle$, then the banknote qubit is in the standard basis. If it's a $|1\rangle$, then the banknote qubit is in the $|+\rangle$, $|-\rangle$ basis
5. Measure the banknot qubit with the correct basis. We now know M_i and N_i

If the i^{th} banknote qubit is in the standard basis, then the chance of failing a verification is proportional to ϵ^2. If the i^{th} banknote qubit is in the $|+\rangle$, $|-\rangle$ basis, then there is no chance of failing the verification.

We will refer the probe qubit $|0\rangle$ rotated counter-clockwise by ϵ degrees as $|\epsilon\rangle = cos(\epsilon)|0\rangle + sin(\epsilon)|1\rangle$. There are 4 possible states that a banknote qubit, $|B\rangle$, can be in.

1. Case 1: $|B\rangle = |+\rangle$. Applying a CNOT gate with $|\epsilon\rangle$ as the control qubit does nothing, since the NOT of $|+\rangle$ is itself. Therefore, verification will always succeed and after each iteration, the probe qubit will be rotated by ϵ until it reaches $|1\rangle$.
2. Case 2:$|B\rangle = |-\rangle$. Applying a NOT gate to $|-\rangle$ gives $-|-\rangle$, so the outcome of a CNOT on $(cos(\epsilon)|0\rangle + sin(\epsilon)|1\rangle)|-\rangle$ would be $(cos(\epsilon)|0\rangle - sin(\epsilon)|1\rangle)|-\rangle$, which is equivalent to $|-\epsilon\rangle|-\rangle$. So a verification will always succeed, but the probe qubit will be rotated clockwise after each iteration until it reaches $|-1\rangle$.
3. Case 3:$|B\rangle = |0\rangle$. Applying the CNOT would give us: $(cos(\epsilon)|00\rangle + sin(\epsilon)|11\rangle)$. When verifying this qubit, there is a $sin^2(\epsilon) \approx \epsilon^2$ chance of failing. However, if it doesn't fail, then the probe qubit will remain at $|0\rangle$ for all iterations.
4. Case 4: $|B\rangle = |1\rangle$. Applying the CNOT would give us: $(cos(\epsilon)|01\rangle + sin(\epsilon)|10\rangle)$. The analysis is the same as Case 3. The probe qubit will remain at $|0\rangle$ if the verification succeeds.

By measuring the probe qubit at the end of $\frac{\pi}{2\epsilon}$ iterations, we can deduce whether the banknote qubit was in the standard basis or the $|+\rangle$, $|-\rangle$ basis.

References

[Aar09] Scott Aaronson. Quantum copy-protection and quantum money. *2009 24th Annual IEEE Conference on Computational Complexity*, Jul 2009.

[AC12] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 41–60, 03 2012.

[Ada94] Colin C Adams. *The knot book*. American Mathematical Soc., 1994.

[BZBW15] Charles Bennett, Djabeur Zekrifa, Seth Breidbard, and Stephen Wiesner. Quantum cryptography, or unforgeable subway tokens. In *Advances in Cryptology: Proceedings of Crypto '82*, pages 267–275, 01 2015.

[ES16] Lior Eldar and Peter W Shor. An efficient quantum algorithm for a variant of the closest lattice-vector problem. *arXiv preprint arXiv:1611.06999*, 2016.

[FGH+12a] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter Shor. Quantum money from knots. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 276–289, 2012.

[FGH+12b] Edward Farhi, David Gosset, Avinatan Hassidim, Andrew Lutomirski, and Peter Shor. Quantum money from knots. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 276–289, New York, NY, USA, 2012. Association for Computing Machinery.

[GR02] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002.

[Kan18] Daniel M. Kane. Quantum money from modular forms, 2018.

[LAF+09] Andrew Lutomirski, Scott Aaronson, Edward Farhi, David Gosset, Avinatan Hassidim, Jonathan Kelner, and Peter Shor. Breaking and making quantum money: toward a new quantum cryptographic protocol. 12 2009.

[NSBU16] D. Nagaj, Or Sattath, Aharon Brodutch, and D. Unruh. An adaptive attack on wiesner's quantum money. *Quantum Information & Computation*, 16:1048–1070, 09 2016.

[Sho20a] Peter Shor. *Quantum Money based on Lattices (Lecture at the Simons Institute)*, 2020 (accessed December 1, 2020).

[Sho20b] Peter Shor. *Quantum Money (Google Quantum Summer Symposion 2020)*, 2020 (accessed December 1, 2020).

[Sho20c] Peter Shor. *Quantum Money presented by Peter Shor (Lecture at Illinois Quantum)*, 2020 (accessed December 1, 2020).

[Wie83] Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, January 1983.