DOI:10.1145/2240236.2240258

Imagine money you can carry and spend without a trace.

BY SCOTT AARONSON, EDWARD FARHI, DAVID GOSSET, AVINATAN HASSIDIM, JONATHAN KELNER, AND ANDREW LUTOMIRSKI

Quantum Money

EVERYBODY LIKES MONEY. It is very easy to spend. With cash and credit cards, you can buy what you want when you want it. So why are quantum computing theorists trying to rethink money?

There are a few things we all take for granted about money. We trust credit card companies to keep our transactions private and to send the right amount of money to the right place quickly. When we use paper money, we are used to the fact that we have to carry it physically with us, and we accept the risk of occasional counterfeiting.

Today, we use two basic kinds of money. First, there is the kind we carry around—coins, bank notes, poker chips, and precious metals. These are objects that are made by a mint or dug out of the ground. It is easy to verify that money is valid. You can look for the security features on paper money, you can feel coins in your hand, and, if you really know what you are doing, you can assay precious metals. All of these kinds of physical money can be counterfeited, though—if you have the right equipment, you can print paper money, stamp out your own coins, or make alloys or platings that look a lot like solid precious metals. Some subway passes and copy cards are also examples of physical money—they contain small computer chips or magnetic strips that are actually worth a certain number of subway rides or copies. But these tend to be even easier to counterfeit.[22] In theory, any physical money can be counterfeited by just using the same production process as the one used to make the original.

The other kind of money is the kind that you entrust to someone else. Think bank accounts and credit lines. You can carry these around with you in the form of checks, credit cards, and debit cards—portable devices that let you instruct your bank to move money on your behalf. Unlike physical money, there is no point in copying your own credit card (it would not double the amount of money in your bank). With a credit card, you can carry as much value as you want without weighing down your pockets and you can send money across the globe nearly instantaneously. But credit cards have disadvantages: every time you pay someone, you need to tell your bank whom to send money to. This leaves a paper trail and does not work if your connection to your bank is down.

Neither of these kinds of money is ideal. For example, imagine that you are going to Las Vegas on a business trip and you want to play some high-stakes games at night. You might feel conspicuous carrying a fat wad of cash around the strip. If you use a credit

» key insights

- **Any digital good can be perfectly copied. This is a major headache for software companies (and for the entertainment industry), and is the reason that digital cash does not exist.**
- **The quantum mechanical "no-cloning' theorem means that in principle it is possible to design quantum systems that cannot be copied. Several recent works propose to use such systems for digital money.**
- **Further research may lead to a new form of digital rights management.**

$= \alpha^2|0\rangle|0\rangle + \alpha\beta|0\rangle|1\rangle + \alpha\beta|1\rangle|0\rangle + \beta^2|1\rangle|1\rangle$
$\frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$
$C(|0\rangle|0\rangle) = |0\rangle|0\rangle$

ILLUSTRATION BY SPOOKY POOKA AT DÉBUT ART

card, your significant other (not to mention anyone else who gets access to your bank statements) will know exactly how much money you gambled. What you really want is some kind of money that you can spend without leaving a trace and that you can carry as much of as you want without weighing down your pockets.

This kind of money would be digital: you could transmit it and fit as much of it as you want on some small handheld computer. It would be self-contained, so you could pay someone without any third party being involved. And it would be cryptographically secure: attackers could never produce a counterfeit bill that passes as real money even with extraordinary resources at their disposal.

The reason we do not have this kind of money today is not for lack of trying. Any digital piece of information that can be sent over a communication channel can be copied. This makes digital money seem impossible: if you had one hundred dollars on your computer, you could back up your computer, spend the money, restore your computer from the backup, and spend your money again.

Enter quantum mechanics. Physicists have known for years that if you possess a single quantum object and know nothing about it, then it is fundamentally impossible to copy that object perfectly. This is called the no-cloning theorem, and it gives us hope that *quantum* information could be used as the basis of a better kind of money.

So can we make the idealized money out of quantum mechanical objects rather than classical ones? In the rest of this article, we will survey recent work that has tackled this question. We will introduce the idea of quantum information, and we will talk about a few proposals for quantum money and some of the open problems in the field.

Quantum Mechanics

If you look closely enough, everything is made out of subatomic particles, and these particles obey the laws of quantum mechanics. Quantum mechanical systems store information in a way that is dramatically different from classical (that is, non-quantum) systems.

One of the simplest examples of a quantum system is a single electron. Electrons spin, and their spin can be characterized by a three-dimensional vector.[a] This vector, like any three-dimensional vector, has three components, S_x, S_y, and S_z. It is possible to do an experiment to measure the vertical component S_z of an electron's spin, but if you do the experiment, you will discover something strange: S_z can only take on two values, +1 and −1.

[a] The vector represents the angular momentum of the electron, but its physical interpretation is not important for this discussion.

Once you have measured S_z, you can measure it again and you will get the same answer as you got the first time around. S_x and S_y work the same way. So far, you might have thought that each component of the electron's spin stores one bit of information.

But if you try to measure more than one of the components, again something strange happens. Take an electron, measure S_z, and suppose that the outcome is +1. Now measure S_x (obtaining either +1 or −1) and then measure S_z again. You would expect to get $S_z = +1$ as before, but if you do this experiment you will get +1 half the time and −1 half the time. Measuring the electron's spin therefore changes the spin state of the electron. Physicists have come to realize that this is not a limitation of their experiments but rather that the universe fundamentally operates this way.

Figure 1. The no-cloning theorem.

Imagine that someone prepares a single qubit in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and gives it to you without telling you what α and β are. Your goal is to copy that qubit. We will call whatever algorithm you use (the supposed "quantum copy machine") *C*. You feed *C* the (unknown) qubit $|\psi\rangle$ and an output qubit that starts in the state $|0\rangle$ and your machine needs to output the original qubit and transform $|0\rangle$ into a copy of $|\psi\rangle$.

You do not know in advance what α and β are, so your copy machine has to work for any values. In particular, your machine needs to work if $\alpha = 1$ and $\beta = 0$, which means

$$C(|0\rangle|0\rangle) = |0\rangle|0\rangle.$$

Similarly, your copy machine needs to work if $\alpha = 0$ and $\beta = 1$, which means

$$C(|1\rangle|0\rangle) = |1\rangle|1\rangle.$$

But quantum mechanics is linear, so any copy machine you could possibly build has to be linear as well. This means that the operator *C* is linear, so we can do some linear algebra:

$$\begin{aligned} C(|\psi\rangle|0\rangle) &= C((\alpha|0\rangle + \beta|1\rangle)|0\rangle) \\ &= \alpha C(|0\rangle|0\rangle) + \beta C(|1\rangle|0\rangle) \\ &= \alpha|0\rangle|0\rangle + \beta|1\rangle|1\rangle. \qquad (1) \end{aligned}$$

Your copy machine was supposed to copy *any* state, so the output should have been

$$\begin{aligned} &(\alpha|0\rangle + \beta|1\rangle)(\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha^2|0\rangle|0\rangle + \alpha\beta|0\rangle|1\rangle \\ &\quad + \alpha\beta|1\rangle|0\rangle + \beta^2|1\rangle|1\rangle \end{aligned}$$

If both α and β are nonzero, then the output (1) of your machine is not correct. This means that your copy machine *C* cannot possibly work.

No matter what encoding you use or how perfect an apparatus you can build, you can only ever reliably encode one bit worth of recoverable classical information in the spin of an electron.[20] Nonetheless, an electron behaves very differently than a classical bit. If we use electron spins instead of classical bits to store information, we can perform tasks that are completely impossible with ordinary computers.

Qubits

An electron's spin is an example of a mathematical object called a qubit. A classical bit can take either of the two values 0 or 1. But a qubit is described mathematically by a normalized state in a two-dimensional complex vector space. We will use notation from physics to denote vectors that represent quantum states, writing a vector named v as $|v\rangle$. We can write any one-qubit state as

$$|q\rangle = \alpha|0\rangle + \beta|1\rangle$$

where the states $|0\rangle$ and $|1\rangle$ form a basis for the 2D vector space and where α and β are complex numbers that satisfy $|\alpha|^2 + |\beta|^2 = 1$. If neither α nor β is zero, then we call the state $|q\rangle$ a *superposition* of $|0\rangle$ and $|1\rangle$ because the qubit $|q\rangle$ is, in a sense, in both states at once.

Just as one qubit can be in the state $|0\rangle$ or $|1\rangle$ or some superposition (linear combination) of both, n qubits can be in any superposition of the states

$$|0\ldots00\rangle, |0\ldots01\rangle, |0\ldots10\rangle,$$
$$|0\ldots11\rangle, \ldots, |1\ldots11\rangle$$

So, an n qubit state is a vector in a 2^n-dimensional space.

The simplest kind of measurement one can perform on a single qubit is one that answers this question: is the qubit in a given state $|r\rangle = a|0\rangle + b|1\rangle$? Let us say our qubit is prepared in the state $|q\rangle$ as above and we make this measurement. Then there are two possible outcomes. We might get the answer YES, in which case the state of the system would change instantaneously from $|q\rangle$ to $|r\rangle$. The probability that this happens is given by the complex inner product squared of the two states in question

$$\Pr[\text{YES}] = |\alpha^{*}a + \beta^{*}b|^2.$$

If on the other hand we obtain the measurement outcome NO, then the state of the system would instantaneously change from $|q\rangle$ to the state $|r^{\perp}\rangle = b^{*}|0\rangle - a^{*}|1\rangle$ that is perpendicular to $|r\rangle$. This happens with probability

$$\Pr[\text{NO}] = |\alpha^{*}b - \beta^{*}a|^2 = 1 - \Pr[\text{YES}].$$

We can use this mathematical framework to explain the measurement statistics of electron spin. We define the states

$$|S_z = +1\rangle = |0\rangle$$
$$|S_z = -1\rangle = |1\rangle;$$

these two states form a basis for a one-qubit vector space. Then

$$|S_x = +1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|S_x = -1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

and

$$|S_y = +1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle)$$
$$|S_y = -1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle).$$

Measuring the spin component S_x is the same as measuring whether the state being tested is $|S_x = +1\rangle$; the outcome YES means $S_x = +1$ and the outcome NO means $S_x = -1$. If the spin started in the $|S_z = +1\rangle$ state then, upon measuring S_x, we will obtain +1 or −1 with equal probability and the state after the measurement would be either $|S_x = +1\rangle$ or $|S_x = -1\rangle$. If we then measure S_z again, we obtain +1 or −1 with equal probability.

Physicists are trying to build devices that can manipulate electrons or other qubits in a manner analogous to the way ordinary computers manipulate bits in their memories. Such a device, if it worked reliably and could store many qubits, would be a functioning quantum computer.

Many computer scientists and physicists believe that if we could build a quantum computer, we could use it to calculate things that would be intractable with classical computers.[2]

Qubits have another strange property: unlike classical bits, they cannot be copied. This is the content of the quantum no-cloning theorem, which says there is no such thing as a perfect quantum copy machine that can copy any quantum state you feed it. (See Figure 1 for the proof in the single qubit case.) There are also limits on how closely you can approximate a quantum copy machine.[7] The no-cloning theorem allows for cryptographic protocols that go beyond the abilities of classical computers. The best known example is quantum key distribution,[4] which allows two parties to communicate privately, using a quantum channel and an authenticated (public) classical channel.

The no-cloning theorem gives us hope that *quantum* information could be used as the basis of a better kind of money.

Quantum Money

The no-cloning theorem means we should not think of qubits the same way we think about bits. One might imagine using a handful of qubits as a form of money. A mint could produce some qubits using some process known only to it, and anyone else, given just those qubits, could not copy them by any means without further information. The no-cloning theorem does not immediately imply secure quantum money is possible; it only says that machines that can copy *all* quantum states are impossible, and a counterfeiter would be content with a machine that only copied quantum states that represented valid quantum money. A counterfeiter could also try to obtain additional information about the quantum money state by using the algorithm that a merchant would use to verify quantum money. By examining concrete schemes for quantum money, we will see how these kinds of attacks can be avoided.

We distinguish two broad categories of quantum money.

In the simpler version, a mint would produce a quantum bill consisting of some number of qubits. Anyone could store the quantum bill and move it around, maybe even trading it for something else. Whenever someone wants to verify the quantum bill is valid (for example, a merchant who is offered a quantum bill as payment), he or she sends the qubits to the mint and the mint checks that they are still in the correct state using some secret process. In this type of scheme, no one other than the mint knows how to verify the money. We call this *private-key quantum money* because the key—that is, the information needed to verify the money—is private to the mint.

The other type of quantum money is *public-key* quantum money. As before, a mint can produce a quantum state and anyone can move it or spend it. Anyone should be able to verify the money themselves without communicating with the mint. Public-key money, if it could be realized, would be the ideal money we discussed earlier.

In the first quantum cryptography paper ever written,[26] Stephen Wiesner described a way to implement private-key quantum money in a provably secure manner. (He wrote the paper in 1969, but it was not published until 1983.) In Wiesner's scheme, each quantum bill is a unique random quantum state,[b] which the mint labels with a serial number. The mint keeps track of the state that corresponds to the serial number of each quantum bill and it can use its knowledge of the state to verify the money.

In 1982, Bennett et al. made the first attempt at public-key quantum money.[5] Their scheme only allowed a piece of money to be spent once (they called their quantum states subway tokens, not bills). In hindsight, their scheme is insecure for two different reasons: first, it is based on an insecure protocol for 1-2 oblivious transfer, and second, it can be broken by anyone who can run Shor's algorithm[23,24] to factor large numbers. (In the early days of quantum cryptography, there was no reason to suspect either of these weaknesses. Shor's algorithm[23] and the general attack[14] on oblivious transfer were not known until more than a decade later.)

Surprisingly, the next paper about quantum money did not appear until 2003 when Tokunaga et al.[25] attempted to modify Wiesner's scheme to prevent the mint from

[b] In fact, this is the random state later used in the BB84 protocol[4] for quantum key distribution.

tracking each individual bill as it is used. They achieved this by requiring that the owner of a bill modify the bill before allowing the bank to verify it. The modification is done in a special way so that valid bills remain valid but are otherwise randomized so that the bank cannot tell them apart. This scheme has the significant disadvantage that upon discovering a single counterfeit bill, the bank is required to immediately invalidate every bill it has ever issued. In our opinion this scheme therefore has limited practical applicability.

The idea of public-key quantum money gained traction in the years that followed. Aaronson proved a "complexity-theoretic no-cloning theorem,"[1] which showed that even with access to a verifier, a counterfeiter with limited computational resources cannot copy an arbitrary state. Mosca and Stebila proposed[18] the idea of a quantum coin as distinct from a quantum bill—each quantum coin of a given denomination would be identical. Using the complexity-theoretic no-cloning theorem they argued it might be possible to implement a quantum coin protocol but they did not give a concrete implementation. Aaronson[1] proposed the first concrete scheme for public-key quantum money; however, this scheme was shown to be insecure in Lutomirski et al.[16] In the latter paper, the authors suggested the idea of collision-free quantum money. Unlike quantum coins, each collision-free quantum bill has a serial number and nobody, not even the mint, can produce two bills with the same serial number. This feature can be useful to prevent the mint from printing more money than it says it is printing. The mint posts a list of all serial numbers of every quantum bill ever produced, and we can be sure the mint produced at most one bill for each serial number on the list. In a subsequent paper, Farhi et al. proposed a concrete scheme they believed was both collision free and secure against counterfeiting.[11]

Here, we tell you how some of these proposals work.

Wiesner's Quantum Money

Wiesner's original quantum money scheme[26] works as follows. To produce a quantum bill using n qubits, the mint first chooses n one-qubit states randomly drawn from the set $\{|S_z=1\rangle, |S_z=-1\rangle, |S_x=1\rangle, |S_x=-1\rangle\}$. The mint then assigns that state a classical serial number. A piece of quantum money consists of the n qubit state and its serial number. The mint keeps a list of all serial numbers issued as well as a description of which state corresponds to which serial number. When you pay for something with a quantum bill, the merchant sends the quantum state and its serial number back to the mint for verification. The mint looks up the serial number and retrieves the description of the corresponding quantum state. Then the mint verifies the given state is the state that goes with the attached serial number. This kind of money cannot be forged by someone outside the mint. Since a would-be forger has no knowledge of the basis that each qubit was prepared in, the quantum no-cloning theorem says he or she cannot reliably copy the n qubit quantum state (Figure 2).

The main weakness in Wiesner's scheme is that the merchant must communicate with the bank to verify each transaction. So this scheme, although theoretically inspiring and provably secure, would not be much more powerful than credit cards. Wiesner's scheme is a private-key quantum money scheme because the mint must keep a private secret—the complete description of the state—to use for verification.

The resurgence of interest in quantum money is centered around the idea of public-key quantum money.

Challenges in Designing Public-Key Quantum Money

The resurgence of interest in quantum money is centered around the idea of public-key quantum money. As we have discussed, a public-key quantum money scheme would have the following properties.[16]

1. The mint can mint it. That is, there is an efficient algorithm to produce the quantum money state.

2. Anyone can verify it without communicating with the mint. That is, there is an efficient measurement anyone can perform that accepts money produced by the mint with high probability and minimal damage.

3. No one (except possibly the mint) can copy it. That is, no one other than the mint can efficiently pro-

duce states that are accepted by the verifier with better than exponentially small probability.

Why is public-key quantum money so hard to design? The difficulty of developing public-key quantum money arises from the fact that the verification algorithm—which is known to everyone in a public-key scheme—can be used by a would-be forger in an attempt to counterfeit the money.

Wiesner's scheme is provably secure on information-theoretic grounds if it is used properly. In Wiesner's scheme, only the bank has the additional information required to verify a given quantum bill is legitimate and therefore only the bank can copy the money.

It turns out that, if the mint is careless, then even the mere act of verifying bills can allow someone to create counterfeit bills.[15] Recall that in Wiesner's scheme, in every transaction the bill is sent by the merchant back to the mint for verification. If the money is confirmed to be valid, the mint sends back the valid bill to the merchant. What happens if the money is determined by the mint to be counterfeit? If the mint sends back the invalid bill, then a counterfeiter can successfully forge the money.

Let us see how this works. A counterfeiter can start with one good quantum bill, which in Wiesner's scheme is n one-qubit states

$$|\psi\rangle = |\psi_1\rangle|\psi_2\rangle\ldots|\psi_n\rangle$$

along with a serial number the bank uses to verify the state. The counterfeiter can produce a random one-qubit state $|\phi_1\rangle$, and, setting aside the first qubit $|\psi_1\rangle$ of the original bill, he or she then sends the mint the state

$$|\psi'\rangle = |\phi_1\rangle|\psi_2\rangle\ldots|\psi_n\rangle.$$

If the bill $|\psi'\rangle$ turns out to be valid (this happens with probability ½), the mint returns the bill, and in this case the mint's measurement will have changed the state to $|\psi\rangle$. So now the counterfeiter possesses both $|\psi\rangle$ and the original qubit $|\psi_1\rangle$ that was set aside, and so he or she has succeeded in copying the first qubit $|\psi_1\rangle$. On the other hand, if the mint determines the bill $|\psi'\rangle$ is not valid, then the state of the bill after the mint's measurement will be

$$|\psi_1^\perp\rangle|\psi_2\rangle\ldots|\psi_n\rangle$$

where $|\psi_1^\perp\rangle$ is the one-qubit state orthogonal to $|\psi_1\rangle$. Note that the states of qubits 2 through n have not been changed by this process. So the counterfeiter can then throw away $|\psi_1^\perp\rangle$, replace it with a random state, and try again. After an average of two tries, the counterfeiter will have copied the first qubit of the quantum bill. Then the counterfeiter can repeat this whole procedure to copy the second qubit, the third qubit, and so on until all n qubits have been copied.[c]

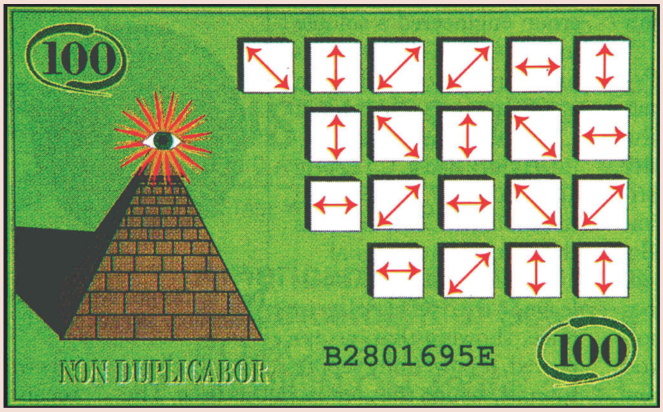
So if the bank sends back quantum states it deems to be invalid quantum money, the whole scheme is unusable. This tells us how *not* to implement Wiesner's scheme in practice. But it also highlights the fact that having access to a verifier that returns the state and a verdict on the validity of the quantum money is in itself a powerful tool a forger can try to exploit, even if the forger cannot look inside the machine that verifies money. This type of attack is particularly applicable to public-key quantum money schemes, in which the verification algorithm is publicly known.

This attack was particularly simple against Wiesner's money because each bill consists of independent (that is, unentangled) qubits. A more general algorithm called quantum state restoration[10] works on entangled states as well: starting with a single valid quantum bill, a counterfeiter can make a sequence of measurements on the state and use the algorithm that verifies the bill to undo the damage caused by measuring the state. So any public-key quantum money scheme must be designed so that the attacker cannot gain enough information to copy the quantum money state by making a reasonable number of measurements on one copy of it. Can we hope to design a public-key quantum money scheme which has this property, or is the access to a verification algorithm already enough information to allow cloning of an arbitrary state? Aaronson answered this question in 2009 with a "complexity-theoretic no-cloning theorem."

The Complexity-Theoretic No-Cloning Theorem

As we mentioned earlier, the standard no-cloning theorem is not good enough to prove secure public-key quantum money is possible, since it does not take into account the counterfeiter can *check* whether a given state is valid quantum money or not. In fact, if a counterfeiter has unlimited time, then it is straightforward to counterfeit public-key quantum money: simply generate a random state and check if that state is valid money. If not, try again. In a secure money scheme, the probability that any attempt succeeds is exponentially small.

Figure 2. Wiesner's quantum money. Source: *Science*, Aug. 7, 1992.



[c] The attack in Lutomirski[15] is different: it is deterministic and twice as fast, but it is less intuitive.

The complexity-theoretic no-cloning theorem[1] says there is no *generic* attack much better than random guessing. What do we mean by a generic attack? Suppose there is a verification machine that checks whether or not a given state $|\phi\rangle$ is equal to a good quantum money state $|\psi\rangle$. The machine takes as input any quantum state $|\phi\rangle$; it outputs 0 if $|\phi\rangle = |\psi\rangle$ and 1 if $|\phi\rangle$ is orthogonal to $|\psi\rangle$. In either case, it also outputs the quantum state is left over after the measurement. Aaronson showed that, as long as that machine is a black box, it can fall into the hands of a counterfeiter without compromising the quantum money scheme. In other words, a counterfeiter with access to some quantum money as well as the verification machine would either need to take the machine apart to figure out how it worked or else use the machine an exponentially large number of times in order to make any more quantum money than he or she started with.

This theorem does not guarantee any particular scheme is secure. For every quantum money scheme that has been proposed, the states $|\psi\rangle$ that are "good" quantum money states are not completely unknown since they come from a restricted set of states generated by the mint's algorithm. If this set of states is small enough then having a "black box" verifier may allow a forger to copy a money state; we have already seen an example of this with Wiesner's scheme. And it might also be possible to design attacks on public-key quantum money that do not use the verifier as a black box. So in order to evaluate any public-key quantum money scheme, we will have to look at the details of the verifier and the set of valid quantum money states that are minted by the bank.

Figure 3. Quantum money from knots.

Figure 4. A knot.

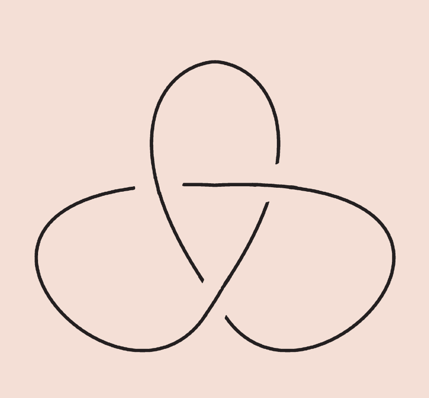
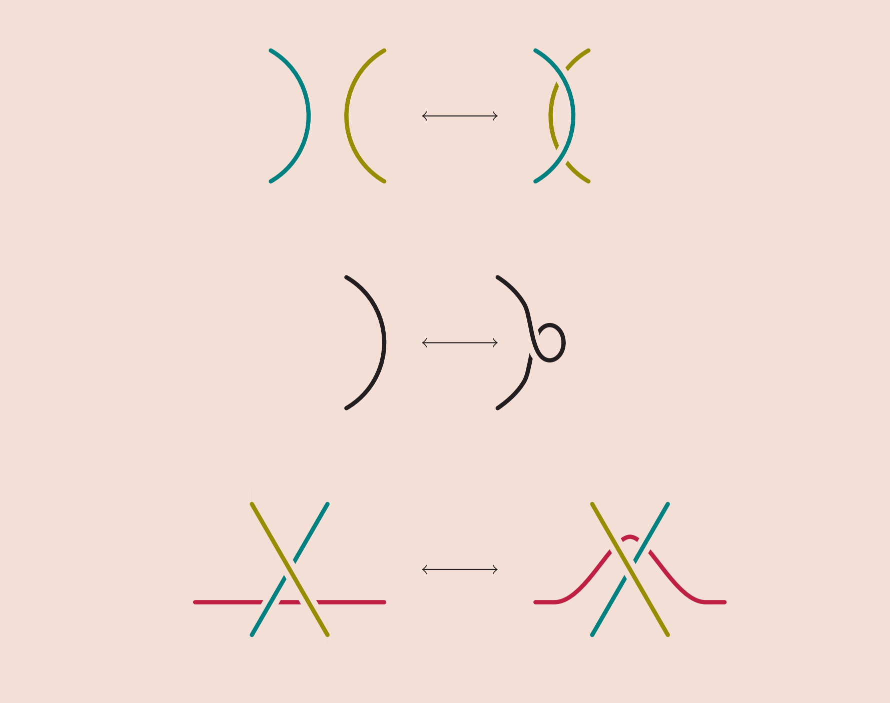


Figure 5. Reidemeister moves.



Quantum Coins

One of the first applications of the complexity-theoretic no-cloning theorem was given by Mosca and Stebila.[18] They showed it might be possible to have public-key quantum money scheme in which every piece of quantum money is identical: they called these *quantum coins*.[18,19]

Quantum coins, like ordinary coins, are all the same with no marks distinguishing each coin. One advantage of quantum coins is they are *anonymous*—no one can tell one coin from another, so it is difficult to keep track of where and when a particular coin was spent.

Mosca and Stebila had two results about quantum coins. They extended the complexity-theoretic no-cloning theorem to quantum coins. If a would-be counterfeiter has access to a machine that verifies quantum coins but cannot look inside that machine, then there is no way to make more coins than he or she started with in any reasonable amount of time. This result gives some hope a public-key quantum coin protocol could be discovered.

Their second result is based on blind quantum computation (introduced by Childs[9] and studied by Broadbent et al.[6]). Blind quantum computation is a protocol whereby a quantum computer with very limited resources (sometimes called a quantum calculator) runs a polynomial size quantum circuit with the help of a server, where the server does not learn anything about the circuit performed (except an upper bound on its size). In the protocol introduced by Mosca and Stebila, the merchant runs an obfuscated verification algorithm from which he or she learns nothing except the final answer: that it is or is not a valid coin. However, this requires (quantum) communication with the bank, and so this quantum coin scheme is a private-key protocol.

To date there is no published concrete proposal for public-key quantum coins.

Public-Key Quantum Money without Secrets

In all of the schemes we have discussed so far, the mint first generates some classical secret and then uses that secret to produce the quantum money. In any scheme like this, if anyone can figure out the secret, then they can use this secret to produce valid quantum money states with the same algorithm that the mint uses. A would-be forger can try to use the (publicly known) verification algorithm along with techniques such as quantum state restoration[10] to try to reverse-engineer the secret.

Lutomirski et al.[16] suggested a different approach to designing quantum money. Imagine a physical process (or a quantum measurement) that can simultaneously generate a random serial number y (drawn from an enormous set of possible serial numbers) and a corresponding quantum state $|\$_y\rangle$. For any given serial number y, a second algorithm would be able to verify some quantum state was indeed $|\$_y\rangle$. A key feature of this scheme is *collision-freedom*: no one can generate more than one copy of $|\$_y\rangle$ for any value of y. (Anyone can generate quantum states corresponding to *different* serial numbers.)

To use these states as money, the mint simply generates a pile of quantum states and corresponding classical serial numbers. The mint then publishes the list of all the serial numbers and the verification algorithm that can be used by anyone to check the validity of a given quantum money state. A quantum state matching a published serial number is valid money; any other state is not. If the mint published an actual list, then anyone could also verify the mint produced no more quantum money than it said it did; as a practical matter, though, the mint would probably use digital signatures instead of a list.

Lutomirski et al. also suggested a way such an algorithm might be designed. Consider a large set S and a function f that assigns each element of S a label. Suppose there is an exponential number of possible labels and an exponential number of elements of S that share each label. Each label corresponds to a serial number, and the state corresponding to the serial number y is a uniform superposition of all of the elements of S that have the label y. Mathematically,

$$|\$_y\rangle = \sum_{x\in S \text{ s.t. } f(x)=y} |x\rangle.$$

One advantage of quantum coins is that they are *anonymous*—no one can tell one coin from another, so it is difficult to keep track of where and when a particular coin was spent.

To produce a quantum money state, the mint first prepares a uniform superposition over all elements of S and measures the label that corresponds to the state. This results in a random label and, like all measurements, changes the state so the new state will always get the same measurement outcome. This means the superposition collapses to exactly those elements of S that have the measured label.

The verification procedure presented by Lutomirski et al. requires anyone who knows some x where $f(x)=y$ find another *random* x' with the same label y, and therefore f must be chosen so this is possible. A merchant who wants to verify a quantum bill first measures the label and confirms it matches the serial number of the bill, and then performs a more complicated quantum measurement to check the state is invariant under the operation that randomizes the elements that share the same label.

Lutomirski et al. conjecture that if f and S are appropriately chosen, then the resulting quantum money will be secure. In that paper, however, they did not describe an appropriate f and S.

Quantum Money from Knots

The only published scheme[11] for public-key quantum money that has not been shown to be insecure is an implementation of collision-free quantum money. In this scheme, the set S is a set of drawings of knots. We will have to take a quick detour into knot theory in order to describe this quantum money (Figure 3).

Most of us have some experience in our day-to-day lives with the basic properties of knots. Mathematicians who study knot theory have formalized these basic properties. For our purposes, a good place to start will be with some definitions. A knot is a mapping of the circle S^1 (like a loop of string) into three-dimensional space. For example, Figure 4 shows a knot.

Usually when we draw a knot, we use a two-dimensional diagram like the one in Figure 4. If we take a knot and then fiddle with it a bit (without cutting the string it is made out of) and then draw it, we might end up with a different diagram. But we would still like to call it the same knot. So the question arises: which pictures represent the same knot? The three modifications to a knot diagram shown in Figure 5 are called the Reidemeister moves. It can easily be seen that by applying these moves you only move between topologically equivalent knots. It is also true (but more difficult to see) that any two diagrams representing the same knot can be mapped into one another using these moves.

There is no known good algorithm to determine whether two knot diagrams represent the same knot; it has only recently been discovered that knot equivalence is decidable.[13] But sometimes there is a way to tell that two diagrams do not represent the same knot; by using a *knot invariant*. A knot invariant is a property of a knot that is the same for all diagrams representing the same knot. If you can find a knot invariant that takes different values for the two diagrams in question, then you can be sure they represent different knots. (The converse of this is not generally true—there can be two different knots that share the same value for a particular knot invariant.) One of the first knot invariants to be discovered is called the Alexander polynomial—any knot has an associated Alexander polynomial, and its coefficients are integers that can be efficiently calculated from any diagram of that knot.

To make quantum money from knots, the set S in the general collision-free scheme is taken to be the set of knot diagrams, and label f associated with each diagram is its Alexander polynomial. Applying a sequence of random Reidemeister moves randomizes among knots with the same diagram, allowing the measurement that verifies the quantum money states. So the mint prepares the superposition over all diagrams and measures the Alexander polynomial's coefficients to make a quantum bill, and a merchant measures the coefficients and verifies the superposition is invariant under the Reidemeister moves.

(The actual scheme is somewhat more complicated because knot diagrams are inconvenient to work with—see Farhi et al.[11] for the technical details.)

While no one has proven that knot money is secure, attempts to break it seem to run into knot theory problems that have no known practical solutions.

What Does the Future Hold for Quantum Money?

Public-key quantum money is one of few quantum protocols that does something that is truly impossible classically, even under cryptographic assumptions. QKD can be used to encrypt information between two parties that did not coordinate keys in advance, but under reasonable security assumptions, lattice based cryptography can perform the same feat.[4,21] Assuming SHA1 is a pseudo random function, one can use it to implement strong coin flipping,[8,17] and encrypted communication channels enable fast Byzantine agreement.[3,12] However, no cryptographic assumption enables a digital analog of cash, as any string of bits that would represent a bill can always be copied.

The idea of some kind of quantum object that only one special entity can produce may have applications beyond being used as money. For example, software companies would like to be able to produce software programs that anyone can use but that no one can copy. Whether this is possible for any useful type of software remains to be seen.

Will a future government replace its currency with quantum money? Maybe. You could use it online to purchase things without transaction fees and without oversight from any third party. You could download your quantum money onto your qPhone (not yet trademarked) and use it to buy things from quantum vending machines. With the advent of quantum money, we hope everybody will like spending money a little bit more.

References

1. Aaronson, S. Quantum copy-protection and quantum money. In *Annual IEEE Conference on Computational Complexity* (2009), 229–242.
2. Bacon, D. and van Dam, W. Recent progress in quantum algorithms. *Commun. ACM 53* (Feb. 2010), 84–93.
3. Ben-Or, M. and Hassidim, A. Fast quantum byzantine agreement. In *STOC* (2005), 481–485.
4. Bennett, C.H. and Brassard, G. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers Systems and Signal Processing* (Bangalore, India, 1984), 175–179.
5. Bennett, C.H., Brassard, G., Breidbart, S. and Wiesner, S. Quantum cryptography, or unforgeable subway tokens. In *Advances in Cryptology—Proceedings of Crypto* (1983), volume 82, 267–275.
6. Broadbent, A., Fitzsimons, J. and Kashefi, E. Universal blind quantum computation. In *FOCS* (2009), 517–526.
7. Bužek, V. and Hillery, M. Quantum copying: Beyond the no-cloning theorem. *Phys. Rev. A 54*, 3 (1996), 1844–1852.
8. Chailloux, A. and Kerenidis, I. Optimal quantum strong coin flipping. In *50th Annual IEEE Symposium on Foundations of Computer Science, 2009 (FOCS'09)* (2009), IEEE, 527–533.
9. Childs, A.M. Secure assisted quantum computation. *Quant. Inform. Comput. 5*, 6 (2005), 456–466.
10. Farhi, E., Gosset, D., Hassidim, A., Lutomirski, A., Nagaj, D., and Shor, P. Quantum state restoration and single-copy tomography for ground states of hamiltonians. *Phys. Rev. Lett. 105*, 19 (Nov. 2010), 190503.
11. Farhi, E., Gosset, D., Hassidim, A., Lutomirski, A. and Shor, P. Quantum money from knots. In *Innovations in Theoretical Computer Science* (2012).
12. Feldman, P. and Micali, S. An optimal probabilistic protocol for synchronous byzantine agreement. *SIAM J. Comput. 26*, 4 (1997), 873–933.
13. Hass, J. Algorithms for recognizing knots and 3-manifolds. *Chaos, Solitons & Fractals 9*(4–5) (1998), 569–581.
14. Lo, H.-K. Insecurity of quantum secure computations. *Phys. Rev. A 56* (Aug. 1997), 1154–1162.
15. Lutomirski, A. An online attack against Wiesner's quantum money. *arXiv:1010.0256*, 2010.
16. Lutomirski, A., Aaronson, S., Farhi, E., Gosset, D., Hassidim, A., Kelner, J. and Shor, P. Breaking and making quantum money: Toward a new quantum cryptographic protocol. In *Innovations in Computer Science* (2010).
17. Mochon, C. Quantum weak coin flipping with arbitrarily small bias. *Arxiv preprint arXiv:0711.4114* (2007).
18. Mosca, M. and Stebila, D. A framework for quantum money. Poster at *Quantum Information Processing (QIP)* (Brisbane, Australia, 2007).
19. Mosca, M. and Stebila, D. Quantum coins. In *Error-Correcting Codes, Finite Geometries, andCryptography*. Contemporary Mathematics, Aiden A. Bruen and David L. Wehlau, eds. volume 523. American Mathematical Society, 2010, 35–47.
20. Nielsen, M.A. and Chuang, I.L. *Quantum Information and Computation*, Cambridge University Press, Cambridge, UK, 2000.
21. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th annual ACM symposium on Theory of computing* (2005), ACM, 84–93.
22. Ryan, R., Anderson, Z. and Chiesa, A. Anatomy of a subway hack. http://tech.mit.edu/V128/N30/subway/Defcon_Presentation.pdf (August 2008).
23. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In *FOCS* (1994), IEEE Computer Society, 124–134.
24. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev. 41*, 2 (1999), 303–332.
25. Tokunaga, Y., Okamoto, T. and Imoto, N. Anonymous quantum cash. In *EQIS* (Aug. 2003).
26. Wiesner, S. Conjugate coding. *SIGACT News 15*, 1 (1983), 78–88.

Scott Aaronson (aaronson@csail.mit.edu) is an associate professor, CSAIL, MIT, Cambridge, MA.

Edward Farhi (farhi@mit.edu) is the Cecil and Ida Green Professor of Physics, and director of the Center for Theoretical Physics, MIT, Cambridge, MA.

David Gosset (dgosset@mit.edu) is a postdoctorate fellow in the Department of Combinatorics and Optimization and the Institute for Quantum Computing, University of Waterloo, Canada.

Avinatan Hassidim (avinatanh@gmail.com) is an assistant professor in the Department of Computer Science, Bar Ilan University, and works at Google Israel. His research is supported by a grant from the German-Israeli Foundation (GIF) for Scientific Research and Development under contract number 1-2322-407.7/2011.

Jonathan Kelner (kelner@mit.edu) is an assistant professor of applied mathematics, and a member of CSAIL, MIT, Cambridge, MA.

Andrew Lutomirski (andy@luto.us) is co-founder of AMA Capital Management LLC, Palo Alto, CA.