COS 597A: Quantum Cryptography — Princeton University
Lecture 17 (November 14, 2018)
Lecturer: Mark Zhandry — Scribe: Xin Sun

Notes for Lecture 17

1 Another Approach to Building Quantum Money

Assume that we have a set $S \subseteq \{0,1\}^n$ and S is sparse. We want the quantum money state to be $|\psi_s\rangle = \frac{1}{\sqrt{|S|}} \sum_{x \in S} |x\rangle$. The verification process involves two steps:

- Test membership in S
- Verify in superposition

Here is an example of such a quantum money scheme based on SIS: Let $A \in Z_q^{n*m}$

To generate the banknote:

- $Gen()$: Construct $\frac{1}{\sqrt{2^m}} \sum_{x \in \{0,1\}^m} |x\rangle$
- Apply H_A in superposition, where $H_A = Ax \ mod \ q$ and $x \in \{0,1\}^m$. Then the state becomes $\frac{1}{\sqrt{2^m}} \sum_{x \in \{0,1\}^m} |x, H_A(x)\rangle$
- Measure H_A and we will get y, and the state collapses to $|\psi_y\rangle = \sum_{x: Ax = y \ mod \ q} |x\rangle$

So the money state $= |\psi_y\rangle$ and the serial number $= y$

One problem with this scheme is that verifying the state $|\psi_y\rangle$ in superposition is hard. It turns out that if A is random and the original state is uniform, then by Dihedral Hidden Subgroup this verification is hard. Therefore the quantum money based on SIS is easily broken.

1.1 Collapsing Hash Function

Now we will introduce the collapsing property for hash functions in the post-quantum setting, which is classical crypto with quantum adversary, solved by a stronger notion of security.

Let's look at SIS hash functions with a game, which involves both an Adversary A and a challenger Ch:

- A first sends the state $|\psi\rangle = \sum_{x\in\{0,1\}^m} \alpha_x|x\rangle$ to the challenger Ch
- The challenger Ch flips a bit b:

  if $b = 0$: Measure $|\psi\rangle$

  if $b = 1$: Measure H_A applied to $|\psi\rangle$

  In the end the challenger sends back the resulting state $|\psi'\rangle$ to A
- A tries to guess whether b is 0 or 1, and it outputs the bit b'

If H_A is collapsing, then A cannot tell what b is, which means that if I can verify superposition, then there is an adversary that distinguishes b. Therefore if I can turn SIS into quantum money algorithm, then I know that SIS is not collapsing.

Theorem 1. *(Informal) For any H, where H is not collapsing but is collision-resistant , H can be used to construct quantum money.*

Note that given a hash function H, and H is guaranteed to be collision-resistant, then we can always get something useful from it:

- If H is collapsing, then we will have classical security against quantum adversaries
- If H is not collapsing, then we can use H to build quantum money

Now We will prove intuitively why the above quantum money scheme is secure.

Proof. Assuming that I have an H that is not collapsing but is collision-resistant, and also assuming that $\Pr[b' = b]$ is roughly 1. We can split A into two adversaries: A_0 and A_1

- A_0: Comes up with the initial message $|\psi\rangle$ to the challenger and also keeps some internal state
- A_1: (The rest of the adversary A) Receives the internal state from A_0 and $|\psi'\rangle$ from the challenger, and in the end outputs b'

To generate quantum money state ($Gen()$):

- Run A_0 and it outputs $|state\rangle$ $|\psi\rangle$
- Evaluate H in superposition on $|\psi\rangle$ and we will get $|state\rangle \sum \alpha_x|x, H(x)\rangle$

- Measure $H(x)$ and get y, and the state becomes: $|state\rangle\ |\psi_y\rangle$, where $|\psi_y\rangle = \sum_{x:H(x)=y}\ \alpha_x|x\rangle$

So the money state = $|state\rangle\ |\psi_y\rangle$, and the serial number = y

Now the verification process goes through both classical phase and quantum phase:

In classical phase:

- Evaluate H in superposition and make sure that I get y back
- Measure $H(x)$ and get y':

  If $(y\ !=y')$, then reject

Note that only verifying in classical phase is not enough because we can break this scheme by measuring as many x's as we want, so we need to do something cleverer, which is verifying in superposition (Make sure that I'm in b=1 case and not b=0 case).

In quantum phase:

- Run A_1 on $|state\rangle\ |\psi_y\rangle$:

  If output is 0, then reject

If it passes both classical and quantum phases, then accept

A couple of observations: Suppose that I have an adversary A which is able to forge with 100% probability, then what I get is $|state\rangle\ |\psi_0\rangle$, $|state\rangle\ |\psi_1\rangle$. Then I know that:

- Since it passes the classical phase, $|\psi_0\rangle$, $|\psi_1\rangle$ have supports on preimages of y.
- Since it passes the quantum phase, $|\psi_0\rangle$, $|\psi_1\rangle$ are in superposition.

So if both states are in superposition, then by measuring $|\psi_0\rangle$, $|\psi_1\rangle$ I can get x_0, x_1, and $H(x_0) = H(x_1) = y$. Notice that x_0 is not equal to x_1 with a high probability because those two states are in superposition. Under the assumption that says H is collision-resistant, this collision is not possible, and therefore this quantum money scheme is secure.

Now we will briefly deal with this: $\Pr[b' = b]$ is roughly 1 issue. The problem is that we cannot always get the perfect distinguisher. One way to resolve this is to run $Gen()$ for many times, and the quantum money state would be the vector of all money states and the serial number would be the vector of all serial numbers.

But one tricky thing would happen: Running A in the quantum phase would perturb the state in the classical phase. So the state either holds for the classical phase or holds for the quantum phase, but we want both of them to hold at the same time. To resolve this, one delicate approach is that we only run one of the {classical, quantum} phase test on each money state randomly. □

Next we will move on to a relative topic, which is quantum lightening.

2 Quantum Lightening

2.1 Observations

We have this money state and we said that if an adversary A can clone this money state then it can find the collision. But it turns out that we get something stronger: with this scheme, even a Mint cannot produce the same state twice. An interesting question is that what we can do with this.

2.2 Applications

- Blockchain-less Cryptocurrency

  Differences from quantum money: it is decentralized so there is not Mint; it is also classical and all transactions are kept in a public registry.

- We can also build another quantum money scheme: The money state is $|\psi\rangle$ with serial number y s.t. $H(y) = 0^k *^n$, which begins with a lot of zeros.

  Note that if H is "ideal", then the only way to mint is to run $Gen()$ roughly 2^k times until find the valid money. But there are tons of problems with this: Over time as the computers become faster and faster, it's more easier to attack this scheme. And also if we try to modify (enlarge) k in order to make it more secure, then we will invalidate all old money states.

3 Certifiable Randomness

3.1 Background

In cryptography we often face random strings, and we want a way to certify that the string is indeed random. Classically we cannot verify this because we cannot tell whether the string comes from a PRG (pseudorandom generator) or not.

3.2 The Game

The game consists of an adversary A and a verification party Ver:

- A sends a string y along with the state $|\psi_y\rangle$ to $Ver(y, |\psi_y\rangle)$
- If y is sampled with low entropy, then Ver rejects with high probability
  If Ver accepts and A is computationally bounded, then y is uniformly random

So $|\psi_y\rangle$ is a proof that y is sampled with entropy.

A couple of observations:

- We cannot prove full entropy (Suppose that I have a procedure which outputs y, $|\psi_y\rangle$, then I can run this procedure multiple times and in the end I can pick one y which begins with 0).
- We can hope for quantum lightening implying the minimum entropy of y with logn (If I run this procedure polynomial times, then I can have two states with the same y, but that violates the quantum lightening security).