An Introduction to Quantum Money

Niall Wingham (20228078 – CS467)

April 13, 2013

Abstract

The goal of this paper is to provide an overview of the field of quantum money, from its invention to current results. We discuss the motivation and history of quantum money, describe the terminology and techniques used to reason about quantum money, and present several proposed schemes for quantum money. Finally, we show two promising implementations based on these schemes and investigate their security.

1 Introduction

The *No-Cloning Theorem* states that it impossible to make perfect replicas of arbitrary quantum states. It is a fundamental result, following from the linearity of quantum mechanics, and it highlights an important distinction: while it is easy to copy classical information, it is impossible to do so for quantum information. This raises interesting possibilities for cryptography. Are there tasks which are impossible using classical information, but which can be accomplished using quantum information? One such task is to create better kinds of money.

1.1 Motivation

Consider the kinds of money we currently use. Some are *physical* currencies like coins and bills, or even raw commodities like gold and diamonds. They have two disadvantages: first, they are possible to forge;[1] and second, they are conspicuous to carry in large amounts. Other types of currencies are *digital*, like debit and credit cards, or PayPal accounts. Because digital information can be copied freely, transactions with these currencies must be verified by some third-party authority. This leads to two new disadvantages: first, they generally require an online connection to use; and second, they are not anonymous.

An ideal currency would combine all the benefits of these two kinds of currencies with none of the defects. It would be unforgeable, inconspicuous, anonymous, and verifiable by anyone. This is precisely the objective of quantum money, and current results suggest it is possible. To be fair, we are still far from able to maintain coherent quantum states for long periods of time outside laboratory conditions, which we would need to do for real quantum money. But that is no reason to delay development of the theory!

[1] Many include security features, but at best these make them *uneconomical* to forge.

1.2 Challenges

Providing a method for anyone to verify quantum money also means providing counterfeiters a powerful tool with which to forge money. If nothing else, a counterfeiter can run a random brute force search against a verifier, though this would succeed with exponentially small probability and is not a serious concern. More worrying is that the counterfeiter will be able to exploit some underlying structure in the verifier to succeed in forging money in polynomial time.

A second challenge in the field of quantum money is that it is very hard to reason about security results. The best that can be said of most current proposals is that a few people tried to break them and have not yet succeeded. This is far from the standard we would like to hold ourselves to.

1.3 Outline

Our paper proceeds as follows. In Section 2, we give a history of early work in quantum money. In Section 3, we outline modern terminology and important theorems used to reason about quantum money. In Section 4, we describe three frameworks for quantum money: these are abstract schemes that rely on black boxes or oracles, but which are still useful as templates. Finally, in Section 5, we investigate the two most promising concrete implementations of quantum money.

2 History

The problem of quantum money was introduced by Wiesner in 1969 [12].[2] He observed that if a bank prepared a piece of money as one of the states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$, it would be able to verify the state by measuring it in either the computational or Hadamard basis. To remember which basis to use and what result it expected, the bank would keep a secret database and distribute each piece of money with a serial number s linking to its record in the database. An adversary trying to copy the money would have to guess which basis to measure it in and could only succeed with probability 3/4: half the time they would use the correct basis and make a perfect copy, and half the time they would use the wrong basis and make a copy that would pass verification with probability 1/2. By using many such qubits for a single piece of money, the bank could make its money arbitrarily hard to copy. This was the first example of a private key scheme, though it required the bank to store a database that grew linearly with the size of its money supply.

Figure 1: Wiesner's Quantum Money

$$\left|\$\right\rangle = \left|s\right\rangle \left|\updownarrow \nearrow \nwarrow \leftrightarrow \leftrightarrow \nwarrow \leftrightarrow \nearrow \nwarrow \nwarrow \updownarrow \nearrow \updownarrow \leftrightarrow \nearrow \nwarrow \updownarrow \leftrightarrow \updownarrow \nearrow \right\rangle$$

Bennett, Brassard, Breidbart, and Wiesner continued this work in [5]. For the private key scheme, they replaced the database with a secret key and pseudorandom function. They also

[2] Despite being a remarkable paper, the field of quantum information did not yet exist, and it remained unpublished for over a decade.

developed a public key scheme that was based on the difficulty of factoring large numbers (though we now know this is not hard for quantum adversaries).

It was later found (e.g. by [4] and [7]) that Wiesner and BBBW's schemes suffered from a serious flaw that allowed counterfeiters to copy money in linear time. It uses the fact that their qubits are independent, and that successful measurements project the qubits onto a valid state. Given a valid piece of money, a counterfeiter can swap out the first qubit with a new one in a random orientation and verify the altered bill, trying multiple times if necessary until it is accepted (it will pass with probability 1/2). Now he has an entirely valid bill, and a copy of the first qubit for a valid bill. The counterfeiter proceeds in this manner along the chain of qubits until he has copied the entire piece of money.

These schemes laid an important foundation: both their construction and their flaws, though relatively simple, are illustrative of modern quantum money. However, research in this area lay dormant for several decades. It was not until circa 2005 that the question of quantum money was revisited, this time by researchers at MIT and the University of Waterloo.

3 Preliminaries

One of the first tasks for these researchers was to formalize terminology and techniques for reasoning about quantum money. Here we introduce vocabulary that will help us describe quantum money schemes, and theorems that will help us prove results about them. This high-level overview should also help to understand the original papers in which these schemes are presented.

3.1 Terminology

There is not yet consistent terminology in the literature on quantum money. For the sake of discussing various papers together, we attempt to provide some general definitions and apply them to each one, even where it is anachronistic. Most come from Aaronson, who formalized them in [2] and [3], but we take e.g. collision-freedom from Lutomirski et al. in [8] and anonymity from Mosca and Stebila in [10].

Quantum Money A piece of quantum money $|\$\rangle = |s\rangle\,|\$_s\rangle$ consists of a serial number $|s\rangle$ and a corresponding quantum state $|\$_s\rangle$.

Schemes A quantum money scheme $\mathcal{S}$ consists of two polynomial-time algorithms: a producer P and a verifier V. $P\,|0\rangle = |\$\rangle$ creates new pieces of money, and $V\,|s\rangle\,|\psi\rangle = \mathit{Accept}$ or Reject validates alleged pieces of money. An issuing bank may also digitally sign its money to distinguish it from other banks or counterfeiters who implement the same scheme.

Private Key In a private key scheme, some secret is needed in order to run V. Typically, this secret is known only by the issuing bank. We are less interested in private key schemes because offline verifiability is an important goal of quantum money.

Public Key In a public key scheme, anyone can run V. Public key schemes may be computationally secure, but they cannot be information-theoretically secure. With access to V, a counterfeiter can always run a brute force search for valid pieces of money.

Completeness For all valid pieces of money $|\$\rangle$, we say a scheme has *completeness error* ε if $V|\$\rangle = Reject$ with probability ε. If $\varepsilon = 0$, we say the scheme is *complete*.

Soundness For any counterfeiting algorithm C, which is allowed as input several valid pieces of money $\{|\$_0\rangle, ..., |\$_k\rangle\}$, we say a scheme has *soundness error* δ if C can produce a new state $|\$_{k+1}\rangle$ such that $V|\$_{k+1}\rangle = Accept$ with probability δ. If $\delta = 0$, we say the scheme is *sound*.[3]

Collision-Freedom A scheme is *collision-free* if P cannot produce two identical pieces of money. In such a scheme, rather than digitally signing its money, an issuing bank could simply publish a list of the serial numbers for the money it has produced.[4]

Anonymity A scheme is *anonymous* if every piece of money produced by P is identical. In this type of scheme, no piece of money can be tracked. This is the opposite of a collision-free scheme.

3.2 Techniques

Aaronson has also made significant contributions to formalizing the security of quantum money schemes. Indeed, most other published schemes rely on his techniques to prove their security. Here we present four of his important results as theorems and describe their meaning and application to quantum money.

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

This result, given as a lemma in [1], says that if we know enough about a quantum state to predict its measurement with high confidence, then measuring it will damage it only slightly. It is used to prove that valid pieces of quantum money can be verified and re-used many times provided the scheme has negligible completeness error.

Theorem 2 (Quantum-Secure Signature Schemes)**.** *If there exists a (classical) one-way function f secure against quantum attack, then there also exists a digital signature scheme secure against quantum chosen-message attacks.*

[3] A counterfeiter can be allowed at most polynomially many pieces of money, or else an attack using quantum state tomography is possible.

[4] Using this kind of verification also has the side effect, pointed out in [8], of making it impossible for an issuing bank to secretly inflate the money supply by creating two pieces of money with the same serial number. Of course, a bank could still secretly deflate the money supply by publishing more serial numbers than it had actually produced.

This theorem is given in [3]. It suggests that although many current digital signatures (e.g. those based on RSA) are not secure against quantum adversaries, it is possible to construct new signatures which are. It allows us to talk about digital signatures in general with some degree confidence, for example, when we said in Section 3.1 that an issuing bank should sign the money it produces.

Theorem 3 (Complexity-Theoretic No-Cloning)**.** *Let $|\psi\rangle$ be an n-qubit pure state. Suppose we are given the state $|\psi\rangle^{\otimes k}$ for some $k \geq 1$, as well as an oracle U_ψ such that $U_\psi |\psi\rangle = -|\psi\rangle$ and $U_\psi |\phi\rangle = |\phi\rangle$ for all $|\phi\rangle$ orthogonal to $|\psi\rangle$. Then for all $\ell > k$, to prepare ℓ registers $\rho_1, ..., \rho_\ell$ such that*

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

we need

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

queries to U_ψ.

This theorem is claimed in [2] though only a sketch of its proof was offered until [3]. If a counterfeiter attempts to copy one or more pieces of money without using the scheme's verifier, then the no-cloning theorem applies. Alternately, if a counterfeiter attempts to find a valid piece of money by consulting the scheme's verifier, but without using any existing pieces of money, then the lower bound for quantum search applies. However, a real counterfeiter will take advantage of both. This theorem shows he still cannot do much better than guessing, as long as there is no underlying structure in the verification algorithm that he can exploit. It is used to prove that oracle-based quantum money schemes have negligible soundness error.

Theorem 4 (Inner-Product Adversary Method)**.** *For some quantum algorithm Q and classical oracle U, let $|\Psi_t^U\rangle$ be Q's state after t queries to the oracle. Given:*

(i) A set of quantum oracles $\mathcal{O}$ with every $U \in \mathcal{O}$ having a subspace $S_U \leq \mathcal{H}^{2^n}$ such that $U|\psi\rangle = -|\psi\rangle$ for all $|\psi\rangle \in S_U$, and $U|\phi\rangle = |\phi\rangle$ for all $|\phi\rangle \in S_U^\perp$

(ii) A symmetric irreflexive binary relation $\mathcal{R}$ with every $U \in \mathcal{O}$ having at least one $V \in \mathcal{O}$ such that $(U, V) \in \mathcal{R}$

Suppose that for every $U \in \mathcal{O}$ and all $|\phi\rangle \in S_U^\perp$, we can show:[5]

$$\mathop{\mathrm{E}}_{V:(U,V)\in\mathcal{R}}\left[F\left(|\phi\rangle, S_V\right)^2\right] \leq \varepsilon$$

Furthermore, suppose that for all $(U, V) \in \mathcal{R}$, when Q starts we have $[\langle\Psi_0^U | \Psi_0^V\rangle] \geq c$, and when Q finishes we have $[\langle\Psi_0^U | \Psi_0^V\rangle] \leq d$. Then Q must make at least $\Omega((c-d)/\varepsilon)$ oracle queries.

[5]Where E is the expected value and F is fidelity. Fidelity is a measure of the closeness of a mixed state to another mixed state or a subspace, but it is beyond the scope of this paper to formally introduce it; for more details see [3], § 2.2.

This theorem is given in [3] and is used to prove the security of Aaronson's oracle-based quantum money scheme as well as the Complexity-Theoretic No-Cloning Theorem. The idea behind it is to give an upper-bound for how well a quantum algorithm can distinguish pairs of oracles as the result of a single query.

Ideally, we would be able to show that the inner product $[\langle \Psi_0^U \,|\, \Psi_0^V \rangle]$ decreases by at most ε with each query, in which case it easily follows that we need to make $\Omega((c-d)/\varepsilon)$ queries to reduce the inner product from c to d. However, it is not generally possible to show this for quantum money schemes since Q also has information beyond the oracles, in the form of valid pieces of money. Instead, we have to examine how much the inner product is *expected* to decrease with each query, for some carefully designed distribution of oracle pairs (U, V).

4 Frameworks

We are now ready to examine three framework schemes for public key quantum money. We call them frameworks because each scheme is dependent on one or more black box oracles that recognize special states by flipping their phase (that is, the recognized state is a -1 eigenstate of the oracle).

Recall that in Section 3.1, we saw that some scheme properties conflict with each other. In particular, a scheme cannot be both *anonymous* and *collision-free*. In physical currencies, coins are indistinguishable from each other while bills are printed with serial numbers. Therefore, we say anonymous schemes make *quantum coins* while non-anonymous schemes make *quantum bills*.

It is beyond the scope of this paper to prove it, but all of the following schemes are complete and with negligible soundness error. The proof for each scheme can be found in the paper in which the scheme was introduced, and should be somewhat familiar based on the techniques introduced in Section 3.2.

4.1 Quantum Coins

The only published framework for public key quantum coins comes from Chapter 8 of Stebila's thesis [11], also presented in [9] and [10]. To create coins of some denomination d, the issuing bank randomly selects a secret pure state $|\psi_d\rangle \in \mathcal{H}^{2^n}$ and uses an oracle $U_{\psi_d} = I - 2\,|\psi_d\rangle\,\langle\psi_d|$. Note that this oracle recognizes exactly $|\psi_d\rangle$ by flipping its phase, similar to an oracle for amplitude amplification. The quantum coin scheme $\mathcal{S}_d = (P, V)$ is then formally defined by:

$$P\,|0\rangle = |d\rangle\,|\psi_d\rangle = |\$_d\rangle$$

$$V\,|\$_d\rangle = \begin{cases} \mathit{Accept} & \text{if } U_{\psi_d}\,|\$_d\rangle = -1 \\ \mathit{Reject} & \text{otherwise} \end{cases}$$

Note that the "serial number" slot of the money is effectively ignored; we include it for the sake of having a consistent definition of quantum money.

4.2 Quantum Bills

There are two recent frameworks of interest for public key quantum bills, one proposed by Lutomirski et al. in [8], and the other by Aaronson and Christiano in [3]. The former scheme is collision-free but requires an oracle which seems more difficult to implement. Both of these schemes involve states that are uniform superpositions over sets. For notational simplicity, we define

$$|S\rangle := \frac{1}{\sqrt{|S|}} \sum_{s \in S} |s\rangle$$

4.2.1 Quantum Money by Postselection

The idea behind the first scheme is to create a uniform superposition over a large, randomly selected set, with a serial number identifying which set was chosen. To verify a piece of alleged money, an oracle checks that the state is a uniform superposition over the correct set.

Let S be the set of n-bit strings, and let L be a "labelling function" which assigns a label ℓ to each $s \in S$. Finally, let $S_\ell = \{s : L(s) = \ell\}$ be the set of elements for a given label. L should be constructed so that there are exponentially many different labels, and so each S_ℓ has exponentially many elements; it should also have as little structure as possible. Finally, we require oracles U_ℓ which recognize states in a uniform superposition over S_ℓ.

The production algorithm P is defined by the following three steps. First, it generates a uniform superposition over S. Then, it applies the labelling function to the superposition to obtain an augmented superposition. Finally, it measures the value of L to collapse the state to a superposition over a particular S_ℓ. A piece of money $|\$_\ell\rangle$ looks like $|\ell\rangle |S_\ell\rangle$.

Figure 2: Production by Postselection

$$\frac{1}{\sqrt{|S|}} \sum_{s \in S} |0\rangle |s\rangle \quad \Rightarrow \quad \frac{1}{\sqrt{|S|}} \sum_{s \in S} |L(s)\rangle |s\rangle \quad \Rightarrow \quad \frac{1}{\sqrt{|S_\ell|}} \sum_{s \in S_\ell} |\ell\rangle |s\rangle$$

Given an alleged piece of money $|\$\rangle = |\ell\rangle |\psi\rangle$, the verification algorithm V is defined by the following two steps. First, it checks that $L|\psi\rangle = \ell$. Second, it checks that $U_\ell |\psi\rangle = -|\psi\rangle$. If both of tests pass, $P|\$\rangle = Accept$; otherwise $P|\$\rangle = Reject$.

Lutomirski et al. propose that the oracles U_ℓ could be implemented using a special Markov matrix which rapidly mixes any distribution of strings S_ℓ to the uniform distribution $|S_\ell\rangle$, but does not mix between strings with different labels. Each update in the Markov chain would consist of a random choice between N unitary update rules. Then, any valid $|\psi_\ell\rangle$ is a $+1$ eigenstate of M, and M^r approximates a projection onto $|S_\ell\rangle$.

4.2.2 Quantum Money from Hidden Subspaces

The idea behind the second scheme is to create superpositions over subspaces $A \leq \mathbb{F}_2^n$, with a serial number identifying the subspace for each bill. To verify a piece of alleged money, an oracle checks that a state is a uniform distribution over the correct subspace.

Let the generator G be a function which takes a random seed r and outputs a list of linearly independent generators $\langle A_r \rangle = \{x_1, ..., x_{n/2}\}$ for $|A_r\rangle$, and an identifying serial

number s_r. We also require oracles U_A and $U_{A^\perp}$ which recognize elements of their respective subspaces (i.e. $U_A|x\rangle = -|x\rangle$ if $|x\rangle \in A$ and $|x\rangle$ otherwise). Note that we can transform $|A\rangle$ to $|A^\perp\rangle$ and back again by applying a Hadamard gate to each qubit. Also note that we can construct a projector P_A onto the set of basis states in $|A\rangle$ by initializing a control bit $|c\rangle = |+\rangle$, applying a controlled-U_A, measuring $|c\rangle$ in the Hadamard basis, and postselecting on the outcome $|-\rangle$.

Now we can formally define the scheme. The production algorithm takes the following three steps. First, it chooses a random seed r. Second, it generates $G(r) = (s_r, \langle A_r\rangle)$. Finally, it outputs the money state $|\$_r\rangle = |s_r\rangle|A_r\rangle$.

Given an alleged piece of money $|\$\rangle = |s_r\rangle|\psi\rangle$, the verification algorithm is defined by $V_r|\psi\rangle = H^{\otimes n}P_{A_r^\perp}H^{\otimes n}P_{A_r}|\psi\rangle$. It projects $|\psi\rangle$ onto $|A\rangle$, transforms it to the Hadamard basis and projects it onto $|A^\perp\rangle$, and finally returns it to the standard basis. The verifier only needs to perform two tests: this is an elegant parallel to Wiesner's original idea of quantum money based on complementary observables.

5 Implementations

Of course, the real challenge is to implement ones of these scheme without using an oracle. The issuing bank must be able to distribute some obfuscated program P that can take the place of its framework's oracle without revealing any kind of structure that allows counterfeiters to treat it as more than a black box. Some previously proposed implementations, e.g. [2] have been shown to be insecure. Here we examine two implementations for public key quantum bills that have not yet been broken. The first implements the postselection scheme, and the second implements the hidden subspaces scheme.

There has not yet been a proposed implementation for quantum coins. It seems that it is even harder to obfuscate P in this case because the pieces of money it is operating on are all identical. This gives an even larger advantage to the counterfeiter.

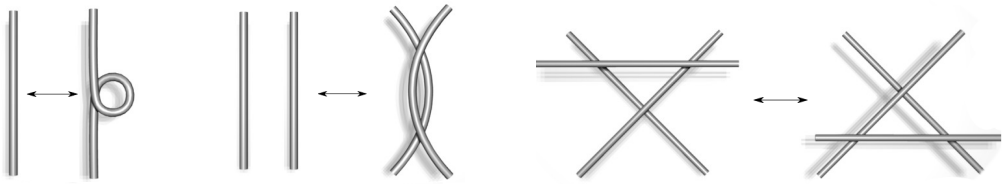
5.1 Equivalent Knots

Recall from section 4.2.1 that to implement the postselection scheme, we need to define a set S, labelling function L, and a carefully designed Markov matrix M that mixes quickly between elements sharing a label but not between others. In [6], Farhi et al. build on their work in [8] to implement such a scheme based on the difficulty of determining if two arbitrary knot diagrams are equivalent.

We will briefly introduce some knot theory. A *knot* can be thought of as a loop of string in three dimensional space, though we usually draw knot diagrams in two dimensions. A *link* is a collection of knots, which may be tangled together. If each knot in the link has a direction, it called an *ordered link*. Two links are equivalent if one can be "rearranged" to be the same as the other without cutting any of the knots. A sequence of *Reidemeister moves* is sufficient to describe any valid rearrangement. These moves are: twisting or untwisting a string; moving one string over another; and moving a string completely over or under a crossing. Finally, the *Alexander polynomial* is a polynomial which can be computed in polynomial time from an arbitrary oriented link diagram. Equivalent knots have the same

Figure 3: A Knot Diagram

Figure 4: The Reidemeister Moves



Alexander polynomial, so it is invariant under the Reidemeister moves. (However, two knots with the same Alexander polynomial are not necessarily equivalent.)

Now we can describe their construction of a postselection scheme. Let the starting set S be the set of oriented links of a certain size encoded as strings in $\mathbb{F}_2^n$. Let the labelling function L be the Alexander polynomial, which divides S into exponentially many groups S_ℓ, themselves having exponentially many elements, as required. Finally, let the Markov chain M be one which at each step selects a random part of the link and applies a random Reidemeister move to it.[6]

If a counterfeiter were to fully measure a piece of money $|\$\rangle = |\ell\rangle\,|S_\ell\rangle$, they would get a single link diagram $d \in S_\ell$ having Alexander polynomial ℓ. From there they could reconstruct a superposition of all link diagrams equivalent to d, which, though not equal to $|S_\ell\rangle$, would still be a +1 eigenstate of M^r and so would pass the verification method. However, this would amount to solving the knot equivalence problem, which is currently thought to be hard even for quantum computers. Farhi et al. also discuss other possible lines of attack and conclude that their scheme is probably secure, issuing the challenge, "Forge it if you can!"

In [3], Aaronson and Christiano point out some limitations of this implementation. As shown above, states other than the ones produced by the issuing bank can pass the scheme's verification test, and simply finding out exactly which states it will accept remains an open problem that may require advances in knot theory to solve. The implementation hasn't been broken yet, but at the same time, it is hard to say much about whether it will remain unbroken in the future.

5.2 Multivariate Polynomials

Recall from Section 4.2.2 that to implement the hidden subspaces scheme, we need a program P that can recognize membership in A and $A^\perp \leq \mathbb{F}_2^n$. In [3], Aaronson and Christiano suggest a strategy based on multivariate polynomial cryptography.

[6]There are other technical details, e.g. to make sure the initial encodings are valid and that a link diagram doesn't grow too large during a Reidemeister move, but this captures the spirit of the implementation.

Given a set $P = \{p_1, ..., p_m\}$ of multivariate polynomials from $\mathbb{F}_2^n$ to $\mathbb{F}_2$, it is hard to find a point $v \in \mathbb{F}_2$ at which P vanishes; that is, at which $p(v) = 0$ for all $p \in P$. However, given a candidate v, it is easy to verify that a collection of polynomials does vanish at v. Therefore, for each piece of money $|\$\rangle$ the issuing bank can provide two sets P_A and $P_{A^\perp}$ of polynomials which vanish on each point of A and $A^\perp$, respectively. These sets will play the part of the oracles.

Let's see how to construct such a P_A. First, note that we are evaluating a polynomial $p(x_1, ..., x_n)$ on a point $v \in \mathbb{F}_n$ by viewing v as an n-tuple $(v_1, ..., v_n)$. We can also restrict ourselves to *multilinear* polynomials since $x^2 = x$ in $\mathbb{F}_2$. Now consider the set $\mathcal{J}_{d,A}$ of multilinear polynomials having degree d which vanish on A. If the basis for A is $X_A = \{x_1, ..., x_{n/2}\}$ then, every monomial of $p \in \mathcal{J}_{d,A}$ must intersect $X_{A^\perp} = \{x_{n/2+1}, ..., x_n\}$. Importantly, this means we don't need to go through the expensive process of enumerating the entire set $\mathcal{J}_{d,A}$. If all we want is to choose a random element $p \in \mathcal{J}_{d,A}$, we can simply iterate over the $O(n^d)$ possible monomials and include ones that intersect $X_{A^\perp}$ with probability $1/2$. Selecting some number β of polynomials in this manner will give us our set P_A in $O(\beta n^d)$ time.

Aaronson and Christiano also provide a method to further disguise the polynomials, by adding random noise, so only a fraction of the polynomials in P_A vanish for a given point $v \in A$. They show that the implementation remains complete even with noise added. This gives a potential "second chance" if the noiseless version is shown to be insecure. Finally, they compare their hardness assumptions to well-studied and similar assumptions in multivariate polynomial cryptography, and conclude that the implementation is very likely secure when $d \geq 3$.

Other researchers have yet to critique this implementation; [3] is a very recent article, and no subsequent work in quantum money has yet been published.

6 Conclusion

We have shown two promising implementations of quantum money based on two different frameworks. It is important to understand that even if an implementation is shown to be insecure, its underlying framework continues to be valuable. For example, any method for distinguishing between states in a subspace A and its complement $A^\perp$ would be sufficient to make money based on hidden subspaces, regardless of the security of multivariate polynomials. This is encouraging.

Some of the open problems shift weight onto classical mathematics: we probably need to understand more about knots to prove stronger results about Farhi et al.'s proposal. But there is also plenty of work to be done in the space of quantum information. We expect that new frameworks and implementations based on different assumptions will continue to be created, widening the range of possibilities for a formal security result.

Then all that's left is to figure out how to actually build the darn things!

References

[1] Scott Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1:1–28, 2005.

[2] Scott Aaronson. Quantum copy-protection and quantum money. In *Computational Complexity, 2009. CCC'09. 24th Annual IEEE Conference on*, pages 229–242. IEEE, 2009.

[3] Scott Aaronson and Paul Christiano. Quantum money from hidden subspaces. *Theory of Computing*, 9:349–401, 2013.

[4] Scott Aaronson, Edward Farhi, David Gosset, Avinatan Hassidim, Jonathan Kelner, and Andrew Lutomirski. Quantum money. *Commun. ACM*, 55(8):84–92, August 2012.

[5] Charles H Bennett, Gilles Brassard, Seth Breidbart, and Stephen Wiesner. Quantum cryptography, or unforgeable subway tokens. In *Advances in Cryptology–Proceedings of Crypto*, volume 82, pages 267–275, 1983.

[6] 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. ACM, 2012.

[7] Andrew Lutomirski. An online attack against wiesner's quantum money. *arXiv preprint arXiv:1010.0256*, 2010.

[8] 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. *arXiv preprint arXiv:0912.3825*, 2009.

[9] Michele Mosca and Douglas Stebila. Uncloneable quantum money. In *Canadian Quantum Information Students Conference (CQISC) 2006*, 2006.

[10] Michele Mosca and Douglas Stebila. Quantum coins. *Error-Correcting Codes, Finite Geometries and Cryptography*, 523:35–47, 2010.

[11] Douglas Stebila. *Classical authenticated key exchange and quantum cryptography*. PhD thesis, University of Waterloo, 2009.

[12] Stephen Wiesner. Conjugate coding. *ACM Sigact News*, 15(1):78–88, 1983.