6.845 Quantum Complexity Theory

Final Project

January 17, 2011 *Paul Christiano*

1 Introduction

We are interested in the design of quantum banknotes: states which can be efficiently recognized but not copied. It is known that quantum banknotes can be produced using a random oracle, if the would-be forger is not allowed access to the verification procedure. The existence of public-key quantum money, which can be recognized by a publicly known verification procedure, remains open. It is known that there exists a quantum oracle— a unitary transformation available to all parties— relative to which public-key quantum money exists. The goal of this project is to design a scheme which is secure using only a classical oracle. We propose a scheme, but its security is dependent on a conjectured query complexity bound.

Formally, public key quantum money consists of a minting procedure M used by the bank and a verification procedure V known publicly, each parametrized by a security parameter k in which their running time is polynomial. The bank repeatedly invokes the M to produce banknotes $|\psi_i\rangle$. The verification procedure V should reject a note $|\psi_i\rangle$ with negligible probability (less than k^{-c} for any constant c). (By the Almost-As-Good-As-New lemma, this shows that a banknote can be verified any polynomial number of times without being signifcantly damaged.) Moreover, the banknotes are hard to forge in the following sense. Suppose an arbitrary polynomial time quantum adversary (the forger) obtains n banknotes: $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle$ and outputs a polynomial number of potentially entangled forgeries. If these forgeries are independently verified using V, then V should almost certainly accept at most n of them.

Although we will not analyze it here, one virtue of the scheme (and the reason for some of its complexity) is that it can also be used for quantum copy-protection in the sense defined by Aaronson.

2 Quantum Money

Let k be the security parameter of the quantum money. For any $S \subset \mathbb{F}_2^{2k}$, let $S^\perp$ denote its orthogonal complement.

To prepare a note, the bank first prepares a uniform superposition over k bit strings.

It then chooses a random k bit secret key sk. When given the secret key, the oracle outputs a corresponding public key pk. For each secret key sk, the oracle chooses a random $S \subset \{0,1\}^{2k}$ of size 2^k and a random bijection $f : \{0,1\}^k \to S$. When given sk, the oracle implements the bijections f and f^{-1}. Using this, the bank prepares the state $|S\rangle = 2^{-k/2} \sum_{s \in S} |s\rangle$. The banknote is $(|S\rangle, \text{pk})$.

When given the public key pk the oracle provides some additional functions used by the verification procedure. For each $x \in \{0,1\}^k$, the oracle chooses a random k dimensional subspace $S_x \subset \mathbb{F}_2^{2k}$. For each (x, pk), the oracle chooses a random bijection f_x between S and S_x. When given (x, pk), the oracle implements the map which is f_x on S and 0 elsewhere, and the map which is f_x^{-1} on S_x and 0^{2k} elsewhere. Furthermore, when given (x, pk), the oracle implements a membership oracle for $S_x^\perp \setminus \{0\}$.

Given $|S\rangle$ and this oracle, anyone can prepare the state $|S_x\rangle = \sum_{s \in S_x} |s\rangle$. The user can then take the Hadamard transform to obtain $\left|S_x^\perp\right\rangle$. To verify a quantum banknote, we choose x at random, prepare $S_x^\perp$ using this method, and feed that state into the membership oracle for $S_x^\perp \setminus \{0\}$. Accept the money if the membership oracle says 1 and reject otherwise. This returns 1 with probability $1 - 2^{-k}$ (2^{-k} is the probability that you get 0) for a real quantum banknote.

Quantum Copy Protection

In order to use this scheme for copy protection we need to modify it only slightly. Rather than a verification procedure, we need to allow an honest user to evaluate the copy-protected function. So the oracle provides a function which takes in any element of $S_x^\perp$ and outputs the copy-protected function evaluated at x. Essentially the same security proof will apply, but there are some complications (and the definitions all become much more difficult).

3 Security

We rely on a slightly strengthened version of the following conjecture.

Claim 1 *Suppose S is a uniformly random k dimensional subspace of $\mathbb{F}_2^{2k}$. and let $N = 2^k$. Given membership oracles for S and its orthogonal complement $S^\perp$, an algorithm making T oracle queries cannot find two distinct non-zero elements of S with probability greater than cT^4/N^2, for a constant c independent of N and T.*

A standard hybrid argument shows that the probability of finding a single non-zero element of S is at most cT^2/N. Intuitively, we need to prove that finding two elements

is twice as hard as finding one. Direct product theorems are known for ordinary Grover search; while this setting is more complicated, pairwise independence still implies that finding a second element in S having already found one is just as hard as finding one from scratch. This suggests that we may be able to strengthen existing results for Grover search to cover this case.

In fact, we need to strengthen this conjecture slightly further still. A would-be forger is presented simultaneously with many pairs of oracles satisfying the above conjecture, and the security of the money depends on their inability to break any one of those pairs.

Claim 2 *Suppose we choose a uniformly random k dimensional subspace $S(x) \subset \mathbb{F}_2^{2k}$ for each k bit string x. Given membership oracles for each $S(x)$ and each $S(x)^\perp$, as well as black box access to a commuting family of bijections $f(x,y) : S(x) \to S(y)$, an algorithm making T oracle queries cannot find two distinct non-zero elements of any $S(x)$ with probability greater than cT^4/N^2, for a constant c independent of N and T.*

With this result in hand it is easy to prove the unforgeability of our quantum banknotes. It is sufficient to convert a forging algorihtm into one which violates the conjecture.

Suppose A is a forger making only T oracle queries. When run with $n = \text{poly}(k)$ banknotes and oracle access A outputs $m = \text{poly}(k)$ arbitrarily entangled forgeries. Let p be the probability that $n+1$ of these forgeries pass the verification procedure (the order or timing of verification does not affect the outcome). Suppose that p is non-negligible (by which we mean k^{-c} for some constant c).

Now consider the public keys of the notes produced by A. Since each of these immediately measured in the standard basis by the verification procedure, we may assume that they are classical. It is clear (by a standard hybrid argument) that A cannot produce a forgery using an unseen public key with non-negligible probability. Therefore p can be non-negligible if and only if the forger has a non-negligible chance of producing two forgeries with the same public key. But we will show that the conjecture then implies that T is exponential in k.

Suppose $(\text{pk}, |\psi\rangle)$ passes the verification procedure. Then for a randomly chosen x, the hadamard transform of $f(|\psi\rangle)$ has non-negligible probability mass in $S_x^\perp$ (by $f(|\psi\rangle)$ we mean the state obtained by applying f and then applying f^{-1} to erase the original state). Because it is statistically difficult to determine an element of $S_x^\perp$ given only one element of S_x, $f(|\psi\rangle)$ must have significant probability mass on two distinct non-zero elements of S_x. Thus $|\psi\rangle$ must have significant probability mass on two distinct elements of S. So by measuring $|\psi\rangle$, we obtain a sample which has the following property: for every $s \in S$, there is a non-negligible probability of measuring $S \backslash s$. If we have two states which pass the verification procedure, we can find two distinct elements of S with significant probability. This implies, by the lemma, that if any algorithm produces two such states with significant probability then T is exponentially large in k.