Secure Shell Communication with Quantum Key Distribution

Statement of Work

Tristan Austin

August 2023

1 Introduction

Secure encrypted communication is a crucial pillar of the modern global internet. Without internet security, credit cards numbers, personal information, and scandalous images would be accessible to anyone with a keyboard and an internet connection. Identity theft would run rampant and paid fan sites would no longer be profitable. This privacy that is valued so much, can be attributed to the power of cryptography. Cryptographic key pairs facilitate identity verification between servers and clients ensuring data is being transmitted to verified devices. Key pairs are also used in terminal applications through secure shells (SSH). Key pairs can also be used to encrypt information during data transfer. However, in the advent of quantum information processing, one of the most crucial cryptographic algorithms, RSA encryption, is at risk of being broken. To ensure network privacy, new ways of securely encoding information will be necessary for the future of communication. As quantum computing transitions from sci-fi to reality, communication algorithms will need to be make similar changes. One possible change is the adoption of a quantum secure encoding scheme called quantum key distribution (QKD). This algorithm utilizes the same quantum power that will crack RSA encryption to generate secure key pairs.

2 Brief Background

2.1 Quantum Key Distribution

QKD is a key generation algorithm that utilizes a unique feature of quantum mechanics to transmit an encryption key that cannot be intercepted. This unique feature is referred to as the no-cloning theorem. Once a quantum object has been measured, it spontaneously collapses to one value and it's original state cannot be reproduced. A quantum bit (qubit) once measured, cannot be reproduced ensuring that an eavesdropper has not been listening while the key is being transmitted. Once a key pair has been generated using QKD it can be used for communication protocols such as SSH.

There are several QKD algorithms which have various benefits and drawbacks. The first and most famous algorithm is called BB84 and follows the following procedure. There are three agents, Alice and Bob who are attempting to generate a secure key, and Eve who is attempting to eavesdrop on their communication to determine the key. Alice and Bob are trying to communicate securely while Eve is trying to intercept this communication. Alice and Bob share a quantum communication channel, information represented as qubits, and a classical communication channel, regular binary data. Eve is able to listen to both communication channels. The identity of both Alice and Bob is confirmed using a previously known secret or a piece of a previous key they have generated. Alice begins by generating a random sequence of qubits that are initialized as either a one or a zero. She then applies a random of sequence of two gates which transform the original sequence of bits so they are encoded in two different random bases. A measurement of these bits in the correct basis will return the original value, a measurement in the wrong basis will return a random binary value. This sequence of bits are transmitted to Bob who measures each bit in a random basis. Half the measurements will return the correct bit Alice generated and half the measurement will return a random bit. Alice and Bob then share with each other the random bases they selected for each measurement discarding all bits that were measured incorrectly. Approximately half the bits will be discarded leaving Alice and Bob with

a shared key. They can then compare a small subset of the remaining bits over the classical channel and verify Bob received the correct bit when measuring in the correct basis. A diagram depicting the entire key generation process can be seen in Figure 1. The successful resulting key can be seen in the bottom row of the table. In the situation where Eve was **not** listening to the quantum channel, Alice and Bob will have identical bit strings. In the case where Eve was eavesdropping they will notice discrepancies between their bit strings provided they compare enough bits. Comparing 72 bits has a probability of 99.9999999% of being completely secure. By using this key as a password or shared key over SSH they can then verify each other's identities.

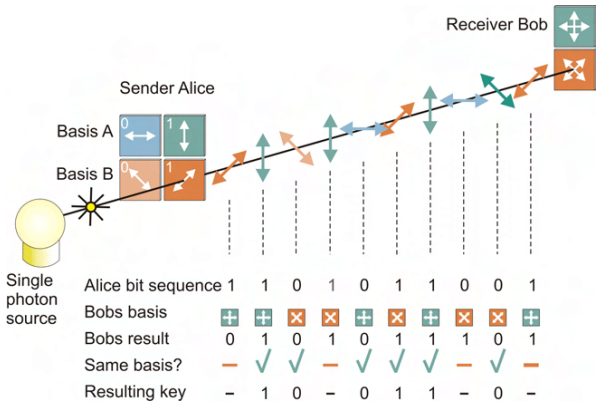

Figure 1: Diagram depicting the key exchange process for BB84 [1].

The crux of this algorithm is that qubits can be encoded or measured in different bases. If the encoding and measurement basis do not match than the result is random. A truth table describing this can be seen in Table 1

Table 1: Qubit QKD truth table.

Initial Qubit	Encoding Basis	Measurement Basis	Measurement Result
$\|0\rangle$	Z	Z	$\|0\rangle : 100\%$
$\|0\rangle$	H	H	$\|0\rangle : 100\%$
$\|1\rangle$	H	H	$\|1\rangle : 100\%$
$\|1\rangle$	Z	Z	$\|1\rangle : 100\%$
$\|0\rangle$	Z	H	$\|0\rangle : 50\%$ $\|1\rangle : 50\%$
$\|0\rangle$	H	Z	$\|0\rangle : 50\%$ $\|1\rangle : 50\%$
$\|1\rangle$	Z	H	$\|0\rangle : 50\%$ $\|1\rangle : 50\%$
$\|1\rangle$	H	H	$\|0\rangle : 50\%$ $\|1\rangle : 50\%$

2.2 Secure Shell

SSH allows users to remotely connect to servers or external machines over the internet. Instead of interacting with a webpage or graphical interface, the user utilizes commands typed into a terminal to interact with the machine. SSH is widely used in information technology, high performance computation, and server development. Typically a user attempting to connect through SSH verifies their identity through a password or a key pair. One key in the pair is private, the other public. Confirmation is done by comparing the keys using an algorithm that is based on factors of large prime numbers, RSA. Shor's algorithm is capable of factoring these numbers using quantum computers and thereby breaking the encryption. SSH communication through keys generated using QKD would be nearly impossible to crack.

A combination of the widely used remote access service with an un-hackable QKD algorithm would then allow superior security in a post-quantum world. User's could interact with remote devices securely and information could be transmitted with the utmost privacy. Quantum computers that

would be capable of breaking RSA encryption are about 10-20 years away and a proactive transition to alternative encryption algorithms, such as QKD, will be important for ensuring internet security.

3 Minimum Viable Product (MVP)

Quantum communication channels over fibre optics have been established experimentally over several kilometres while wireless quantum communication using satellites is currently being tested. Neither of these communication channels are publicly available and free. A real quantum communication channel will not be possible during the course of this project. However, simulated communication can be done by passing simulated quantum objects over a network. The **MVP** for this project will then be to implement a simulated QKD protocol for generating keys and communicating using SSH. Simulated quantum objects will be passed from user to user or user to server and used for generating the keys. These keys will then be used to verify and encrypt information during communication. An eavesdropper will be able to intercept the data, measure, and then pass on the bits to the intended user. This key verification system will be integrated directly in a custom version of the SSH client.

3.1 Scope Changes

The first reduction to the scope will be on native compatibility to SSH servers. Open-ssh is more than 23 years old and adapting this source code to function with QKD will be quite difficult. Instead, it is possible to make ssh connections using Python and key verification can be done using Python scripts. The next reduction of scope will be the removal of the eavesdropper from the networking. Instead of having a third user intercept the messages between Alice and Bob, the presence of an eavesdropper can be specified during the key generation step. This can then be used to demonstrate what a failed QKD algorithm will look like. The third and final scope reduction will be to remove the networking component completely. Instead, local sockets, on one machine, will be used to simulate a remote connection. Data will be passed between different instances of the same program on one machine to demonstrate SSH using QKD.

3.2 Methods

Quantum Python libraries such as Qiskit [2] or Pennylane [3] will be used to simulate the quantum circuits required for QKD. Initially, this will just be the BB84 algorithm, however many quantum algorithms can be investigated in the future. The Python socket library can be used for communication after quantum circuit objects have been converted to byte streams using the pickle package. Sockets can be used to communicate both locally and over a network. A custom version of Open-ssh will be used to test the keys after generation for both identity verification and encryption.

The team can be partitioned into three groups each working on a different element of the project. One group will make the code for creating the quantum circuits required for QKD, one group will write the code for sending the quantum circuit objects to different users over the internet, and one group will work on integrating custom keys into open-ssh. Each group will be able to write code that can function individually of the other two groups and have possibilities to go beyond the original MVP. For example, the group working on QKD can develop versions for different algorithms, circuits running on real quantum hardware, and circuits that include the effects of noise or uncertainty. An early step in this project will be to determine which data types will be passed to each of these separate components and how they will all be integrated together. Following this each group will be able to complete a majority of their work separately. To coordinate between each group Github issue tracking and a Trello board will be used.

Github and Trello provide an excellent platform for managing code based projects. Work that still needs to be completed can be added to the issues board in Github and assigned to members. Github also has direct integration with Trello to manage large milestones as code gets completed. Trello will facilitate task delegation from a higher level than the Github issue board. Tasks that involve self education, presentation preparation, or writing will be assigned using Trello. Using these two time management techniques team members will have tasks with clearly defined goals as well as a timeline for completion.

4 Onboarding

The team will initially be onboarded all together before transitioning to groups. Every team member will be given a high level understanding of quantum computing, QKD, and SSH. They will also be taught the importance of encryption and post quantum cryptography in a future where the RSA algorithm is no longer secure. Every team member will get guided hands on experience in either Qiskit or Pennylane (quantum computing packagees), network communication using Python, and connecting and using SSH. There are numerous tutorials and educational resources for learning about QKD and programming quantum algorithms. Similarly, network communication using Python has extensive documentation. Team members will also become familiar with navigating using the terminal, writing simple bash scripts, and setting up SSH verification using generated key pairs. Following this initial onboarding, the team will be split up into groups based on interest and knowledge. It is at this point where team members will become independent in learning the required material for completing their portion of the project. I will provide guidance when I can and give groups deadlines for when certain features need to be completed.

5 Timeline

The timeline seen in Table 2 outlines the timeline for completion of the project. This assumes each team member will on average be spending 5-10 hours per week on the project.

Table 2: Timeline for Completion of The Project

Task Name	Category	Expected Time (Weeks)	Members Required
Intro to Quantum	Education	1	6
Intro to bash, shell, and SSH	Education	1	6
Intro to networking in Python	Education	1	6
How to use Git	Education	1	6
RSA Encryption and QKD	Education	1	6
Local QKD Script	Development	2	2
Python Local Socket	Development	2	2
SSH Source Code Dive	Development	3	2
Alternate QKD Algorithms	Development	2	2
QKD Run On Quantum Hardware	Development	2	2
Custom RSA Algorithm With SSH	Development	2	2
Python Network Sockets	Development	2	2
Sending Quantum Circuit With Socket	Development	2	2
Intercepting QC with Eve	Development	3	2
Running QKD Using Sockets	Testing	2	4
Integrate QKD With SSH	Development	3	2
Full System Test	Testing	2	6
Prepare Figures Describing System	Communication	2	6
Prepare Slide On Project	Communication	2	6
Prepare Final Report	Communication	2	6
Publish Report On Internet	Communication	2	6
	Sum of Weeks 40 Weeks	Total Work Time 144 Weeks × People	**Total Project Time** **24 Weeks**

Since the project has many components that can be worked on simultaneously, it will take roughly 24 weeks, or 6 months to complete. This fits within the timeline from team hiring to CUCAI. A more detailed task breakdown will be completed when the team size is confirmed.

References

[1] E. Stock, "Self-organized Quantum Dots for Single Photon Sources," Jan. 2011.

[2] M. Treinish, "Qiskit/qiskit-metapackage: Qiskit 0.44.0," July 2023.

[3] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi, J. M. Arrazola, U. Azad, S. Banning, C. Blank, T. R. Bromley, B. A. Cordier, J. Ceroni, A. Delgado, O. Di Matteo, A. Dusko, T. Garg, D. Guala, A. Hayes, R. Hill,

A. Ijaz, T. Isacsson, D. Ittah, S. Jahangiri, P. Jain, E. Jiang, A. Khandelwal, K. Kottmann, R. A. Lang, C. Lee, T. Loke, A. Lowe, K. McKiernan, J. J. Meyer, J. A. Montañez-Barrera, R. Moyard, Z. Niu, L. J. O'Riordan, S. Oud, A. Panigrahi, C.-Y. Park, D. Polatajko, N. Quesada, C. Roberts, N. Sá, I. Schoch, B. Shi, S. Shu, S. Sim, A. Singh, I. Strandberg, J. Soni, A. Száva, S. Thabet, R. A. Vargas-Hernández, T. Vincent, N. Vitucci, M. Weber, D. Wierichs, R. Wiersema, M. Willmann, V. Wong, S. Zhang, and N. Killoran, "PennyLane: Automatic differentiation of hybrid quantum-classical computations," July 2022. arXiv:1811.04968 [physics, physics:quant-ph].