Quantum Computing: Foundations to Frontier Spring 2022

Lecture 8 - Quantum Money

Lecturer: Henry Yuen *Scribes: Sastry Garimella*

1 Overview

In the first half of this lecture we discussed about basic ideas behind Quantum Cryptography with some examples. Similar to the case of classical cryptography, we have honest parties trying to communicate securely and adversaries trying to break the security (either privacy or integrity). Quantum Cryptography was classified mainly into two categories, namely Post-Quantum Cryptography and "Fully"-Quantum Cryptography. Post-Quantum Cryptography is when the adversaries have a Quantum computer to break a classical cryptography. This mainly includes finding alternative crypto-systems that are resistant to quantum attacks. Traditionally classical cryptography uses RSA, which assumes hardness of factoring and Post-Quantum Cryptography is being researched using the hardness of lattice problems, mainly shortest vector problem (SVP). "Fully"-Quantum Cryptography is when both the honest parties and the adversaries have quantum computers. Quantum Key Distribution using BB84 [1] (Bennet, Brassard - 1984) is discussed in detail in the lecture.

In the second half of this lecture we discussed another application of Quantum Cryptography, namely Quantum Money. This is a currency scheme in which the currency is difficult or even impossible to counterfeit.

2 Quantum Money

Quantum money is one of the earliest demonstrations of quantum information processing and is an illustration of the interplay between the principles of quantum mechanics and cryptography. It was conceived by Stephen Wiesner in the 1970s, in his paper on "Conjugate Coding" which treats a class of codes made possible by restrictions on measurement related to the uncertainity principle without analogue in communication channels adequately described by classical physics. The idea that he used for quantum money inspired Bennett and Brassard to start to design the BB84 protocol.

One of the important but undesirable aspect of classical money is that, in principle, it is copyable, that is, with endless resources it is possible to make counterfeit money perfectly. Quantum money is a proposal to create a type of currency that is in principle not copyable; this is based on the idea of the non-cloning theorem. However, a non-copyable quantum state itself is not a useful form of currency, since users also need a way to verify that the money state is valid and has a value. Therefore, quantum money has to satisfy two properties:

- Non-clonability
- Verifiability

2.1 Wiesner's Quantum Money Scheme

In Wiesner's quantum money scheme, it assumes a central bank that is in charge of producing and distributing quantum money and a way to verify that a quantum money state comes from the central bank and no one else. The central bank distributes quantum money states of the following form:

$$(s, |\psi_{f(s)}\rangle)$$

which consists of a classical serial number $s \in \{0,1\}^m$ and an n-qubit quantum state $|\psi_{f(s)}\rangle$ that depends on s.

The central bank secretly generates a random function $f : \{0,1\}^m \to \{0,1,+,-\}^n$. This function is picked randomly ahead of time and is fixed once generated.

The quantum money state $|\psi_{f(s)}\rangle$ is a tensor product of n qubits

$$|\psi_{f(s)}\rangle = |f(s)_1\rangle \otimes |f(s)_2\rangle \otimes \cdots \otimes |f(s)_n\rangle$$

Each qubit of $|f(s)\rangle$ looks like a BB84 state,

$$|f(s)_i\rangle \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$$

Since f is picked uniformly at random, the money states are independent of each other.

The bank keeps a giant database of all money states in secret. For every possible s, it generates a money state $(s, |f(s)\rangle)$ and distribute it to the public.

Question: How could one verify the money state?

Let's say that Alice has one such money state $(s, |\psi_{f(s)}\rangle)$ and she goes to Bob for a coffee. Bob has to take the money to bank to verify it. Notice that Bob has no access to the random function f and thus cannot verify the bill by himself. Knowing the serial number and f, the bank computes $f(s)$, looks up in the database all states of $|f(s)\rangle$, and measures each of the qubits using the appropriate bases.

For example, if $n = 2$ and $f(s) = \{1, 4\}$, then $|f(s)\rangle = |0\rangle|-\rangle$. The bank will measure the first qubit in the standard basis and expect to get outcome $|0\rangle$ with probability 1. It will then measure the second qubit in the $\{|+\rangle, |-\rangle\}$ basis and expect an outcome $|-\rangle$ with certainty. If at any moment, one of the measurements yield an unexpected outcome, the bank will report to Bob that the bank note is invalid.

Note that, for a valid money state, this verification process will not destroy the money states at all, because each state will be correct with probability 1. This demonstrates that the system has is verifiable.

Question: Is such a money state non-clonable?

The public has no knowledge of the secret random function f and hence the bases each of the n qubits of the money states are in. A naive counterfeiter will guess for each state which basis to measure the qubit in. If the counterfeiter messes up at least once, he will damage the state. If the counterfeiter guesses all the bases correctly, then he will be able to copy the money state, however,

the probability of it happening is only 2^{-n}. However, even if he damages one state, the fact of using only 2 orthogonal basis gives a probability of 1/2 for the bank to notice this error in the state.

It was shown by Molina, Vidick and Watrous that, given a quantum money note, the best counterfeiting attack succeeds (creating two copies of a bank note from one, where both copies pass the bank's test for validity) with a probability at most $(3/4)^n$ [2]. This comes from the fact that, we are using two orthogonal basis and probability of guessing the correct state of 1/2 and the probability of guessing wrong but the bank not able to confirm it is 1/4 (1/2 for guessing wrong basis by counterfieter and 1/2 for the state to be in the correct alignment when verified by the bank). Thus the total probability for 1 qubit to be conterfieted without the bank able to notice it is 3/4. For large n, the probability that verification passes is extremely low.

Interactive Attacks. Assume that one has a correct bank note and want to figure out what the quantum money states are. Assume that he can verify the states at the bank any times. Each time the bank will report if the verification passes or not and returns the states to the person. One can figure out each state one by one by replacing one correct qubit with a test qubit while keeping the other correct qubits and submit to the bank multiple times. If every time the bank reports that the money state is correct, then with high probability the test qubit is correct. Note that the verification will not damage the other states since they are all correct. By repeating the verification procedure, it is possible to identify all coordinates of $f(s)$ and reproduce the money states as many times as one wants.

There is an easy fix. The bank could put a small fee for multiple verifications and/or limit the number of times the same person could consecutively verify a quantum money state, or it could destroy the bank note if it rejects too many times.

There are some bigger issues with Wiesner's quantum money scheme. One problem is that the bank has to keep track of the exponential large database which is infeasible. This can be solved by using a pseudo-random function. Users limited in computational power cannot distinguish between a pseudo-random function and a uniformly random function,

2.2 Quantum Money with Classical Communication

Another problem with the weisner's scheme is that one has to have a quantum communication with the bank, that is, they need to physically give the bank the note to verify. Gavinsky [3] introduced an alternative scheme in which bills can be authenticated using only classical communication with the bank.

In this scheme the bank sends the user a random challenge $c \in \{0,1\}^n$. An honest user should measure the $i-$th qubit in the standard basis if $c_i = 0$, or in the hadamard basis if $c_i = 1$, and send the measurement outcomes $b \in \{0,1\}^n$ to the bank. The bank then validates the states when the challenge c_i has the same basis as the function $f(s)_i$ and then checks these corresponding outcomes b_i to the expected outcomes of the bank. In this setting, a simple counterfeiting attack is one in which a counterfeiter tries to succeed in two independent authentications with the bank, given access to a single valid bank note.

For the classical verification analogue of Wiesner's quantum money scheme, the optimal simple counterfeiting attack has success probability exactly $(3/4+\sqrt{2}/8)^n$ which is again shown by Molina, Vidick and Watrous [2]. Even in this scheme a comminication with the bank is required which may

not always be desirable and thus a public key quantum money scheme is introduced.

2.3 Public Key Quantum Money Scheme

In public key quantum money scheme (PKQM), there still exists a central bank that generates all bank notes. Besides, the bank also generates one private key k_{priv} and one public key k_{pub}. The public key is published online so every one has access to it, while only the bank has access to the private key.

The bank is distributing money states of the following form:

$$(s, |\psi_{s,k_{priv}}\rangle)$$

which consists of an n-bit classical serial number $s \in \{0,1\}^n$ and a quantum state that depends on the serial number s and the secret private key k_{priv}. The bank is also distributing a verification algorithm

$$Ver(s, |\psi\rangle, k_{priv})$$

that takes a potential serial number, a potential quantum money state and the public key. The algorithm will accept a quantum dollar bill with probability 1 if $|\psi\rangle = |\psi_{s,k_{priv}}\rangle$. Given the verification algorithm, one can verify a quantum money state by himself.

The algorithm also has the property that one cannot generate new money states without knowing the private key. Given the source code of $Ver(s, |\psi\rangle, k_{priv})$ and a quantum money $(s, |\psi_{s,k_{priv}}\rangle)$, there exists no polytime algorithm that can produce two money states that are accepted with greater than exponentially small probability. In order for the non-clonability property to hold, we have to have computational assumptions on users. This is one important feature of any such scheme. The point is that if one have infinite amount of time and unlimited access to the source code of the verification algorithm, one can try all possible settings of the private key to come up with all quantum states until the verification algorithm accepts the money, in which case the counterfeiter would know the correct private key and be able to reproduce quantum bank notes.

People are excited about quantum money because the concepts and techniques that go into constructing quantum money could be useful to build other cryptographic primitives such as quantum software copy protection and quantum digital signature.

Quantum Software Copy Protection Imagine that you would like to distribute an expensive video game program P that you have spent significant resources building. You don't want people to copy the program endlessly. Classical computer software programs are in principle copyable, but one may be able to create software programs that is in principle not copyable with quantum information. You can encompass the program P as some quantum state $|\psi_{P,k_{priv}}\rangle$ that depends on P and a private key k_{priv}, and distribute the quantum state to the public. Software users who have accesses to the state $|\psi_{P,k_{priv}}\rangle$ can run some procedure with probably a public key to verify and execute the program. This quantum state should also have the feature that one cannot create two copies of the program without knowing some secret information of the private key. A big open question is whether this is possible or not. In some sense this is a generalization of quantum money and for similar reasons require computational assumptions.

Quantum Digital Signature Imagine you would like to sign a document using a digital signature. If the recipient of the signature has enough resources, he can theoretically forge multiple digital signatures of yours and this is a very undesirable aspect of classical signatures. Quantum digital signature scheme [5] allows you to encode the signature in a quantum state and then give a public quantum-bit key to verify the authenticity of the signature. These signatures utilize quantum one-way functions which allows the users with the public key to verify the authenticity of the signature. This is a completely safe method where the probability of successfully forging the signature is exponentially small.

References

[1] C. H. Bennett and G. Brassard. *Quantum cryptography: Public key distribution and coin tossing*. In Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, volume 175, page 8. New York, 1984. `http://researcher.watson.ibm.com/researcher/files/us-bennetc/BB84highest.pdf`

[2] A. Molina, T. Vidick, J. Watrous, *Optimal Counterfeiting Attacks and Generalizations for Wiesner's Quantum Money*, in Theory of Quantum Computation, Communication, and Cryptography. TQC 2012. Lecture Notes in Computer Science, vol 7582. 10.1007/978-3-642-35656-8_4.

[3] D. Gavinsky. *Quantum Money with Classical Varification.* Available as arXiv.org e-print 1109.0372, 2011.

[4] M. Zhandry, *Quantum lightning never strikes the same state twice*, arXiv:1711.02276, 2017.

[5] Daniel Gottesman, Isaac L. Chuang *Quantum Digital Signatures*, arXiv:quant-ph/0105032, 2001