Quantum Secret Sharing of Classical Information

CSE 664 – Applied Cryptography
Project Proposal

Alessandro N. Baccarini
anbaccar@buffalo.edu

May 4, 2020

Abstract

We explore the field of Quantum Secret Sharing (QSS) with classical information. We introduce quantum computing fundamentals and implement the first of the three protocols, given in [HBB99].

1 Quantum Secrecy

The necessary background information for Quantum Computing is provided as an appendix. We demonstrate how to leverage quantum mechanical properties to share classical bits against an eavesdropper, who can either be external or internal. This is effectively a key distribution protocol [HBB99], where Alice wants to establish a shared key between two participants Bob and Charlie.

1.1 Quantum Key Distribution

We first demonstrate how to invoke quantum mechanical properties to protect classical information. Suppose Alice, Bob, and Charlie each have one particle from a GHZ triplet in the entangled state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0_a 0_b 0_c\rangle + |1_a 1_b 1_c\rangle). \tag{1}$$

Each randomly choose to measure their particle in the x or y direction, and then publicly announce the chosen direction (but not the actual measurement results). We define the x and y eigenstates:

$$|+x\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right), \tag{2}$$

$$|-x\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - |1\rangle\right), \tag{3}$$

$$|+y\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle + i\,|1\rangle\right), \tag{4}$$

$$|-y\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle - i\,|1\rangle\right). \tag{5}$$

Note, adding/subtracting Equations (2) and (3) yields

$$|0_i\rangle = \frac{1}{\sqrt{2}}\left(|{+x}\rangle + |{-x}\rangle\right), \tag{6}$$

$$|1_i\rangle = \frac{1}{\sqrt{2}}\left(|{+x}\rangle - |{-x}\rangle\right), \tag{7}$$

where $i = a, b, c$. We can examine how Alice and Bob's measurements affect the state of Charlie's particle. We can substitute Equations (6) and (7) (for $i = a, b$) into the GHZ triplet (Equation (1)) to obtain

$$\begin{aligned}
|\psi\rangle =& \frac{1}{\sqrt{2}}\left\{\left[\frac{1}{\sqrt{2}}(|{+x}\rangle_a + |{-x}\rangle_a)\cdot\frac{1}{\sqrt{2}}(|{+x}\rangle_b + |{-x}\rangle_b)\cdot|0\rangle_c\right]\right. \\
&+ \left.\left[\frac{1}{\sqrt{2}}(|{+x}\rangle_a - |{-x}\rangle_a)\cdot\frac{1}{\sqrt{2}}(|{+x}\rangle_b - |{-x}\rangle_b)\cdot|1\rangle_c\right]\right\}
\end{aligned} \tag{8}$$

$$\begin{aligned}
=& \frac{1}{2\sqrt{2}}\left[(|{+x}\rangle_a + |{-x}\rangle_a)\cdot(|{+x}\rangle_b + |{-x}\rangle_b)\cdot|0\rangle_c\right. \\
&+ \left.(|{+x}\rangle_a - |{-x}\rangle_a)\cdot(|{+x}\rangle_b - |{-x}\rangle_b)\cdot|1\rangle_c\right]
\end{aligned} \tag{9}$$

$$\begin{aligned}
=& \frac{1}{2\sqrt{2}}\left[(|{+x}\rangle_a|{+x}\rangle_b + |{+x}\rangle_a|{-x}\rangle_b + |{-x}\rangle_a|{+x}\rangle_b + |{-x}\rangle_a|{-x}\rangle_b)\cdot|0\rangle_c\right. \\
&+ \left.(|{+x}\rangle_a|{+x}\rangle_b - |{+x}\rangle_a|{-x}\rangle_b - |{-x}\rangle_a|{+x}\rangle_b + |{-x}\rangle_a|{-x}\rangle_b)\cdot|1\rangle_c\right]
\end{aligned} \tag{10}$$

$$\begin{aligned}
\implies |\psi\rangle =& \frac{1}{2\sqrt{2}}\left[(|{+x}\rangle_a|{+x}\rangle_b + |{-x}\rangle_a|{-x}\rangle_b)(|0\rangle_c + |1\rangle_c)\right. \\
&+ \left.(|{+x}\rangle_a|{-x}\rangle_b + |{-x}\rangle_a|{+x}\rangle_b)(|0\rangle_c - |1\rangle_c)\right]
\end{aligned} \tag{11}$$

If Alice and Bob's measurements are the same, then Charlie will have the state $(|0\rangle_c + |1\rangle_c)/\sqrt{2}$. Conversely, if their measurements are different then Charlie will have the state $(|0\rangle_c - |1\rangle_c)/\sqrt{2}$. He can determine which of these states he has by measuring in the x-direction, and Table 1 contains all possible states he can be in given Alice and Bob's measurements. If Charlie knows the direction of Alice and Bob's measurements (either x or y), he can determine whether their results are the same or opposite, while simultaneously not learning what their results actually are.

Table 1: Summary of the effects of Alice and Bob's measurements (edges) on Charlie's state (given in the gray boxes). A cleaner representation is given in the second table.

		A			
		$+x$	$-x$	$+y$	$-y$
B	$+x$	$\|0\rangle + \|1\rangle$	$\|0\rangle - \|1\rangle$	$\|0\rangle - i\,\|1\rangle$	$\|0\rangle + i\,\|1\rangle$
	$-x$	$\|0\rangle - \|1\rangle$	$\|0\rangle + \|1\rangle$	$\|0\rangle + i\,\|1\rangle$	$\|0\rangle - i\,\|1\rangle$
	$+y$	$\|0\rangle - i\,\|1\rangle$	$\|0\rangle + i\,\|1\rangle$	$\|0\rangle - \|1\rangle$	$\|0\rangle + \|1\rangle$
	$-y$	$\|0\rangle + i\,\|1\rangle$	$\|0\rangle - i\,\|1\rangle$	$\|0\rangle + \|1\rangle$	$\|0\rangle - \|1\rangle$

$\Longrightarrow$

		A			
		$+x$	$-x$	$+y$	$-y$
B	$+x$	$+x$	$-x$	$-y$	$+y$
	$-x$	$-x$	$+x$	$+y$	$-y$
	$+y$	$-y$	$+y$	$-x$	$+x$
	$-y$	$+y$	$-y$	$+x$	$-x$

We note that if Alice chooses the x direction to measure her qubit, Bob and Charlie must measure theirs in the same direction (either both x or y). However, if Alice chooses the y direction, then Bob and Charlie must measure theirs in opposite directions (Bob and Charlie's directions are anti-correlated).

Since each party's chosen direction is random, only half of the GHZ triplets provide meaningful results. For example, if Alice and Bob both measure in the x direction, then Charlie must also measure in the x direction. Otherwise, he will gain no information. The complete construction is summarized in Figure 1.

***Quantum Key Distribution Scheme*:**

Shared Input: The GHZ state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0_a 0_b 0_c\rangle + |1_a 1_b 1_c\rangle)$$

is shared among the three parties A, B, and C, where A is defined as the dealer.

Output: A bit b shared among the parties.

The Protocol:

1. The parties independently and randomly choose a measurement direction $d \in \{x, y\}$. and announce it to each other.
2. If $d_A = x$, then $d_B = d_C$. Otherwise, discard the round.
3. If $d_A = y$, then $d_B \neq d_C$. Otherwise, discard the round.
4. Each party measures their qubit in their respective directions, which causes it to collapse to a classical bit b_i.
5. Set A's measurement $b_A = b$, the shared bit that B and C must determine.
6. B and C share their measurements b_B and b_C, and conduct a table lookup to determine b.

Figure 1: The complete quantum key distribution protocol.

1.2 Eavesdropper Detection

We want to show how the presence of an "eavesdropper" (a non-cooperative party) can be detected with this construction. We first consider the following illustrative example. For simplicity, we will always assume Bob is the dishonest party.

Example 1. Let Bob be dishonest and managed to obtain Charlie's particle, as well as his own. He measures the particles and sends one to Charlie. His goal is to determine Alice's bit without using Charlie without being detected. Bob does not know the direction Alice measured her particle, and therefore must guess with a probability of 1/2 which basis to choose:

$$|\pm x\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle \pm |11\rangle\right), \text{ or} \tag{12}$$

$$|\pm y\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle \pm i\,|11\rangle\right). \tag{13}$$

If he is correct, he can determine Charlie's measurement, and hence know Alice's bit. If Alice measured in the x direction and found $|+x\rangle$, then the state Bob receives is $(|00\rangle + |11\rangle)/\sqrt{2}$. If Bob measures in the $|\pm x\rangle$ basis, then he knows what the two-particle state is because

$$\frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right) = \frac{1}{\sqrt{2}}\left(|+x\rangle\,|+x\rangle + |-x\rangle\,|-x\rangle\right). \tag{14}$$

Now suppose Bob chooses incorrectly. Alice measured in the y direction, and again Bob measures in the $|\pm x\rangle$ basis. He has a probability of 1/2 of getting either basis vector. Bob sends one particle to Charlie, then both of them measure their particles. To produce a valid key bit, Bob and Charlie must make different measurements (one measures x and the other y). In the $|\pm x\rangle$ basis, there is no correlation between x and y measurements. The overall probability of an error in this scheme is 1/4: 1/2 from picking the wrong basis, then 1/2 from getting the wrong result.

2 Quantum Instructions

Prior to discussing the details of our implementation, it is necessary to provide an overview of the various quantum gates and operations that are required for the protocol.

1. *The Hadamard gate*:

   The Hadamard gate H is arguably one of the most useful gates [NC10] in quantum computing. Essentially, the gate rotates the states $|0\rangle$ and $|1\rangle$ to $|+\rangle$ and $|-\rangle$, respectively, where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. The gate can be expressed as the matrix

   $$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{15}$$

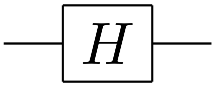
   

   Figure 2: Hadamard gate.

   It is clear that $H^2 = I$, thus applying H twice to a state has no final effect. Its usage in quantum algorithms as an initial step corresponds to mapping n qubits initialized with $|0\rangle$ to a superposition of all 2^n orthogonal states in the $|0\rangle, |1\rangle$ basis.

2. *The CNOT (cX) gate:*

   The controlled NOT (CNOT or cX) gate acts on a pair of qubits, where one is specified as the "control" and the other the "target." If the control qubit is set to 1, then the target qubit is flipped. As a matrix, we have

   $$\text{CNOT} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{16}$$

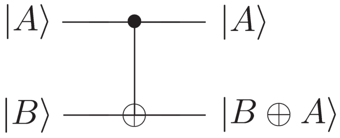
   

   Figure 3: CNOT gate.

   The CNOT gate can also be interpreted as the gate that maps $|A, B\rangle \rightarrow |A, A \oplus B\rangle$ (see Figure 3).

3. *The $S^\dagger$ gate*:

   The $S^\dagger$ gate is corresponds to rotating a qubit on the Bloch Sphere around the z axis by $\pi/2$ radians. As a matrix, it is defined as

   $$S = \begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}. \tag{17}$$

It is useful for moving information between the x and y bases.

3 Implementation Description

Since we are using unconventional cryptographic techniques, we explicitly outline the functionality of our protocol in this section. The protocol is implemented in Python. We first note that we changed our chosen quantum programming library from Rigetti Computing's pyQuil [rig] to IBM's Qiskit. The primary motivation is the lack of documented examples using pyQuil, whereas IBM has published extensive information on their Quantum Experience web page [IBM20]. Consequentially, the program requires the `qiskit` Python package to run.

The first two steps of the protocol (state creation and measurement) are contained within the `create_and_measure` function within `ibm.py`. The function takes in the three measurement directions of Alice, Bob, and Charlie as a list parameter `d` (such as `d = [’y’, ‘y’, ‘x’]`).

3.1 State Creation

The first step of our protocol is creating the shared GHZ state

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0_a 0_b 0_c\rangle + |1_a 1_b 1_c\rangle). \tag{18}$$

When measuring this state, half the results should be $|000\rangle$, and the other half $|111\rangle$. To construct the GHZ state, we declare a three-qubit system with three classical registers and create a new circuit:

```
q = QuantumRegister(3)
c = ClassicalRegister(3)
ghz = QuantumCircuit(q, c)
```

Next, we apply the Hadamard gate H to qubits `q[0]` and `q[1]` and the X gate to `q[2]`. This takes the ground state of the system to $\frac{1}{2}\left(|001\rangle + |011\rangle + |101\rangle + |111\rangle\right)$.

```
ghz.h(q[0])
ghz.h(q[1])
ghz.x(q[2])
```

We then apply two CNOT gates that entangle the qubits into the state $\frac{1}{2}\left(|001\rangle + |010\rangle + |100\rangle + |111\rangle\right)$.

```
ghz.cx(q[1],q[2])
ghz.cx(q[0],q[2])
```

Lastly, we apply the Hadamard gate to every qubit such that previous state is mapped to our original GHZ sate $|\psi\rangle = \frac{1}{\sqrt{2}}(|0_a 0_b 0_c\rangle + |1_a 1_b 1_c\rangle)$.

```
ghz.h(q[0])
ghz.h(q[1])
ghz.h(q[2])
```

The graphical representation of the circuit is shown in Figure 4. In normal usage (as we will see), it is recommended to include a "barrier" (denoted by a vertical dotted line) in the circuit after state creation, since it prevents the compiler from automatically combining gates for the sake of optimization. If we were to measure the system in the standard basis at this point, the final result would be either `000` or `111`. Repeating this a large number of times shows that the results are evenly distributed between these two states. However, we are interested in measuring the system in the X and Y bases.

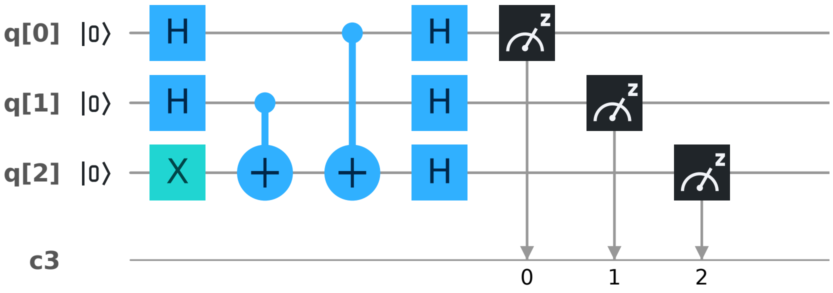


Figure 4: GHZ state as a quantum circuit.

3.2 Measurement

At this point, we have successfully initialized our three-qubit GHZ state. We now wish to determine the effect of measuring in the X and Y bases. To measure a qubit q in the X basis, applying H is sufficient.

```
ghz.h(q)
```

Conversely, to measure q in the Y basis, we first apply $S^\dagger$, followed by H.

```
ghz.sdg(q)
ghz.h(q)
```

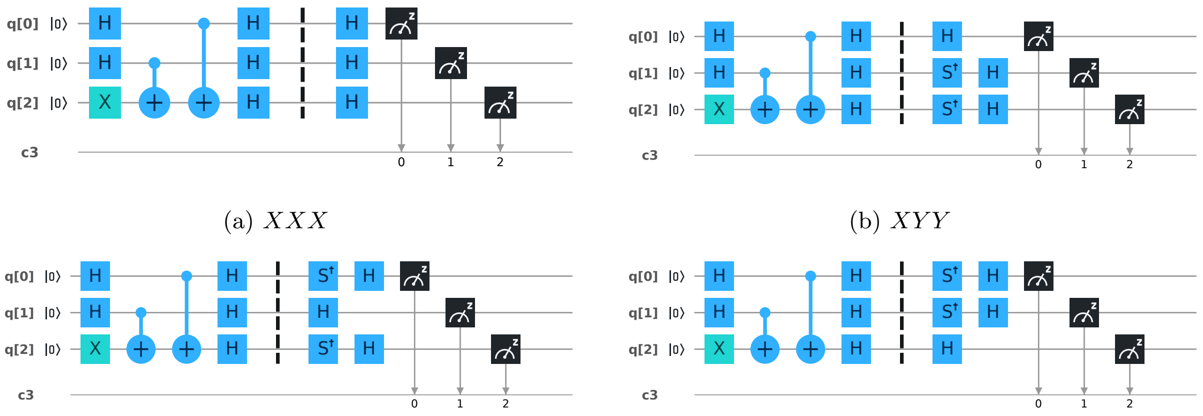
We know from Table 1 that there are four allowable measurement direction combinations: XXX, XYY, YXY, and YYX. Therefore, we can generate four possible circuits (shown in Figure 5).

After applying the appropriate gates, the whole system is measured. The final circuit is executed and the results are stored in the classical registers.

```
ghz.measure(q[0], c[0])
ghz.measure(q[1], c[1])
ghz.measure(q[2], c[2])

job = execute(ghz, backend = Aer.get_backend('qasm_simulator'), shots=1)
result = job.result()

return result.get_counts(ghz)
```



(a) XXX

(b) XYY

(c) YXY

(d) YYX

Figure 5: The four possible combinations of measurement bases of the GHZ state.

The final output of the circuit is a combination of three bits, which denote the measurement signs of A, B, and C. For our convention, `1` corresponds to a positive measurement, and `0` to a negative one. As an example, suppose the directions chosen by the parties are `[‘x’, ‘y’, ‘y’]`. The system may output `101` for the experiment. Therefore, this corresponds to Alice measuring in $+x$, Bob in $-y$, and Charlie in $+y$ (as per Table 1). We set the output bit b as Alice's measurement.

```
result = create_and_measure(dirs)
measurements = getList(result) # converting the output to a list

A_key = int(measurements[0][0]) # setting A's output bit

B_bit = int(measurements[0][1])
C_bit = int(measurements[0][2])
```

3.3 Reconstruction

Now Bob and Charlie must use their measurements to deduce Alice's bit. This is accomplished with a simple table lookup that corresponds to Table 1. In the `main` function, we call the `reconstruct` function, which is parameterized by the `directions`, B's measurement `B_bit`, C's measurement `C_bit`, and "honesty" `hnsty`. The last parameter `hnsty` is set to `0` if Bob and Charlie are intending to cooperate.

```
hnsty = 0

A_bit = reconstruct(dirs, B_bit, C_bit, hnsty)
...
def reconstruct(directions, B_bit, C_bit, honesty):

    if honesty == 0: # B and C cooperate

        return measure_table(directions,B_bit, C_bit)

    else: # B and C do not cooperate
        guess = random.randint(0, 1)
        return measure_table(directions, B_bit, guess)
```

If we want to experiment with a non-cooperative party, we set `hnsty = 1`, which corresponds to Bob simply guessing Charlie's measurement (either `0` or `1`). The `measure_table(dirns, bbit, cbit)` is Table 1 converted into if-else statements, and outputs Bob and Charlie's conclusion of what Alice's measurement was. If we wish to generate multiple k bits of a shared key, we simply call the `main()` function k times.

As an aside, we opted to maintain all computation to a single, local program, rather than distributing it across several programs to simulate actual communication (using sockets). We determined that implementing the protocol locally was sufficient for our final analysis.

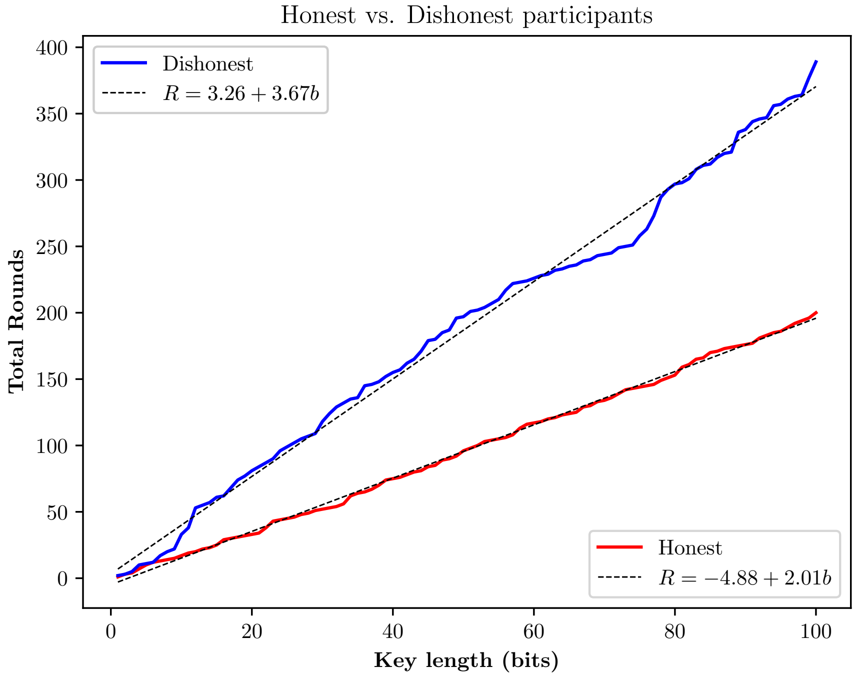


Figure 6: Protocol efficiency when dealing with honest and dishonest (non-cooperative) participants.

4 Results and Discussion

We quantitatively evaluate our protocol in three ways: how many rounds are required to generate a k-bit key with honest and dishonest participants (non-cooperative), and how many rounds are required on average to generate a 100-bit key.

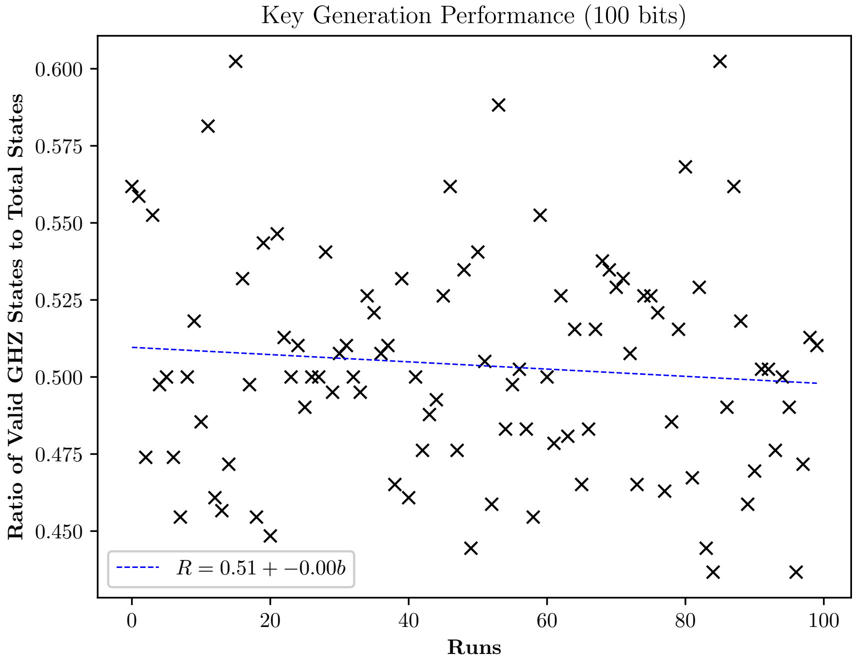


Figure 7: Ratio of valid GHZ states to total states generated for a 100-bit key (100 iterations, all parties cooperating).

We first analyze the number of rounds (number of GHZ states generated) required to generate a k-bit key in the presence of honest and dishonest participants. For honest parties (the red line) Figure 6 demonstrates that we have to generate approximately 2.01 GHZ states in order to generate a k-bit key (as evident by the slope of the first trend line). This result clearly agrees with the conclusion made in [HBB99], that on average $2N$ GHZ triplets are used to generate a N-bit key.

When examining the performance implications of a dishonest party on the protocol, we count failed guesses of the final bit (where Bob and Charlie's key bits do not match Alice's) towards the total number of rounds required to generate a k-bit key. We see in Figure 6 (the blue line) that the number of rounds required increases significantly more rapidly than in the honest case. We note that up until about 5 bits, the results are essentially indistinguishable from the honest case. In a realistic implementation of the protocol, counting the number of rounds required after the first 5 bits and comparing it to the expected results would be an effective way of detecting the presence of non-cooperative party.

We verify the consistency of our results by evaluating the protocol 100 times to generate a 100-bit key, and we take the ratio of valid GHZ states to the total states generated for a single run. In Figure 7, we see most iterations collect between 0.475 and 0.550, with several outliers as high as 0.600 and as low as 0.425. Nonetheless, these results agree with the prediction the previous prediction.

Given these results, can we assume that this protocol is suitable for practical usage? The answer is unquestionably *no*. This conclusion is derived from our fundamental definitions of security.

Earlier in the course, we specified the experiment $\mathtt{PrivK}^{\mathsf{eav}}_{\mathcal{A},\mathcal{E}}$ for determining if an encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is perfectly secret in the presence of an eavesdropper. We define a new experiment for our quantum key distribution scheme $\mathcal{Q} = (|\psi\rangle_{\text{GHZ}}, \mathcal{D})$ for a dishonest (non cooperative) party $\mathcal{B}$:

***Experiment*:** $\mathsf{QuantK}^{\mathsf{non\text{-}co\text{-}op}}_{\mathcal{B},\mathcal{Q}}$

1. Three valid directions d_i are chosen from $\mathcal{D} = \{x, y\}$.
2. Alice, $\mathcal{B}$, and Charlie measure $|\psi\rangle_{\text{GHZ}}$ in their d_i's. Charlie does not share his measurement with $\mathcal{B}$.
3. $\mathcal{B}$ outputs a bit $b' \in \{0, 1\}$ as its guess of Charlie's measurement $b \in \{0, 1\}$.
4. Experiment outputs 1 if $b' = b$ ($\mathcal{B}$ wins) and 0 otherwise.

The probability of $\mathcal{B}$ winning this experiment will always be 1/2, namely

$$Pr\left[\mathsf{QuantK}^{\mathsf{non\text{-}co\text{-}op}}_{\mathcal{B},\mathcal{Q}} = 1\right] = \frac{1}{2}. \tag{19}$$

In the lens of perfectly secret encryption schemes, one may conclude that this is a "perfectly secret" key distribution protocol. However, the experiment is effectively meaningless, since it has no correlation to its classical counterpart ($\mathsf{KE}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n)$ [KL14]). The protocol may be suitable in the generating a single bit, but the performance is severely limited by the communication component when generating longer keys.

5 Conclusion

In this project we implemented the key distribution scheme outlined in [HBB99]. We analyzed its performance in the presence honest and dishonest parties. We conclude that while it is an interesting exercise, it has no real practical applications.

A Classical Secrecy

A classical cryptographic *secret sharing scheme* is defined by the following canonical scenario: a "dealer" holds some secret x and distributes shares x_i to a set of n participating parties $P_i, \dots P_n$. Any t users can reconstruct the secret, but any amount $t-1$ or fewer cannot recover any information about x. This is referred to as (n,t)-*threshold scheme* (with $t \leq n$) [KL14]. A Shamir Threshold Scheme is constructed as follows:

Definition 1. (**(n,t) Shamir Threshold Scheme**) Given a secret $s \in \mathbb{F}$, the dealer chooses a uniform $a_1, \dots, a_{t-1} \in \mathbb{F}$. Letting $a_0 = s$, we define the polynomial

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1} = \sum_{i=0}^{t-1} a_i x^i. \tag{20}$$

Each party P_i receives a unique point $(x_i, p(x_i))$ on the polynomial.

To reconstruct the shares, t users pool their shares $y_{i_1}, \dots, y_{i_t}$ and compute the unique degree-$(t-1)$ polynomial

$$p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}. \tag{21}$$

Each party's $y_i = p(x_i)$ obtains one linear equation in the t unknowns $a_0, a_1, \dots, a_{k-1}$, which gives us following system of linear equations:

$$\begin{aligned} a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_{t-1} x_1^{t-1} \quad &= y_1 \\ a_0 + a_1 x_2 + a_2 x_2^2 + \cdots + a_{t-1} x_2^{t-1} \quad &= y_2 \\ \vdots \qquad\qquad\qquad &\quad \vdots \\ a_0 + a_1 x_t + a_2 x_t^2 + \cdots + a_{t-1} x_t^{t-1} \quad &= y_t. \end{aligned} \tag{22}$$

Expressed as a matrix, we have

$$\begin{pmatrix} 1 & x_1 & x_1^2 & \dots & x_1^{t-1} \\ 1 & x_2 & x_2^2 & \dots & x_2^{t-1} \\ 1 & x_3 & x_3^2 & \dots & x_3^{t-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{t-1} & x_{t-1}^2 & \dots & x_t^{t-1} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_t \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_t \end{pmatrix}. \tag{23}$$

The coefficient matrix (denoted by $V_t(x_0, \dots, x_{t-1})$) is the *Vandermonde matrix.* The determinant of a square Vandermonde matrix is expressed as

$$\det(V) = \prod_{1 \leq i \leq j \leq n} (x_j - x_i), \tag{24}$$

which is equal to zero if all elements x_i are distinct. In our system of equations, we know from our construction that each x_i is unique, so all terms in the product are nonzero, implying the existence of a unique solution. Therefore, t or more participants can recover the secret.

B Quantum Computation

A *quantum state* $|\psi\rangle$ (hereinafter referred to simply as a *state*) is a *superposition* of classical states, written as

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \cdots + \alpha_{N-1} |N-1\rangle \,, \tag{25}$$

where $\alpha_i \in \mathbb{C}$ is the *amplitude* of $|i\rangle$ in $|\psi\rangle$. The states $|0\rangle, \ldots, |N-1\rangle$ form the orthonormal basis of an N-dimensional complex vector space with an inner product, known as a *Hilbert Space* $\mathcal{H}$. A state $|\psi\rangle \in \mathcal{H}$ is written as

$$|\psi\rangle = \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{pmatrix}, \tag{26}$$

where $\sum_{j=0}^{N-1} |\alpha_i|^2 = 1$ (we elaborate on this restriction shortly). This satisfies the *normalization condition*, which states for $|\psi\rangle$ to be a unit vector, then $\langle\psi|\psi\rangle = 1$. We can conduct two broad operations on a state: we can let it *evolve unitarily* on its own without measurement, or we can apply a *measurement*.

B.1 Unitary Evolution

We can describe how a state changes as time goes on by defining a *unitary transformation*: the state $|\psi\rangle$ of a *closed* system at time t_0 is related to the state $|\psi'\rangle$ of the system at time t (where $t > t_0$) by a $N \times N$ unitary matrix operator U (that depends only on t_0 and t) such that [NC10]

$$|\psi'\rangle = U |\psi\rangle \,. \tag{27}$$

As an example, suppose we want to change (or *evolve*) our state $|\psi\rangle$ to $|\psi'\rangle$, such that

$$|\psi'\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle + \cdots + \beta_{N-1} |N-1\rangle \,. \tag{28}$$

We can represent this transformation with U as

$$U \begin{pmatrix} \alpha_0 \\ \vdots \\ \alpha_{N-1} \end{pmatrix} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_{N-1} \end{pmatrix}, \tag{29}$$

where $\sum_{j=0}^{N-1} |\beta_i|^2 = 1$. The transformation must be *unitary*, meaning that the matrix U must have an inverse U^{-1} equal to its conjugate transpose U^*.

B.2 Measurement

Since a state is defined by its amplitudes of classical states, attempting to observe, or “measure”, the state $|\psi\rangle$ will yield a singular classical state $|i\rangle$. Furthermore, we cannot deterministically predict which classical state we would observe. Rather, we have a *probability* of $|\alpha_i|^2$ to observe state $|i\rangle$.

This is where the aforementioned condition that $\sum_{j=0}^{N-1}|\alpha_i|^2 = 1$ stems from. An inherent property of quantum states is that if $|\psi\rangle$ is measured to i, then $|\psi\rangle$ itself "collapsed" to the classical state $|i\rangle$, thus destroying any information contained within the amplitudes α_j in the process. Hence, measurement of a quantum state is a *destructive* operation that we will leverage later on in the context of security.

Example 2. Suppose we have a state $|\psi\rangle$. For $N = 3$, we can have any of the following states (and the subsequent probability p of observing i):

$$\begin{aligned}
|\psi\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle + (0)|2\rangle \\
&\implies p(0) = p(1) = \frac{1}{2}, p(2) = 0 \\
|\psi\rangle &= \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{2}|1\rangle + \frac{1}{2}|2\rangle \\
&\implies p(0) = \frac{1}{4}, p(1) = p(2) = \frac{1}{4} \\
|\psi\rangle &= \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{2}|1\rangle + \frac{i}{2}|2\rangle \\
&\implies p(0) = \frac{1}{2}, p(1) = p(2) = \frac{1}{4}
\end{aligned}$$

We can more precisely define a *projective measurement* as described by projectors $P_1, \dots P_m$ (which sum to 1). The projector P_i projects onto some subspace $\mathcal{H}_i$ of the total Hilbert space $\mathcal{H}$, and every state $|\psi\rangle \in \mathcal{H}$ can be uniquely decomposed as $|\psi\rangle = \sum_{i=1}^{m}|\psi_i\rangle$, with $|\psi_i\rangle = P_i|\psi\rangle \in \mathcal{H}_i$. Upon measuring $|\psi\rangle$, the probability of observing i is

$$p(i) = \||\psi_i\rangle\|^2 = \langle\psi| P_i |\psi\rangle . \tag{30}$$

Subsequently, immediately after measurement of i the state of the system collapses to

$$\frac{|\psi_i\rangle}{\||\psi\rangle\|} = \frac{P_i|\psi\rangle}{\|P_i|\psi\rangle\|}. \tag{31}$$

In the computational basis, we have $m = N$ and $P_i = |i\rangle\langle i|$, meaning P_i projects onto the computational basis state $|i\rangle$. If we have the state $|\psi\rangle = \sum_{i=1}^{N-1}\alpha_i|i\rangle$ from above, then $P_i|\psi\rangle = \alpha_i|i\rangle$. The probability of measuring $|\psi\rangle$ to i will be

$$p(i) = \|\alpha_i|i\rangle\|^2 = |\alpha_i|^2, \tag{32}$$

and the state collapses to

$$\frac{\alpha_i|i\rangle}{\|\alpha_i|i\rangle\|} = \frac{\alpha_i}{|\alpha_i|}|i\rangle \to |i\rangle , \tag{33}$$

where we disregard the *normalization factor*, since it has no physical significance. Therefore, we reach the state $|i\rangle$ of our system that we previously described.

B.3 Superposition and Entanglement

We now consider two (or more) distinct physical systems that constitute a *composite* quantum system. Therefore the state space of a composite system is the tensor product of the state spaces of the component physical systems 0 through n. If system i is in the state $|\psi_i\rangle$, then the *joint state* of the whole system is

$$|\psi_0\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle \,, \tag{34}$$

where "⊗" corresponds to the tensor product. A *superposition* describes the addition of two (or more) states to produce another valid state. A quantum system in state $|\psi\rangle$ therefore simultaneously exists in all classical states, each with its own amplitude α_i. As an example, if we have a system with states $|\psi_0\rangle$ and $|\psi_1\rangle$, then any superposition $\alpha_0 |\psi_0\rangle + \alpha_1 |\psi_1\rangle$ is a valid state such that $|\psi\rangle = |\psi_0\rangle \otimes |\psi\rangle$.

A useful visual representation of superimposed states is the *Bloch sphere*. Consider the superimposed single qubit state

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle \,, \tag{35}$$

where $\alpha, \beta \in \mathbb{C}$. Since $|\alpha|^2 + |\beta|^2 = 1$, we can rewrite the above equation as

$$|\psi\rangle = e^{i\gamma} \left(\cos\frac{\theta}{2} |0\rangle + e^{i\varphi} \sin\frac{\theta}{2} |1\rangle \right) , \tag{36}$$

where $\theta, \varphi, \gamma \in \mathbb{R}$. The $e^{i\gamma}$ is referred to as a *global phase factor*. However, quantum mechanical states exhibit global phase *invariance*, and hence the phase no observable effects on the state, which allows us to write

$$|\psi\rangle = \cos\frac{\theta}{2} |0\rangle + e^{i\varphi} \sin\frac{\theta}{2} |1\rangle \,. \tag{37}$$

The angles θ and φ are spherical coordinates that define a point on the three-dimensional unit sphere shown in Figure 8. The poles represent classical bits, but the sphere itself represents all possible qubits of the state. Measuring $|\psi\rangle$ causes the system to collapse to one of the two poles depending on the relative "position" of the state. Note, this representation is restricted to a single-qubit system.

We can now define one of the most critical properties of quantum systems, *entanglement*.

Proposition 1. In the two qubit state

$$|\psi\rangle = \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle \right) , \tag{38}$$

no single qubit states $|a\rangle$ and $|b\rangle$ exist such that $|\psi\rangle = |a\rangle \otimes |b\rangle$. This particular state is denoted as the *Bell Basis* $|\Psi_+\rangle$, which we will refer to later.

Proof. Let $|a\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$ and $|b\rangle = \beta_0 |0\rangle + \beta_1 |1\rangle$, where $\alpha_i, \beta_i \in \mathbb{C}$. We assume

$$|\psi\rangle = |a\rangle \otimes |b\rangle = \left(\alpha_0 |0\rangle + \alpha_1 |1\rangle \right) \otimes \left(\beta_0 |0\rangle + \beta_1 |1\rangle \right) . \tag{39}$$

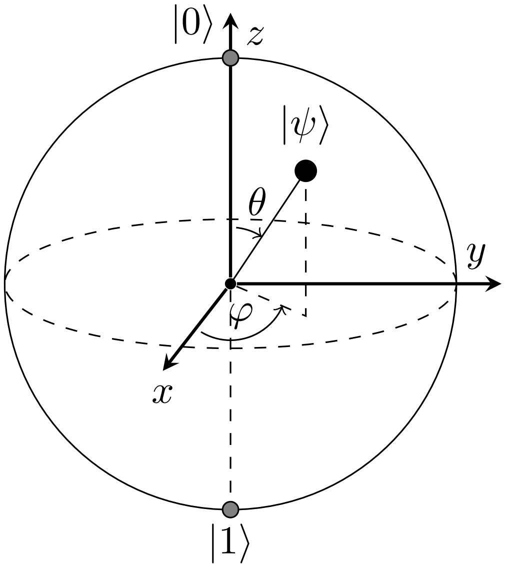


Figure 8: Bloch sphere representation of a qubit.

Applying the distributive property yields

$$|\psi\rangle = \alpha_0\beta_0 |00\rangle + \alpha_0\beta_1 |01\rangle + \alpha_1\beta_0 |10\rangle + \alpha_1\beta_1 |11\rangle , \tag{40}$$

which must be equal to $\frac{1}{\sqrt{2}} (|00\rangle + |11\rangle)$. Therefore, we must find values for $\alpha_0, \alpha_1, \beta_0$, and β_1 such that

$$\frac{1}{\sqrt{2}} (|00\rangle + |11\rangle) = \alpha_0\beta_0 |00\rangle + \alpha_0\beta_1 |01\rangle + \alpha_1\beta_0 |10\rangle + \alpha_1\beta_1 |11\rangle . \tag{41}$$

Our final expression must contain $|00\rangle$ and $|11\rangle$, so $\alpha_0, \beta_0 \neq 0$ and $\alpha_1, \beta_1 \neq 0$. The state $|\psi\rangle$ does not contain $|01\rangle$ or $|10\rangle$, so either α_0 or β_1 must be zero, as well as either α_1 or β_0. However, we already required these coefficients to be nonzero, thus reaching a contradiction of our original assumption. □

This conclusion can be trivially extended to N-qubit systems. A state that demonstrates this property is considered an *entangled* state. Entanglement is a uniquely quantum phenomena: it has no classical counterpart.

B.4 Quantum Secret Sharing

A quantum adversary $\mathcal{A}$ is defined as one who can eavesdrop on and manipulate a system through measuring states. For Parts 1 and 2, admittedly $\mathcal{A}$'s capabilities are weak. In the context of a (n, t)-threshold scheme, $\mathcal{A}$'s capabilities align much with the classical counterpart. If they obtain $t - 1$ or fewer shares, the adversary will not be able to learn any information about the secret quantum state [CGL99].

Theorem 1. *For any (n,t) threshold scheme with $n > t$, a $(n-1,t)$ threshold scheme can be constructed by discarding one share.*

Theorem 2. *If $n \geq 2t$, then no (n,t) threshold scheme exists.*

Theorem 3. *If a quantum code with code words of length $2t-1$ corrects $t-1$ erasure errors, the it is also a $(2t-1,t)$ threshold scheme.*

Corollary 1. *If a $[2t-1,1,t]_q$ code exists, a (n,t) threshold scheme exists for any $n < 2t$.*

Theorem 4. *If $n < 2t$, then a (n,t) threshold scheme exists. Furthermore, the dimension of each share can be bounded above by $2\max(2t-1,s)$, where s is the dimension of the quantum secret.*

We outline the procedure for share creation and reconstruction for a (n,t)-threshold scheme as follows.

Definition 2. ((n,t)-threshold quantum scheme) Let n and t be given with $n < 2t$, and let s be the dimension of the state to be encoded. Choose a prime q such that

$$\max(n,s) < q \leq 2\max(n,s), \tag{42}$$

which is guaranteed from Bertrand's postulate (*for every $n \geq 1$, there is some prime number p with $n \leq p \leq 2n$*). Let $\mathbb{Z}_q$ denote the finite field, and let $\mathbf{c} = (c_0, c_1, \ldots, c_{t-1}) \in \mathbb{Z}_q^k$ be some given coefficients. Let $|\psi\rangle$ be the q-ary secret quantum state we wish to share to n parties, such that

$$|\psi\rangle = \alpha_0 |x_0\rangle + \alpha_1 |x_1\rangle + \cdots + \alpha_{s-1} |x_{s-1}\rangle = \sum_{i=0}^{s-1} \alpha_i |x_i\rangle , \tag{43}$$

where each $x_i \in \mathbb{Z}_q$ are distinct. Lastly, we define the following polynomial

$$p_c(r) = c_0 + c_1 r + \cdots + c_{t-1} r^{t-1}. \tag{44}$$

We can now encode $|\psi\rangle$ by a linear mapping U defined on the basis states $|x_i\rangle$ as

$$U |x_i\rangle \mapsto \sum_{\substack{c \in \mathbb{Z}_q^k \\ c_{t-1} = x_i}} |p_c(x_0), \ldots, p_c(x_{n-1})\rangle = |x_i'\rangle \tag{45}$$

Each qubit $|x_i'\rangle$ constitutes one share, and are all subsequently distributed to the n parties.

To reconstruct the secret, we first recall the $d \times d$ Vandermonde matrix. The matrix is invertible iff each z_i is distinct for $i = 0, \ldots, d-1$. We also note that applying $V_d(z_0, \ldots, z_{d-1})$ to registers in the state $|c_0, \ldots, c_{d-1}\rangle$ will produce the state $|p_c(z_0), \ldots, p_c(z_{d-1})\rangle$. The procedure for reconstructing a state given t or more qubits is outlined as follows: each party applies the inverse Vandermonde matrix $V_t(x_0, \ldots, x_{k-1})^{-1}$ to their qubits into the shared state

$$\sum_{i=0}^{s-1} \alpha_i \sum_{\substack{c \in \mathbb{Z}_q^k \\ c_{t-1} = x_i}} |c_0, \ldots, c_k\rangle |p_c(x_k), \ldots, p_c(x_{n-1})\rangle . \tag{46}$$

Then, cyclically shift the first t registers to the right by one giving

$$\sum_{i=0}^{s-1} \alpha_i \left|c_{k-1}\right\rangle \sum_{\substack{c \in \mathbb{Z}_q^k \\ c_{t-1}=x_i}} \left|c_0, \ldots, c_{k-2}\right\rangle \left|p_c(x_k), \ldots, p_c(x_{n-1})\right\rangle . \tag{47}$$

Next, we apply $V_{k-1}(x_k, \ldots, x_{m-1})$, yielding

$$\sum_{i=0}^{s-1} \alpha_i \left|c_{k-1}\right\rangle \sum_{\substack{c \in \mathbb{Z}_q^k \\ c_{t-1}=x_i}} \left|p_c(x_k), \ldots, p_c(x_{n-1})\right\rangle \left|p_c(x_k), \ldots, p_c(x_{n-1})\right\rangle . \tag{48}$$

Lastly, we add $c_{k-1} \cdot (x_{k+i-1}^{k-1})$ to each register to obtain

$$\sum_{i=0}^{s-1} \alpha_i \left|c_{k-1}\right\rangle \sum_{y \in \mathbb{Z}_q^{k-1}} \left|y_1, \ldots, y_{k-1}\right\rangle \left|y_1, \ldots, y_{k-1}\right\rangle , \tag{49}$$

thus reconstructing the secret.

References

[CGL99] Richard Cleve, Daniel Gottesman, and Hoi Kwong Lo. How to share a quantum secret. *Physical Review Letters*, 83(3):648–651, 1999.

[HBB99] Mark Hillery, Vladimír Bužek, and André Berthiaume. Quantum secret sharing. *Physical Review A - Atomic, Molecular, and Optical Physics*, 59(3):1829–1834, 1999.

[IBM20] IBM. Ibm quantum experience, may 2020.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. Chapman and Hall/CRC, 2014.

[NC10] Michael Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[rig] rigetti/pyquil: A python library for quantum programming using quil. https://github.com/rigetti/pyquil. (Accessed on 02/24/2020).