Post-Quantum Cryptography VPN Application: Python Implementation

COMP8037 – Major Project Proposal

Kuan-Yu(Gray) Chen – A01088589

11-20-2022

Table of Contents

1. Student Background

I am an international student from Taiwan and am currently in the Bachelor of Technology in the Computer Systems program. Before joining the bachelor's program, I completed the Computer Systems Technology Diploma and have two years of programming experience. Currently, I specialize in network security applications development, and I will be using those experiences and knowledge to take advantage of learning new application and encryption protocols.

1.1. Education

British Columbia Institute of Technology

- Bachelor of Technology – Network Security Applications Development   Sep 2021 – Current
- Computer Systems Technology Diploma – Web and Mobile   Jan 2019 – Dec 2020

University of British Columbia

- Bachelor of Science – Applied Biology   Sep 2017 – Apr 2018

1.2. Projects

- Search Engine Website for Recycling for City of Vancouver   Jan 2019 – April 2019
  - ➢ Team member, BCIT
- Mobile-friendly Web Game   Apr 2019 – May 2019
  - ➢ Team member, BCIT
- Cross-Platform Mobile Application for Voting   Apr 2020 – May 2020
  - ➢ Team member, BCITSA
- Mobile Application for Construction   Sep 2020 – Dec 2020
  - ➢ Team member, BICC Professionals

2. Project Description

Post-quantum cryptography, which can be referred as quantum-resistant cryptography or quantum-safe cryptography, aims to construct public key cryptosystems that remain secure from attackers with quantum computers. This project's goal is to create a VPN application that integrates post-quantum cryptography, specifically the **CRYSTALS-KYBER Algorithm,** with the **OpenVPN** protocol. By implementing this application, users will be able to use the internet with their data encrypted and secured even after quantum computers are built. This application will consist of two parts: VPN Client and VPN Server. The VPN Client will encrypt the data from the user by applying CRYSTALS-KYBER algorithm and sends the traffics through the VPN Tunnel to the VPN Server. On the other side, the VPN Server will listen to communications on specified port to receive data and decrypt the received data. It will also make requests to the internet with encryption while hiding the client's address and geolocation.

3. Problem Statement and Background

3.1. Background

As the internet grows, companies start to move their businesses online, allowing customers to do transactions online. However, websites become vulnerable once their confidential information, such as credit card numbers, passwords, or social insurance numbers, are leaked to cyber criminals. To prevent cyber-attacks, cryptography plays an essential and irreplaceable role in the security of all internet communications.

Nowadays, public key cryptography is widely used for securing data. The most common example of public key cryptography is the security of "HTTPS" (Hypertext Transfer Protocol Secure) web pages. When the customers are making the transactions through the web page, Transport Layer Security (TLS) or Secure Sockets Layer (SSL) protocols are applied to encrypt the exchanged data with their own algorithms. However, all the public key algorithms are based on complicated mathematical problems, such as discrete logarithms or prime factors of a composite number. Currently, those mathematical problems are almost impossible for classical computers to solve. This means that if a technology is developed to solve those mathematical problems, the data that are encrypted with the problems will no longer be safe.

Currently, scientists and engineers use supercomputers to solve difficult problems. However, most of the supercomputers are very large. They are built with thousands of classical CPU and GPU cores, which makes them so longer "super". Also, if the problem has a high degree of complexity, such as containing a very large number of variables, supercomputer will often fail to solve, which makes it no longer "super". The technology that are rising for solving such high complexity problems is called quantum computing. Quantum computing is a rapid-emerging technology that are developed to solve problems that are too complex for classical computers, including supercomputers. The computers that are built based on this technology is called quantum computers.

Fortunately, quantum computing is still a fairly new technology. Therefore, quantum computers have not existed and not been built in the world yet. However, by the time when quantum computers are built, they can easily break most of the problems that are generated by the current encryption algorithms. As a result, all data that are encrypted with the current encryption algorithms are no longer safe. To prevent this to happen, quantum-safe cryptography emerged. Quantum-safe cryptography aims to develop algorithms that are safe enough to prevent from attacks by both classical and quantum computers. One way of the implementation is to use mathematical techniques such as error correcting codes, lattices, or multivariate equations to develop quantum-safe cryptography (Open Quantum Safe, 2022).

3.2. Problem

The problem my application will solve is that users do not have to worry about the security of their data anymore, even when quantum computers are built. With the combination of the current encryption protocol and post-quantum cryptography, the application not only secures the user's data from classical computers but also prevents future threats when quantum computers are built and widely used by attackers. **By implementing post-quantum cryptography into my application, the VPN application will allow users to keep their data secure against both quantum and classical computers.** The protocol I will use to develop my VPN application is the OpenVPN protocol, which is the one of the most widely used protocol for VPN applications. The algorithm I will use for implementation is the CRYSTALS-KYBER Algorithm, which is one of the algorithms that are selected by the NIST (National Institute of Standards and Technology) for Post-Quantum Cryptography in 2022.

4. Scope and Depth

The scope of this project is to build a VPN application that secures the connections between the client and the server by implementing post-quantum cryptography algorithms. There is a set of functional requirements for the application to implement to ensure the security of the communications. There will also be a set of non-functional requirements for the application that the core functionality does not rely on them. Lastly, there will be a set of requirements that are out of scope of this project.

4.1. Functional Requirements

- The application will be built and ran on Linux.
- The application will implement post-quantum cryptography with CRYSTALS-KYBER algorithm for key encapsulation mechanism (KEM).
- The application will be built using the OpenVPN protocol that uses OpenSSL to secure communications.
- The application will contain two parts: Client and Server
- The application will be built with a basic GUI.
- PyQt will likely be used for building the GUI for the application, but not decided
- In the Client application, there will be configurations for entering the IP address and port number of the server.
- In the Server application, there will be configurations for entering the port number to run the applications on.

4.2. Non-Functional Requirements

- The server can handle at least two clients at a time.
- The application will contain a user manual.
- The application will write all traffic to a log file on the Server.

4.3. Out of Scope

- The application will be cross-platform.
- The application can handle different protocols other than OpenVPN.
- The application can forward traffics only from specific application.
- The application will implement other algorithms of post-quantum cryptography such as FALCON or SPHINICS.
- The application detects attempted attacks such as port scanning or spoofing.
- Optimize performance to reduce packet loss or delay when connecting to multiple clients.

5. Test Plan

To test this application thoroughly in different phases, the testing of this project involves the combination of manual testing, acceptance testing, and security testing. The test cases are listed below.

5.1. Test Cases

VPN with OpenVPN Protocol

Test Case Number	Description
#1	Traffics are forwarded through the tunnel to the server. Pass: The server receives to the network request. Fail: The server does not receive the network request.
#2	The client's address and geolocation are hidden. Pass: The IP address on the internet is the server's address. Fail: The IP address on the internet is the client's address.
#3	Two or more clients can secure their addresses and geolocations. Pass: The IP addresses on the internet for all clients are all the server's address. Fail: One or none of the IP addresses are the server's address.
#4	Data are secured with encryption. Pass: The tokens are all encrypted. Fail: The tokens are not encrypted.

CRYSTALS-KYBER Algorithm Implementation to VPN

Test Case Number	Description
#1	Traffics are forwarded through the tunnel to the server. Pass: The server receives to the network request. Fail: The server does not receive the network request.
#2	The client's address and geolocation are hidden. Pass: The IP address on the internet is the server's address. Fail: The IP address on the internet is the client's address.
#3	Two or more clients can secure their addresses and geolocations. Pass: The IP addresses on the internet for all clients are all the server's address. Fail: One or none of the IP addresses are the server's address.
#4	Data are secured with encryption. Pass: The tokens are all encrypted. Fail: The tokens are not encrypted.
#5	Data is quantum safe. Pass: The encrypted data cannot be decrypted by algorithms that are used in the previous implementation (OpenVPN protocol). Fail: The encrypted data can be decrypted by algorithms that are used in the previous implementation.

GUI Integration

Test Case Number	Description
#1	Client can connect to specified IP address with port number without crashes. Pass: GUI successfully responds to the client to each input. Fail: GUI does not respond to the client to each input.
#2	Server can run the service on specified port. Pass: GUI successfully responds to the server with the result (port is available/port is used). Fail: GUI does not respond to the server or crashes if port is currently used.
#3	Traffics are collected in a log file. Pass: GUI writes all traffics to a log file. Fail: GUI does not write all the traffics to the log file or the log file is not created.

6. Methodology

6.1. Methodology and Approach

The methodology I use for this project will be a simplified Agile approach. Agile approach is known for developing software in iterative and incremental way. This allows developers to manage their code so that it is always ready for adding more features on top of the existing code. Each cycle of an agile approach will involve the following phases:

1. Planning of Concept
2. Design
3. Implementation
4. Testing
5. Documenting

The reason I choose this approach is because this approach is based on building a minimum viable product (MVP) within each cycle. It is important for my application to function normally before adding extra requirements such as the OpenVPN protocol and the CRYSTALS-KYBER algorithm for post-quantum cryptography. The development of the application will be divided into the follow phases:

1. Building a VPN with the OpenVPN Protocol
2. Implementing CRYSTALS-KYBER algorithm into the VPN application
3. GUI Integration

As a result, the final product of this project will be a VPN application developed with the OpenVPN protocol and encrypted by the CRYSTALS-KYBER algorithm. There will be a client software to enter the server's IP address to start the VPN application. There will also be a server application that will set up the device as a server to listen to connections from the clients, encrypt and decrypt the communications, and forward the communications back and forth.

6.2. Tools and Technologies

For this project, the following tools and technologies will be used:

- **GitHub**: used for source code
- **Git**: used for version control
- **Python 3**: used for coding and testing
- **PyCharm**: Integrated Development Environment (IDE) for Python 3
- **Draw.io**: used for creating diagrams
- **Google Docs and Microsoft Word**: used for documentation

7. System/Software Architecture Diagram

The following diagram (Figure 1) is the system diagram for the application.

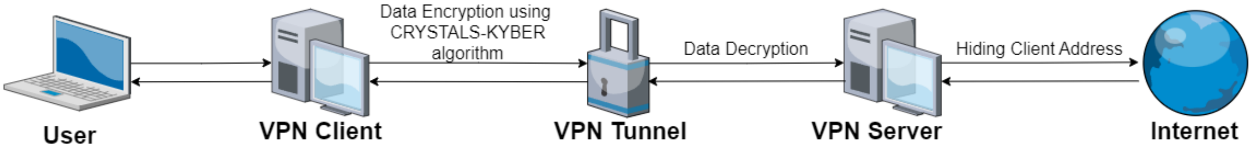

Figure 1: System Diagram of post-quantum cryptography VPN

The application is consisting of two parts:

1. **VPN Client**: VPN Client will be running on the user's device. It will encrypt the data from the user by applying CRYSTALS-KYBER algorithm and sends the traffics through VPN Tunnel to the VPN Server.
2. **VPN Server**: VPN Server will be running on the server's device. It will listen to communications on specified port and receive data. Afterward, it will decrypt the received data and make requests to the internet while hiding the client's address and geolocation.

The application will be developed with the **OpenVPN protocol** and encrypted by the **CRYSTALS-KYBER** algorithm. The security of the CRYSTALS-KYBER algorithm is based on the harness of solving the learning-with-errors (LWE) problem over module lattices.

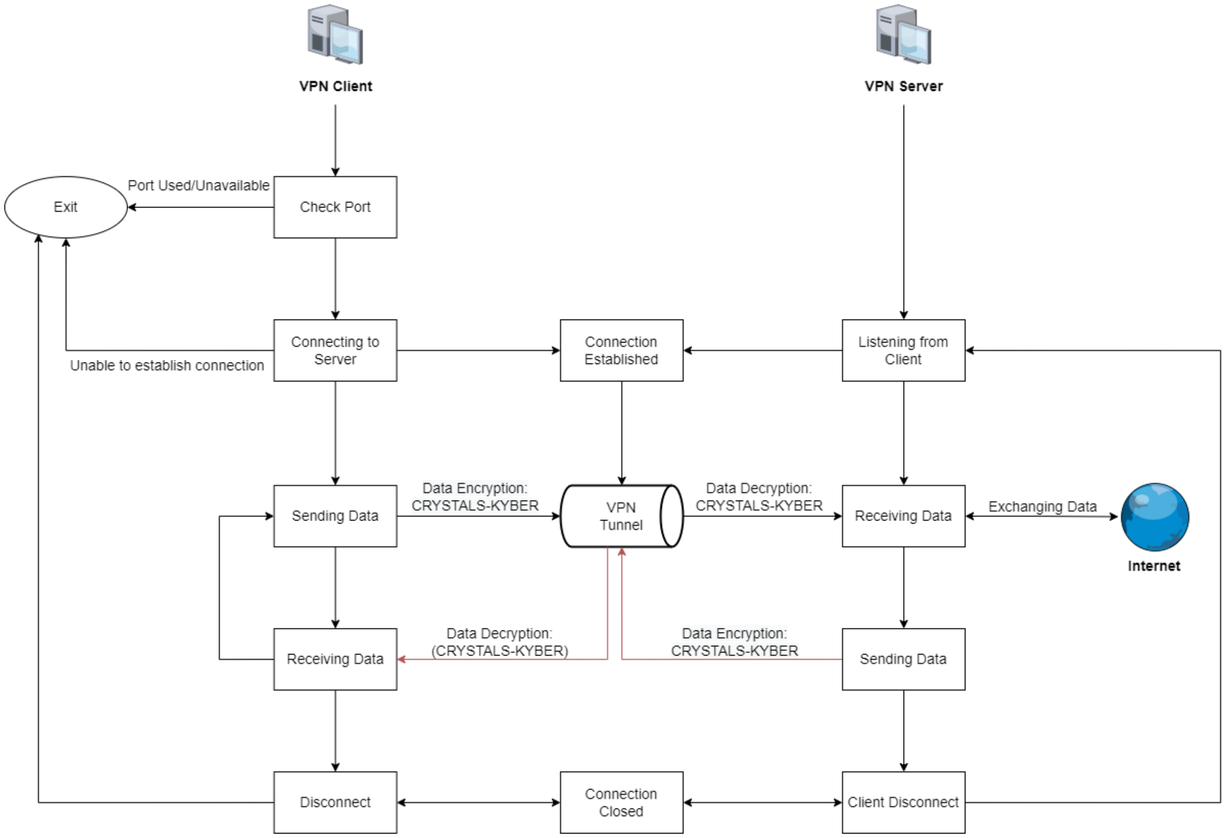

Figure 2: Software Architecture Diagram of post-quantum cryptography VPN

7.1. Pseudocode

Client

This program allows the client to connect to a VPN server with a specified IP address and port number and exchange data with encryption through a VPN tunnel. It also allows the client to receive encrypted data from the server and decrypts it to retrieve the data.

```
connect (ip_address, port_number) {
        connects to the server with ip_address and port_number
        if (port unavailable or port used) {
                return False
                exit
        }
        set up a tunnel
        return True
}

send_data (data) {
        encrypt the data
        send the data to the server
}

receive_data () {
        listen to packets coming from the server
        decrypt the data
        return the data
}
```

Server

This program allows the server to listen and establish connections with the client. It receives encrypted data from the client through a VPN tunnel and decrypts it to see the destination. Afterward, it visits and exchanges data with the destination. Finally, it sends back the data with encryption through the VPN tunnel to the client.

```
listen (port) {
    listen to connections on specified port
    if receive connection {
        set up a tunnel
        return True
    }
}

receive_data () {
    listen to packets coming from the client
    decrypt the data
    return the data
}

exchange_data (data, destination) {
    send data to the destination
    return data_from_destination
}

send_data (data, client_ip_address) {
    encrypt the data
    send the data to the client
}
```

8. Innovation

The Open Quantum Safe (OQS) project that aims to create quantum-resistant cryptography was started in late 2016, which is a relatively new concept in cryptography, so its implementation of it should be considered innovative as it is considered a future leading-edge algorithm. My application not only implements cryptography but also integrates the encryption algorithms with the SSL protocol that is used in OpenVPN. The implementation creates an entirely new VPN application that is well-prepared for future attacks from both classical and quantum computers.

One of the similar products with my application is Mullvad VPN. It is a VPN application that also implements post-quantum cryptography. It uses WireGuard as its communication protocol and classic McEliece as its post-quantum encryption algorithms (Mullvad, 2022). Since WireGuard is written in around 4000 to 5000 lines, it does not provide much freedom when it comes to encryption and security compared to OpenVPN, which is written in at least 70,000 lines. If we are to implement a new algorithm in our application, WireGuard would have more restrictions when it comes to development.

9. Complexity

The complexity of this project depends on the newly introduced cryptography and the integration with an existing VPN protocol. Because post-quantum cryptography is relatively new, it will be constantly changing when the developer implements it. Since the liboqs library, which is an open-source library for quantum-safe cryptographic algorithms, only provides algorithms that are accepted by NIST, students will need to implement the entire encryption process manually, including setting up keys and ciphertext and validating keys, by using the CRYSTALS-KYBER algorithm. Diploma students will not be able to solve this problem because they have no knowledge of cryptography and very limited knowledge of VPN protocols. This project requires enough knowledge of networking and application protocols to be able to complete it. Also, for the OpenVPN protocol, students will need to know how SSL (Secure Socket Layer) protocol works, as well as the knowledge of encryption.

10. Technical Challenges

There will be several technical challenges when developing the application. They are listed as follows:

- **OpenVPN Tunneling Protocol**
  OpenVPN tunneling protocol is commonly used for VPN implementation to secure data using AES-256 encryption. I do not have any knowledge of this protocol and will need to find out how can I develop my VPN application with the OpenVPN protocol.

- **Knowledge of Cryptography**
  Since all the networking courses in the program do not cover cryptography, my knowledge of cryptography is very limited. I will have to devote a reasonable amount of time to researching and learning. Simultaneously, I am planning to take the cryptography course starting next semester in January to help.

- **liboqs Library**
  liboqs is an open-source C library for quantum-safe cryptographic algorithms. I will need to have enough knowledge of cryptography mentioned above to understand and make use of this library. Since I am developing my application in Python, I will be using the liboqs-python library for my implementation.

- **Integration of Post-Quantum Cryptography with OpenVPN Protocol**
  After I implemented both the OpenVPN Protocol and understood the liboqs library, I will need to resolve the problem of implementing the CRYSTALS-KYBER algorithm with the SSL encryption that the OpenVPN protocol used.

- **Performance of the Application**
  Since both the CRYSTALS-KYBER algorithm and SSL encryption are very resource intensive, the performance of the application will be worsened as it might cause packet loss and delay. To ease this problem, one possible way is to limit the number of client connections from the server. As a student, I have limited resources to experiment intensive performance of the application.

11. Development Schedule and Milestones

Date	Task	Breakdown	Total
	Project Initialization		
Jan. 5th to Jan. 6th	Setup GitHub Repository	1 hours	5 hours
	Setup New Project (Pycharm)	1 hours	
	Import Libraries (liboqs, OpenVPN)	3 hours	
	Milestone 1: VPN with OpenVPN Protocol		
Jan. 9th to Jan. 27th	Researching basic VPN implementation with OpenVPN Protocol	15 hours	100 hours
	Planning and Gathering Requirements	10 hours	
	Designing architecture of the VPN application	15 hours	
	Implementation	30 hours	
	Writing and Running Tests	20 hours	
	Documentation	10 hours	
	Milestone 2: CRYSTALS-KYBER Algorithm Implementation		
Jan. 30rd to Mar. 3rd	Researching and learning about encryption algorithms	30 hours	175 hours
	Exploring liboqs library	15 hours	
	Exploring CRYSTALS-KYBER Algorithm	15 hours	
	Designing architecture	15 hours	
	Implementation	60 hours	
	Writing and Running Tests	30 hours	
	Documentation	10 hours	

Milestone 3: GUI Integration			
Mar. 6th to Mar. 24th	Exploring PyQt library	10 hours	85 hours
	Designing/Drawing wireframes for both applications (Client and Server)	5 hours	
	Building basic elements for home/main page of the application	10 hours	
	Integrating functions into GUI	30 hours	
	Writing and Running Tests	20 hours	
	Documentation	10 hours	
Milestone 4: Final Documentation			
Mar. 27th to Mar. 31st	Finishing and revising the Final Report	20 hours	30 hours
	Gathering files for deliverables	10 hours	
Total			395 hours

12. Deliverables

- Source Code: VPN Client
- Source Code: VPN Server
- Executable for VPN Client and Server
- Software Manual/User Guide for both the Client and Server software
- Project Final Report

13. Conclusion and Expertise Development

Unlike the class assignments and projects, this project aims to extend on the development of VPN application, which is a perfect project for a BTech student in the Network Security Applications Development Option. It involves implementing VPN with a well-known and well-used protocol, OpenVPN. Also, it requires the understanding of cryptography to implement the algorithms of encryption for post-quantum purpose. It is a great opportunity to gain knowledge of cryptography as it is the last piece for network security.

14. References

Barker, E. B., & Roginsky, A. L. (2015). Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths. https://doi.org/10.6028/nist.sp.800-131ar1

Chen, L., Jordan, S., Liu, Y.-K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). Report on post-quantum cryptography. https://doi.org/10.6028/nist.ir.8105

Computer Security Division, I. T. L. (n.d.). *Post-quantum cryptography: CSRC*. CSRC. Retrieved October 26, 2022, from https://csrc.nist.gov/projects/post-quantum-cryptography

Crystals. Kyber. Retrieved October 26, 2022, from https://pq-crystals.org/kyber/

IBM. What is quantum computing? Retrieved November 7, 2022, from https://www.ibm.com/topics/quantum-computing

Mullvad. Experimental post-quantum safe VPN tunnels - blog. Mullvad VPN. Retrieved November 7, 2022, from https://mullvad.net/en/blog/2022/7/11/experimental-post-quantum-safe-vpn-tunnels/

Open-Quantum-Safe. *Open-quantum-safe/liboqs-python: Python 3 bindings for liboqs*. GitHub. Retrieved October 26, 2022, from https://github.com/open-quantum-safe/liboqs-python

Post-quantum cryptography VPN. Microsoft Research. (2020, September 14). Retrieved October 26, 2022, from https://www.microsoft.com/en-us/research/project/post-quantum-crypto-vpn/

Post-quantum cryptography. Open Quantum Safe. Retrieved September 28, 2022, from https://openquantumsafe.org/post-quantum-crypto.html

15. Change Log

2022.11.20 – Version 2

- **Section 4**: Scope and Depth (Page 4-5)
  - Subsection 4.3 Out of Scope: Added Optimizing Performance
- **Section 7**: System/Software Architecture Diagram (Page 9-11)
  - Rename Figure 1 to System Diagram
  - Added Figure 2(Software Architecture Diagram)
  - Added Subsection 7.1. Pseudocode
- **Section 9**: Complexity (Page 12)
  - Added explanation of the complexity to implement the encryption process with the liboqs library
- **Section 10**: Technical Challenges (Page 13)
  - Added performance of the application

2022.11.06 – Version 1

- Initial Submission