# Quantum Secret Sharing in Practice

Keanu Spies

June 9, 2019

## Motivation

Secure access to data has been a problem in Computer Science since its creation. Thus, finding methods to establish communication between parties in a way that is both secure and efficient is a major field in cryptography. In this project we will present work to define practical solutions to the problem using Quantum Programming.

## Introduction

### 0.1 The GHZ state

The first step to sharing a quantum secret is creating a GHZ (Greenberger–Horne–Zeilinger) [1] state. This is the maximally entangled three qubit state:

$$|\psi_i\rangle = \frac{1}{\sqrt{2}}|000\rangle + \frac{1}{\sqrt{2}}|111\rangle$$

.

### 0.2 Sharing a Secret Between Three Parties

First I will work with the theory defined Hillery et al. in their paper "Quantum Secret Sharing" [2] in order to understand how the sharing protocol must occur.

Quantum secret sharing works under the above GHZ state. Assume that Alice is the party sharing the secret to Bob and Charlie. In order for Bob and Charlie to uncover exactly which qubit alice sent with certainty they will both need to do the reading and communicate with Alice under which Basis to do the reading. This works becase we can expand the GHZ state into the following:

$$|\psi_i\rangle = \frac{1}{2\sqrt{2}}\Big[(|+x\rangle_a|+x\rangle_b + |x\rangle_a|x\rangle_b)(|0\rangle_c + |1\rangle_c) + (|+x\rangle_a|x\rangle_b + |x\rangle_a|+x\rangle_b)(|0\rangle_c|1\rangle_c)\Big].$$

Meaning that Charlie is able to tell whether Alice and Bob made their readings along the same basis (±x) or not but has no way of knowing what their readings were precisely (Hillery et all 1998). This is however reqiures that Charlie do his reading in the same direction that Alice and Bob did their readings (i.e. x direction) in order to determine if

Alice and Bob's readings were correlated or anti-correlated. Similarly Bob needs Charlie in order to determine if his reading is the same or different to Alices.

Thus the three parties should all perform their reading in the relevant direction which should be announced beforehand. This should happen as follows (according to Hillery et al.): Charlie and Bob each measure in the same direction and send their direction to Alice, who measures in a specific direction and sends all three directions to Bob and Charlie.

This system is general to Alice doing her reading in the Y direction as well since we can expand the entire system into the following table (Alice's measurements fall in the columns, Bob's measurements fall in the rows and Charlie's measurements are within the cells):

| | | **Alice's Reading** | | | |
|---|---|---|---|---|---|
| | | $\|+x\rangle$ | $\|-x\rangle$ | $\|+y\rangle$ | $\|-y\rangle$ |
| **Bob's** | $\|+x\rangle$ | $\|0\rangle + \|1\rangle$ | $\|0\rangle - \|1\rangle$ | $\|0\rangle - i\|1\rangle$ | $\|0\rangle + i\|1\rangle$ |
| | $\|-x\rangle$ | $\|0\rangle - \|1\rangle$ | $\|0\rangle + \|1\rangle$ | $\|0\rangle + i\|1\rangle$ | $\|0\rangle - i\|1\rangle$ |
| | $\|+y\rangle$ | $\|0\rangle - i\|1\rangle$ | $\|0\rangle + i\|1\rangle$ | $\|0\rangle - \|1\rangle$ | $\|0\rangle + \|1\rangle$ |
| | $\|-y\rangle$ | $\|0\rangle + i\|1\rangle$ | $\|0\rangle - i\|1\rangle$ | $\|0\rangle + \|1\rangle$ | $\|0\rangle - \|1\rangle$ |

From this we can see that if Alice measures in the X direction, Charlie and Bob need to measure in a correlated fashion in order to produce the correct results. However, if Alice measure in the Y direction, Charlie and Bob need to measure in different directions in order to uncover the measurement that Alice made.

# 1 Methods

## 1.1 Practical implementation of a measurement

In order for each party to measure in their required direction we need to project their reading into that space before hand. Below I present a sample reading between Alice, Bob and Charlie.

```
DECLARE ro BIT[3]
// Alice entangles all three states into the GHZ
H 0
CNOT 0 1
CNOT 0 2
// Alice measure in the Y direction
Y 0
CNOT 2 0
CNOT 1 0
H 0
MEASURE 0 ro[0]
H 1
CNOT 1 2
// Bob measure in the X direction
X 1
CNOT 2 1
H 1
MEASURE 1 ro[1]
```

```
// Charlie measure in the X direction
Y 2
MEASURE 2 ro[2]
```

## 1.2 Creating a key between parties

In order for secret sharing to occur Alice needs to be able to create a key between the two parties in the exchange in order to ensure that no fraudulent behavior has occured. In Hillerey et al. they describe how the Quantum secret is robust to eavesdropping between the two parties receiving the qubits (Bob and Charlie) and any interception from another party (ie. Eve).

### 1.2.1 Practical implementation of creating a key between parties

Given that we have shown how to do a single measurement between parties we can now perform multiple measurements to create a key between the three parties. To do this we can entangle 8 GHZ states and send 2 qubits of each state to the two recieving parties.

Bob and Charlie will measure in whatever direction they please (most optimally randomly so that the other party cannot predict which direction they are measuring in). They inform Alice of their measurement direction to which she will measure either in the X direction if Bob and Charlie have correlated measurements or in the Y direction if they have opposing measurements.

Once this has occurred for each of the GHZ states Bob and Charlie can work together to uncover what Alice had measured, thus allowing to create a key with Alice for the exchange to occur.

Below is psuedocode for this system:

```
def create_secret_key(num_bits, ghz):
    alice_out = bob_out = charlie_out = []
    for i in range(num_bits):
        # bob and charlie choose their directions to measure
        bob_dir = random.choice([X, Y])
        charlie_dir = random.choice([X, Y])
        # they send their directions to alice
        # she chooses X if bob and charlie has the same dir
        # or Y if they had opposite
        if bob_dir == charlie_dir:
            alice_dir = X
        else:
            alice_dir = Y
        # measure for all parties.
        ro = one_bit_measure(alice_dir, bob_dir, charlie_dir)
    return alice_out, bob_out, charlie_out
```

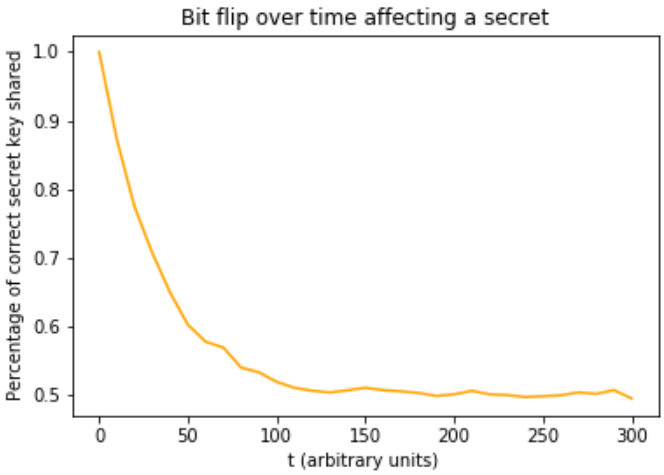
Once the measurements have been made, Bob and Charlie will need to do a table lookup of their results to determine what alice must have measured allowing them to create a key with her.

# 2 Noise Introduction

Given that the system requires for transport across a distance between Alice and Bob and Charlie, a great deal of noise might apply to the system. This could be due to many factors.

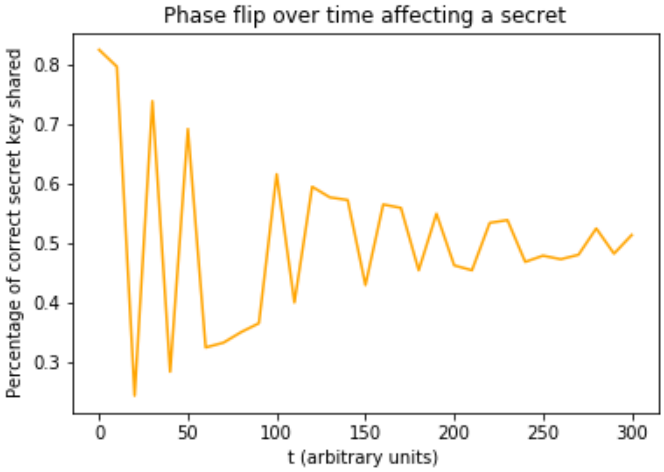
## 2.1 Bit and Phase flips

First I will show how the possibility of bit-flip and phase-flip will affect the efficacy of a key being created. For a bitflip I model the probability that an identity gate applied to the system will decay the state of a qubit over time.




Above we can see that the system slowly decays to a 50% correctness, which is the probability of guessing the correct direction by a malicious party and is thus not a helpful key to create.

For a phase-flip I model the probability that a X, Y, Z, or I gate applied to the system will decay the state of a qubit over time. Since these gates are involved in the measurement applied by each party this will affect the efficacy of their measurements and thus the key created by Bob and Charlie.




Similar to the bit flip decoherence noise oscillates around a 50% correctness scheme which is as good as guessing. With no error correction the secret is not secure and cannot be trusted.

Both of these could be prevented using a Shor Code, which would have the ability to correct for bit and phase flips over a distance. However there is the problem where the qubits in charge of fixing the noise themselves undergo noise, at which point we would not be able to recreate the original system as it is transferred between parties. This is the foundational problem in QEC which is still an active field.

### 2.2 Readout Errors

Another problem that arrises, especially with GHZ states is readout error [3]. Here the GHZ state which is in the general form of:

$$\alpha|000\rangle + \beta|111\rangle$$

can have different evaluations of $\alpha$ and $\beta$ given noise introduced in the readings.

PyQuil allows for an estimation of the readout error of a system. I found that $\alpha$ and $\beta$ under my model were:

$$\bar{\alpha}^2 = 0.30000$$
$$\bar{\beta}^2 = 0.3557$$

As opposed to the $1/\sqrt{2}$ that a perfect maximally entangled state would have. This is especially a problem in Quantum Secret Sharing since so much of the computation relies on the GHZ state and a maintenance of the maximally entangled system. If this readout noise is introduced I found that the system was only able to generate a correct key around 29% of the time, which is quite below the 50% guessing strategy of an eavesdropper.

## 3 Conclusion

In this project I have shown that it is possible to implement a quantum secret sharing code on a modern day quantum virtual machine with a possible extension into a real quantum computer in the future. The biggest issues that occur are to do with noise over time, since

the system struggles to maintain the GHZ state over time if any noise is introduced at all. Without fidelity the parties can no longer trust the security of their exchange and will struggle to benefit from a secure encoding.

# References

[1] D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond bell's theorem," *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, p. 69–72, 1989.

[2] M. Hillery, V. Buzek, and A. Berthiaume, "Quantum secret sharing," 1998.

[3] Rigetti, "Noise and quantum computation," 2019.