The Effect of Noise on Quantum Secret Sharing

Emma Dauterman
Stanford University

Zoë Bohn
Stanford University

1 Introduction

We implemented and tested the effect of noise on the quantum secret sharing scheme outlined by Hillery et. al. [2]. Secret sharing is a cryptographic technique for splitting a secret between multiple parties such that a single party cannot learn any useful information, but when all the shares are assembled, the parties can reconstruct the original secret. Shamir famously proposed a classical mechanism for t-out-of-n secret sharing where some dealer who knows the secret can split the secret into n shares and distribute it to parties such that t shares are necessary to reconstruct the secret [3]. Secret sharing is used in many modern cryptosystems to distribute trust and remove single points of failure, which is useful in applications such as certificate authorities and cryptocurrencies.

However, it is well known that noise can limit the effectiveness of many quantum algorithms. Error-correcting codes such as Shor's code can help mitigate the effect of noise, but for some algorithms, such as Grover's search, error correction becomes so expensive that the algorithm is no longer practical. With this in mind, we set out to explore the practicality of Hillery's quantum secret sharing scheme by asking the following question: how does noise affect the failure rate of quantum secret sharing? To answer this question, we implemented Hilery's quantum secret sharing scheme and evaluated its failure rate under different types of noise, focusing on noise models typical of near-term devices.

2 Implementation

We used Pyquil to implement the quantum secret sharing algorithm as described in the paper by Hillery et. al. [2]. Specifically, after initializing three qubits to be in an entangled GHZ state and dividing them among three parties (Alice, Bob, and Charlie), each party can select a random measurement direction (x or y) such that Bob and Charlie can, working together, reconstruct Alice's measurement result, or, given Alice's measurement direction, determine that the trial did not succeed and restart the process.

The algorithm (both as described and as implemented) can be broken into four parts: 1) GHZ state preparation and qubit distribution, 2) measurement direction selection, 3) measurement application, and 4) measurement direction reveal. In the first step, in order to share a single bit of information, Alice prepares three qubits in an entangled GHZ state and gives Bob and Charlie each one of the qubits, retaining one for herself. Next, all parties randomly select a measurement direction (x or y). In the third step, they perform a measurement in this randomly selected direction on their respective qubits. Finally, they reveal to each other the direction in which they measured (the protocol specifies that this must be done such that Bob and Charlie send their directions to Alice first, who then sends all three directions back to both Bob and Charlie, to avoid cheating by either Bob or Charlie). At this point, Bob and Charlie can combine their measurement results to learn the result of Alice's measurement.

If Alice and Bob both randomly chose the same measurement direction, then Charlie must have measured in the x direction in order for him and Bob to learn the result of Alice's measurement. If they randomly chose different directions, then Charlie must have measured in y direction. Thus we can see that this protocol has a built in 50% failure rate, as Charlie chooses his measurement direction at random. However, such failures can always be detected by the participants, and therefore the protocol can simply be repeated as necessary.

3 Evaluation

We wanted to test the effect of noise on this protocol. Noise introduces additional points of failure; the protocol can now fail due to noise during the measurement application step in addition to randomness in the measurement selection step. Furthermore, unlike failures due to random measurement direction selection, failures due to noise are impossible to detect (at least, without modification to the existing protocol).

We evaluated the robustness of the quantum secret sharing protocol under noise by varying the different noise parameters of Pyquil's noise model interface: T1 and T2 values, gate times, and read-out fidelity. We were interested in how this protocol would perform under a noise model characteristic of near-term quantum devices (see Table 1), but we were also curious how increasing different noise parameters would affect the failure rate of the protocol. We ran four experiments varying four noise parameters: T1 and T2 values, 1-qubit gate times, 2-qubit gate times, and read-out fidelity. Because failures due to randomness of the algorithm are detectable (unlike failures due to noise), we discounted trials that failed due to randomness rather than noise.

[1]See pyquil noise documentation: `http://docs.rigetti.com/en/stable/apidocs/autogen/pyquil.noise._decoherence_noise_model.html#pyquil.noise._decoherence_noise_model`.

Noise type	Value
T1	$30\mu s$
T2	$30\mu s$
1-qubit gate time	50ns
2-qubit gate time	150ns
Readout-fidelity	95%

Table 1: Noise model typical of near-term devices [1]

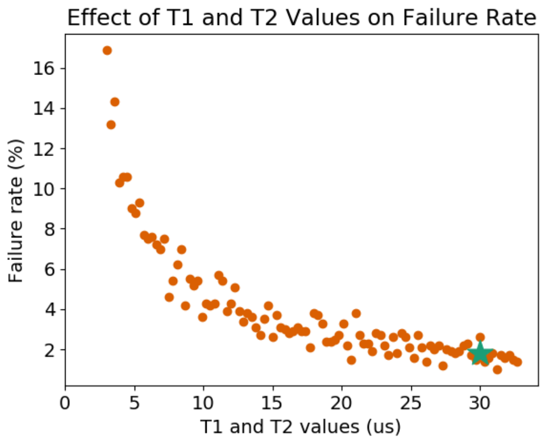

Figure 1: Failure rate as measured over 1,000 trials with different T1/T2 values. Other noise parameters set to the values in table 1 (with the exception of read-out fidelity, which is set to 100% to reduce variance between points). The green star represents the performance of devices with near-term T1/T2 values.

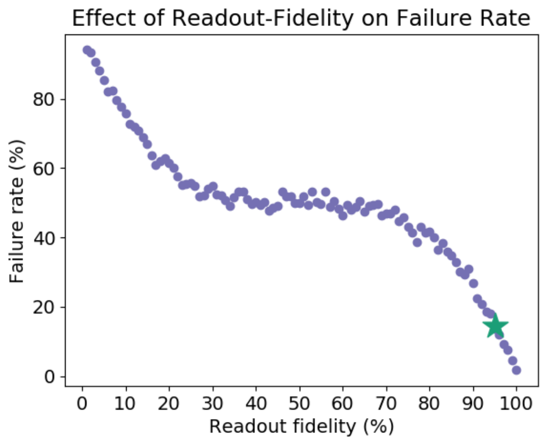

Figure 2: Failure rate as measured over 1,000 trials with different read-out fidelities values. Other noise parameters set to the values in table 1. The green star represents the performance of devices with near-term read-out fidelities.

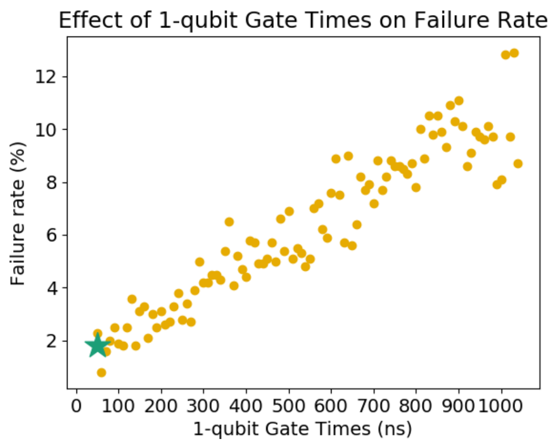

Figure 3: Failure rate as measured over 1,000 trials with different 1-qubit gate times. Other noise parameters set to the values in table 1 (with the exception of read-out fidelity, which is set to 100% to reduce variance between points). The green star represents the performance of devices with near-term 1-qubit gate times.

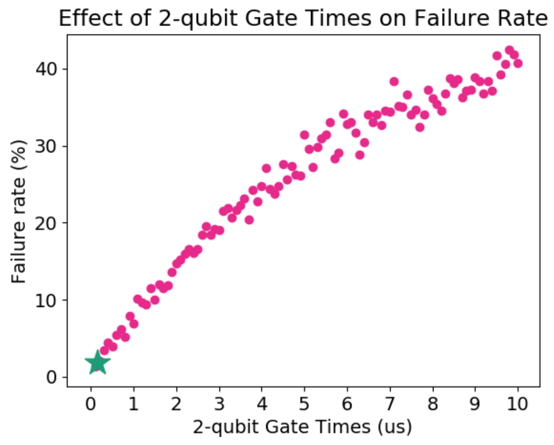

Figure 4: Failure rate as measured over 1,000 trials with different 2-qubit gate times. Other noise parameters set to the values in table 1 (with the exception of read-out fidelity, which is set to 100% to reduce variance between points). The green star represents the performance of devices with near-term 2-qubit gate times.

Failure type	Failure rate
Algorithmic	50%
Noise	14.4%
Total	64.4%

Table 2: Expected failure rate for near-term devices (assuming all parties are honest). Note that algorithmic failures can be easily detected, while noise can cause the protocol to fail silently.

Methodology: We measured failure rate over 1,000 trials (each dot in each plot represents 1,000 trials). For each experiment, we varied a single parameter and held the others constant at the values in Table 1. The only exception is read-out fidelity: we found that setting read-out fidelity to 95% increased variance between points, making it more difficult to capture trends, and so we set read-out fidelity to 100% (except when measuring the effect of read-out fidelity). We used the values in Table 1 as a starting place for our noise parameters, and then observed the effects of increasing these noise parameters. We varied T1 and T2 values together because in real quantum devices, these values are not independent, and Pyquil limited the T1 values that could be chosen based on T2 and vice versa. To make these measurements, we compiled our program to a native gate set and used the Pyquil `add_decoherence_noise` feature.

Results: We found that our protocol was most sensitive to readout-fidelity: with readout-fidelity set to 100% and all the other parameters set to the near-term values in Table 1, the error rate was only 1.8%, but setting readout-fidelity to 95% caused the error rate to jump to 14.4%. This makes sense given that very few gates are required to share a single bit (and therefore noise introduced through gates has a smaller chance of disrupting the protocol). More importantly, Bob and Charlie reconstruct the original secret through measurements. If both measurements must be correct for the protocol to succeed and each measurement has a 5% chance of being incorrect, it makes sense that reducing the readout-fidelity from 100% to 95% causes a roughly 10% increase in error rate.

In Table 2, we show the expected failure rate for near-term devices, broken into failures that are inherent in the algorithm and failures introduced by noise. We find that the error rate introduced by noise is much smaller than that introduced by the algorithm itself. However, as previously described, unlike the potential error due to randomness in measurement direction choice in the original algorithm, errors introduced by noise cannot be easily detected by Bob and Charlie. Thus Bob and Charlie cannot easily tell if they reconstructed Alice's measurement result correctly.

Reproducing our results: You can reproduce our results using the code at `https://github.com/zoebohn/cs269q_secret_sharing`. Launch the Rigetti QVM using `qvm -S` and the Quil Compiler using `quilc -S`. To run all experiments, run `python3 run_eval.py`, and to generate the plots, run `python3 plot_data.py`.

4 Future Work

Error-correction codes: As discussed earlier, one line of future work would be to explore the effectiveness of error-correction codes and their impact on the efficiency of this protocol. How much can we reduce the failure rate due to noise, and what is the overhead associated with this reduction? While we are curious to see what effect error-correction codes could have on this protocol, we are not overly optimistic: as our evaluation shows, readout-fidelity is responsible for a large percent of the failure rate expected on near-term devices, and error-correction codes cannot remedy this problem. However, error-correction codes could help reduce the failure rate introduced by other types of noise, and if we could build machines with high readout fidelities, then this could have a significant impact on failure rate.

Other quantum secret sharing schemes: Another line of work would be to explore the effect of noise on other quantum secret sharing schemes, such as the scheme proposed by Cleve [1] and more modern schemes such as the one proposed by Xiao [4]. Here it would be interesting to examine the relationship between algorithmic failure rate (if any) and noise failure rate. Are there algorithms that have a 0% algorithmic failure rate but at the cost of greater noise failure rate on near-term devices? Are there algorithms that achieve better algorithmic failure rates and better noise failure rates? Can error correction be applied to these algorithms with relatively low overhead? We think that addressing these questions would lead to a better understanding of best practices for quantum secret sharing protocols in the presence of noise.

5 Conclusion

We evaluated the effect of noise on the quantum secret sharing scheme outlined by Hillery et. al. [2]. We found that a near-term quantum device would experience almost a 15% increase in failure rate due to noise, and that this increase would be primarily due to imperfect readout fidelity. As such, error-correction codes may be of limited use. Furthermore, without making changes to the protocol, simply running the protocol multiple times and taking the majority vote will also not work, as the security of the protocol relies on all parties picking their measurement directions at random, and thus Alice's measurement result should change on successive runs. Therefore, making this protocol useful in the near-term is an area of future work that remains of interest.

Our code is already public (available at `https://github.com/zoebohn/cs269q_secret_sharing`) and we would prefer for our report to not be made available publicly.

References

[1] R. Cleve, D. Gottesman, and H.-K. Lo. How to share a quantum secret. *Physical Review Letters*, 83(3):648, 1999.

[2] M. Hillery, V. Bužek, and A. Berthiaume. Quantum secret sharing. *Physical Review A*, 59(3):1829, 1999.

[3] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[4] L. Xiao, G. L. Long, F.-G. Deng, and J.-W. Pan. Efficient multiparty quantum-secret-sharing schemes. *Physical Review A*, 69(5):052307, 2004.