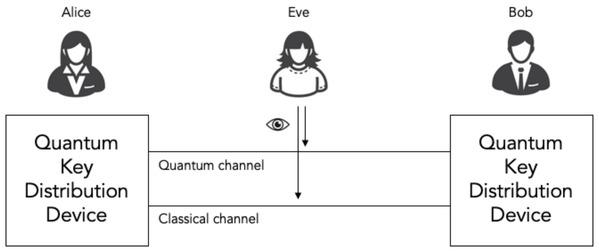
Demonstrating and Evaluating Quantum Key Distribution

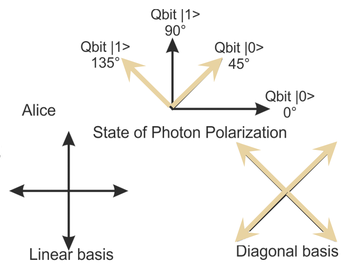
Suhani Jain, Nick Yama, Prof. Kai-Mei Fu

INTRODUCTION

- Quantum Key Distribution (QKD) is a fundamentally-secure method of sharing a key, which is used for encryption



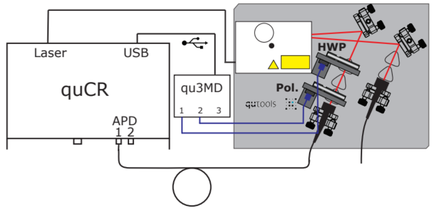
- Current public-key encryption isn't unconditionally secure – an efficient algorithm for number factoring could break it [1]
- BB84 – Bennett and Brassard: encodes the key bits into a single-photon polarization state
- Alice and Bob randomly chose basis to measure photon in – get corresponding bit



- Detect eavesdropper "Eve" by using a single photon source : No-Cloning Theorem and can't measure photon without destroying it [2]

GOALS:

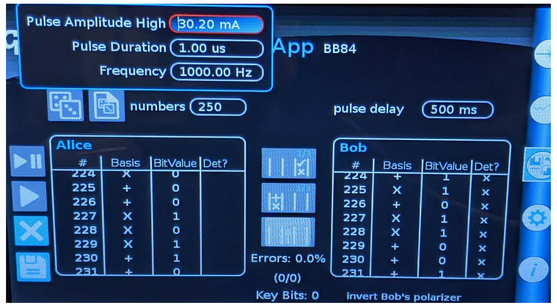
- Perform QKD on quED
- Find parameters for single photon source
- Test effectiveness of information reconciliation



[3] Set up of QKD on quED with polarizer and half-wave plate

METHODS

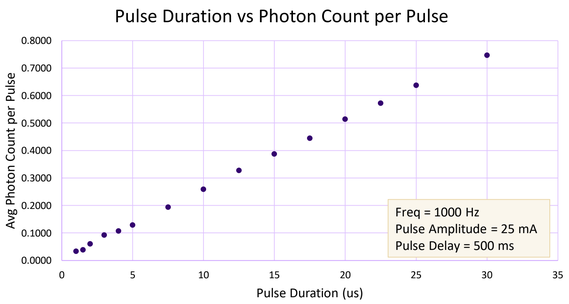
- BB84 Protocol was run on the quEd from quTools
- Randomly generates list of basis and bit values for Alice and Bob to measure in



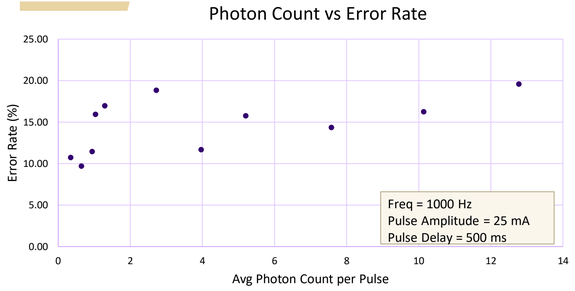
- The quED outputs a data file with the information shared in the quantum channel
- Python code can simulate the classical channel – processing the key and error reconciliation

SINGLE PHOTON SOURCE

- Detectors have 10% efficiency
- 1 photon/pulse ⟶ pulse duration = 4 μs
- Is theoretically secure but quED is an avg of one photon/pulse, so Eve could be undetected



ERROR:



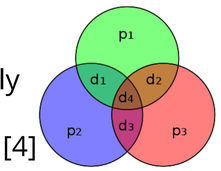
- 15.96% error between Alice's and Bob's key for one photon/pulse
- Higher error, harder to get same key

INFORMATION RECONCILIATION:

- Finds and corrects errors in Bob's key

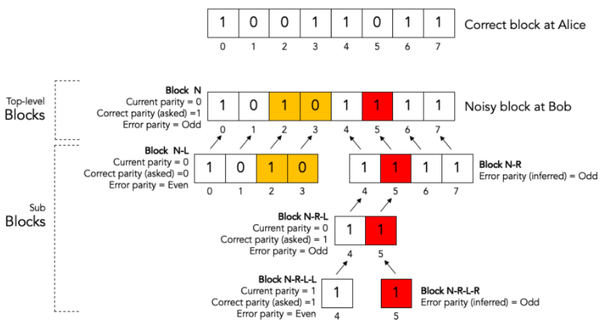
HAMMING (7, 4):

- Parity check code
- Not suitable since it only corrects at most 2 error bits per 7 total bits [4]



CASCADE PROTOCOL:

Uses recursive methods to find and fix errors by parity checks for random subsections



[5] Key is split into sections and Bob checks parity of section with Alice

Example:

Original Image that Alice wants to encrypt:

Image with shared key added:

Image decrypted by Bob with unreconciled key vs reconciled key:

NEXT STEPS

Privacy amplification:

- Compensates for information leakage
- Generates new key from old key so Eve has negligible information about key

ACKNOWLEDGEMENTS

This project funded in part by the WA NASA Space Grant

REFERENCES:

[1] https://cascade-python.readthedocs.io/en/latest/protocol.html
[2] https://www.researchgate.net/figure/Polarization-basis-of-the-protocol-BB84_fig2_324460751
[3] https://www.qutools.com/files/quED/quED-QKD-manual.pdf
[4] https://upload.wikimedia.org/wikipedia/commons/b/b0/Hamming%287%2C4%29.svg
[5] https://cascade-python.readthedocs.io/en/latest/protocol.html