A Novel Multiple Access QKD Network for Secure Communication

Faisal Saleem[1] Demetres D Kouvatsos[2]

Faculty of Engineering and Computer science
University of Bradford
Bradford, UK

Abstract

The quantum key distribution (QKD) system are an alternative to the traditional cryptographic system. These system uses quantum physics to transmit quantum states from one party to another. QKD systems have some success and have different protocols, but until now they are mostly point-to-point protocols, they have a very long way to go. When these systems are mature enough, they will be required to work with current internet infrastructure, which can be costly and will bring so much complexity to the network that it will not be feasible to implement. This research paper proposes a novel multiple access QKD network models for secure communication in a local area network (LAN) environment to handle these problems.

Keywords: Quantum Key Distribution, Communication Protocols, Security, Quantum Physics, Local Area Network (LAN)

1 Introduction

Quantum cryptography is a new and vital occurrence in the history of cryptography. It is a procedure that can be used to safeguard the privacy of data communicated among two parties, by exploiting the counter-intuitive conduct of essential elements such as photons. The laws of quantum mechanics govern the physics of elementary items. At nuclear scales, primary components do not have an exact position or speed, as we would instinctively expect. Particles will lose information on its speed and location by observation as apprehended by the famous Heisenberg uncertainty principle. This principle is not a restriction due to the observer's technology but instead an essential limitation that no one can ever overcome. In the late sixties, Wiesner [1] presented the first use of quantum information theory. He proposed the first application of unforgeable banknotes were the first application of the spin of particles. The observer cannot get single-particle spin information without destroying its other parts. By effectively encoding identification information on banknotes, the bank can check the consistency of particles to verify the authenticity of notes. The quantum information stored in subatomic particles is not forgeable. Copying the identification information of banknotes is subject to the uncertainty principle. Bennett and Brassard proposed a key distribution protocol following the track of Weisner's [1] idea of quantum mechanics called QKD [2,3]. They developed a key exchange process to exchange quantum keys.

1.1 Quantum Key Distribution (QKD)

QKD is a new idea which practices the laws of quantum physics to dispense the encryption key between two parties. It uses two separate channels for communications, the public channel for routine communications and quantum channel for distributing encryption key as quantum states explained in [Figure 1]. QKD uses photons. Each photon encodes one binary value into its polarisation [2,3,4,5]. An ordinary laser can produce a single photon. If Eavesdropper tries to measure the subatomic properties of a moving photon, it disturbs the whole system. This attempt causes an increase in QBER.

[1] Email:F.Saleem4@student.bradford.ac.uk
[2] Email:D.Kouvatsos@bradford.ac.uk

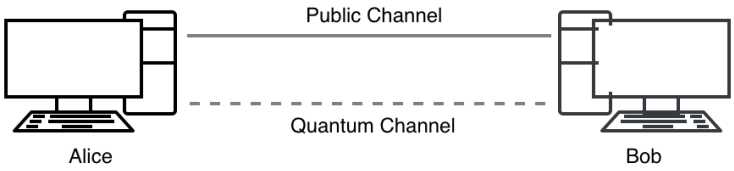

Fig. 1. QKD Communication Channels [2]

1.2 Qubit and Qubit Pairs

Traditional computing uses binary to process information where data at its lowest level is in the form of binary (Zeros 0 and Ones 1), which corresponds to ON and OFF. If a value is 0, that means switch/circuit is OFF, and if it is 1, then the switch/circuit is ON. In quantum computing, these bits encode in particles with different states called Qubits with a new position called supper position where a qubit can be zero or one at the same time. When these qubits are entangled together in a way where the quantum state of one particle cannot explain independently. If the spin of one particle from two entangled particles is clockwise, the other particle spin counter-clockwise [6].

1.3 QKD Work Process

The BB84 was the first QKD protocol. In this protocol, two parties, "Alice" and "Bob," secretly exchange keys by using laws of quantum physics. Alice prepares qubits by encoding binary value to the polarisation of photons. Polarisation or spin of photon represents the superposition [7]. Table 1 illustrates four polarisation states used in the BB84 protocol. Alice selects one of two conjugate bases to determine the spin direction of each qubit. After encoding, Alice transmits these qubits to bob over a quantum channel. Bob, on the receiving end, measures the photons with a rectilinear or diagonal basis randomly [7,8]. If an eavesdropper attempts to measure the quantum states of these qubits, this attempt disturbs the whole system increasing the Quantum Bit Error Rate (QBER)[9,10].

Bit Value	Basis	Polarization State	Spin Direction	Quantum Notation
1	Rectilinear	Horizontal	↕	$\|1\rangle$ or $\|\updownarrow\rangle$
0	Rectilinear	Vertical	↔	$\|0\rangle$ or $\|\leftrightarrow\rangle$
0	Diagonal	Anti-Diagonal	↙↗	$\|0\rangle$ or $\|\swarrow\nearrow\rangle$
1	Diagonal	Diagonal	↖↘	$\|1\rangle$ or $\|\nwarrow\searrow\rangle$

Table 1
Photons Polarization States

2 Multi Access QKD LAN Network Model

QKD requires a segregated quantum channel to transmit quantum states and public channels to send encrypted data and other protocol-related data from one party to another. Several protocols developed on the same principles, but our proposed model uses the first QKD protocol BB84. The proposed model has a QKD enabled network switch consists of public and quantum channels bound together physically and logically. Figure 2 shows a QKD enabled local area network with a custom-designed network switch. All connected terminals are using custom-designed network adapters, which supports QKD and can be linked directly to the network switch (see Figure 2).

2.1 Switch

The proposed multiaccess QKD network model uses a customised network switch with a custom-designed operating system to support multiaccess QKD across a local area network. The proposed switch help reduce the cost and complexity and give quantum physics-based key exchange process among all network nodes. Based on QKD requirements of having two different channels of public and quantum channels to be successful, this proposed network switch has two different kinds of connections bound physically and logically together to support multipoint access. Figure 4.2 shows the front end of switch model with 4 Fast Ethernet connections (Fe0, Fe1, Fe2, and Fe3) for public communication and four quantum channels (Ψ0, Ψ1, Ψ2, Ψ3) for transmission of quantum states from one terminal to another access the network. The proposed switch also maintains

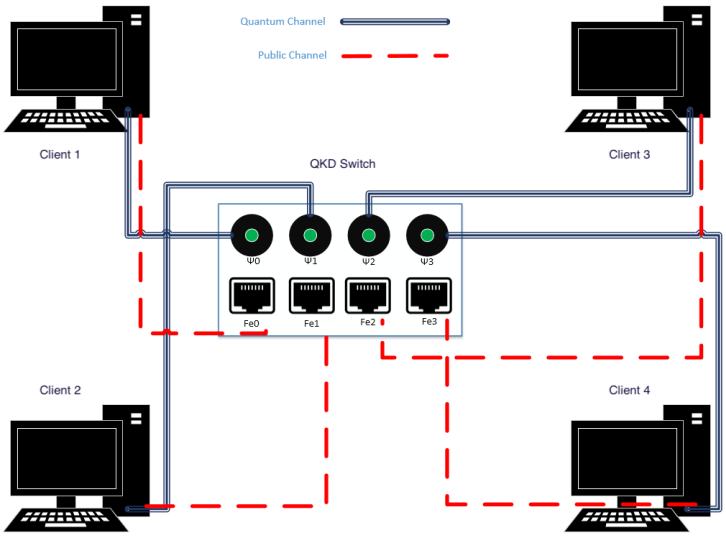

Fig. 2. Multi Access QKD LAN Network Model

Interface	MAC Address	Ψ Interface	Ψ MAC Address	Type
Fe0	AABBCCDDEE01	Ψ0	QQQQQQQQQQ01	Static
Fe1	AABBCCDDEE02	Ψ1	QQQQQQQQQQ02	Static
Fe2	AABBCCDDEE03	Ψ2	QQQQQQQQQQ03	Static
Fe3	AABBCCDDEE04	Ψ3	QQQQQQQQQQ04	Static

Table 2
MAC Address Table

an extended logical MAC Address table in its memory based on the physical binding of these interfaces (see Table 2). Each Fast Ethernet connection is bound physically with its corresponding quantum interface, which reduces the complexity of having many separate direct quantum channel connections with other nodes in the network. Fast Ethernet 0 physically connects to Ψ0, Fast Ethernet 1 to Ψ1. During the boot process, the proposed network switch checks the physical binding of these interfaces. It builds the MAC address table with the actual MAC address of these interfaces and mark this as Static in type section to show that this entry has been added statically by the switch and is not learned dynamically during the learning process.

2.2 Architectural Design of Switch

The proposed switch model for multiaccess QKD simulated in OMNeT++, which is a discrete event network simulation framework. Our model includes Fast Ethernet interfaces bound with the quantum interface defined in a quantum binding table maintained by switch operating systems during the boot process in switch memory. The proposed switch has simple modules like fast ethernet interfaces, quantum interface, quantum states processor, CPU. all these components have been joined together to form a compound module called Switch in our simulation model. Simple modules are communicating with each other by passing messages through in and out gates. Clients communicate with each other over public channels (Fast Ethernet) for regular data transfer. For transferring quantum states from one client to another, the quantum interface uses subinterfaces (see Figure 3), where Q01, Q02, Q03, and Q04 are quantumly interfaced with sub interfaced connected to each of its correspondent quantum sub interface.

3 Client

A proposed network model for multiple access QKD is using a custom design architecture. This architecture is using the modules different than traditional network clients to support QKD. Client architecture has CPU, memory, and network interface like a conventional computer with some extra components which support the transmission of quantum states from one client to another and also uses components to receive the quantum states on the receiver side. These components are quantum states generator

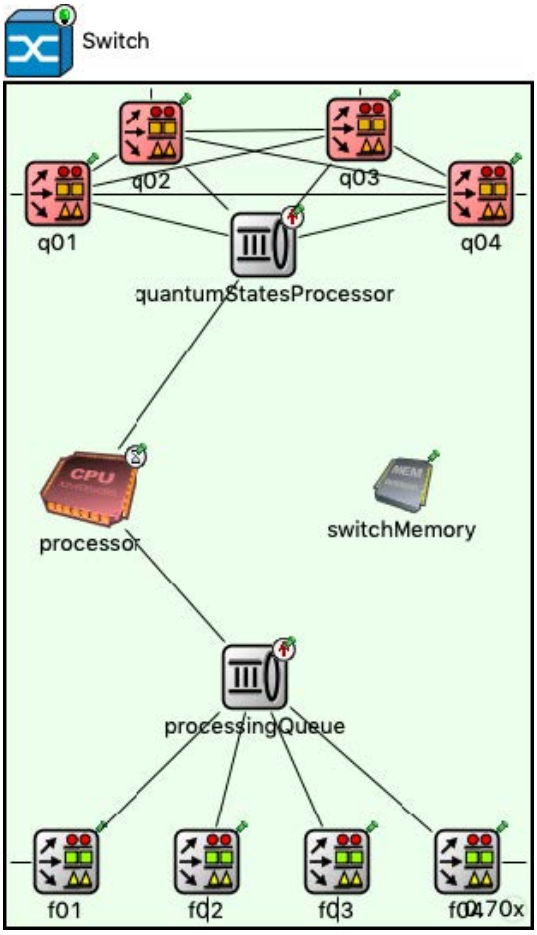

Fig. 3. Switch Internal Design

- Quantum States Generator
- Sender Side Polarisation Filters
  - · Diagonal Left polarisation filter
  - · Diagonal Right polarisation filter
  - · Horizontal Polarisation filter
  - · Vertical polarisation Filter
- Receiver Side Polarization Filters
  - · Diagonal Polarization Filters
  - · Straight Polarization Filters
- Quantum Interface to send or receive quantum states over a quantum channel

The client process generates a random stream of binary zeros and ones (0, 1) and sends them to quantum states generator. This module encodes each binary digit into photons polarisation and transmits them over a quantum channel (see Table 1). On the receiver side, these qubits (binary encoded photons) randomly passed through 2 (diagonal, straight) polarisation filters. These filters remove the encoding and send them to the quantum states generator to convert them back to original binary values. During this process, if the polarisation of photons has been affected by any external force, then the binary value will be completely different from the original, which cause an increase in QBER. With a higher QBER, the receiver ignores the received key. The sender receives the response and retries after a random period.

4 Network

This section merge both of the custom-designed elements (Network Switch and Client). A network consists of 4 clients connected to a QKD supported network switch is simulated with our selected simulation tool OMNet++. This section explains the connection between all components and the actual process of message

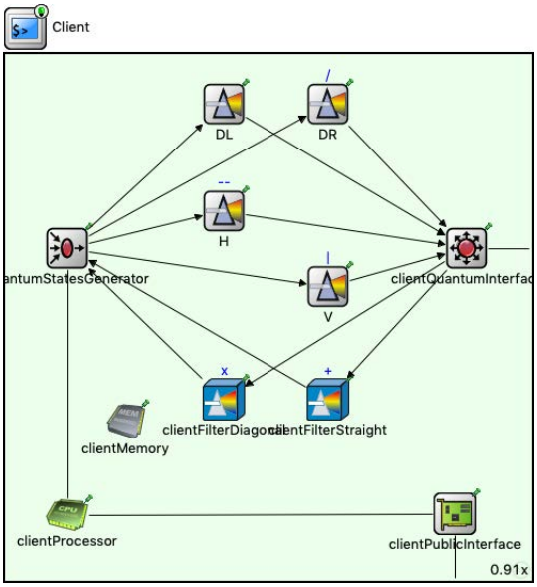

Fig. 4. Client Internal Design

passing among these components.

4.1 Network Model and its components connectivity

The network is a system or a group of connected components or devices, where devices or components exchange information between each other or use each other resources to reduce the cost and increase productivity. The proposed network model in this research is using QKD mechanisms (which is using quantum physics laws) to ensure the safety of the key exchange process. Figure 5 illustrates a network diagram of 4 clients (client01, client02, client03, and client04) connected directly to a centralised point, which is a QKD support network switch. All clients connected to a centralised switch via two separate channels. The channel showing as the black line is the public channel which is used by all clients and switch to transmit the standard communication data. The second channel with red dots is simulating the quantum channel, which is used by these connected components to send quantum states from one to another.

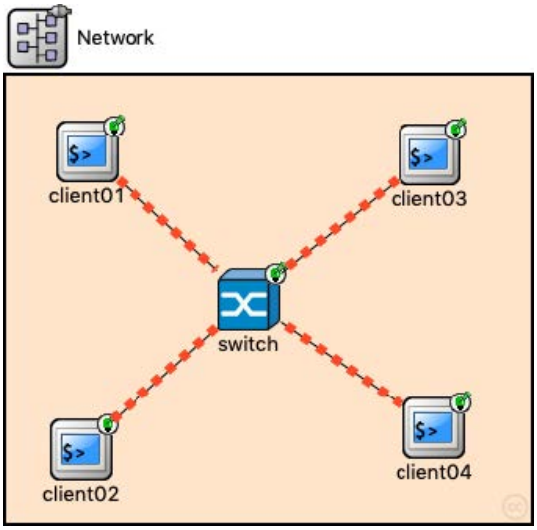

Fig. 5. Network Design

4.2 Clients Module behaviour

Each module has its capability, and for the successful implementation of QKD, all these modules have to work together. Traditional components are responsible for all communication based on standards set by ISO, which include the OSI reference model. The parts related to the quantum module are responsible for encoding binary values received from traditional elements, encode them into quantum states or spins of photons, transmit them over the quantum channel to the second party. On receiving side quantum modules, analyse these photons to determine the existence of eve by measuring QBER (Quantum Bit Error Rate), decoding these received encoded qubits to binary values after key sifting and sending them to the public module. Then the public module sends the filters used for these measurements to original send for comparison so both parties can have the same key.

4.3 Switch Modules Behaviour

Switch is the central point of the network where it receives the communication from all directions on its public interface and makes the decision based on the layer two address of a packet. For broadcast it receives the broadcast from a specific interface on MAC address FF: FF: FF: FF: FF: FF and send out to all interface except the one it receives. In general, switch makes its forwarding decisions based on layer two addresses. In our proposed, we used the same layer two address to make forwarding decisions. Our proposed switch has two different kinds of modules bound physically and logical together for the successful implementation of QKD. The proposed switch maintain the connectivity status of each quantum interface. A QKD request initiated by a client to let switch know he wants to transfer quantum states to a specific client. On receiving the request, the switch checks its internal QKD session table if another client is not in the process of sending or receiving quantum states. If there is another entry in the session table, swith terminate the session. Initiating client will try again after a random amount of time. If there is no existing session, the switch request the second client and later allow both of these clients to transmit quantum states.

5 Process

Two clients (client02 and client04) need to communicate quantum keys among each other by using the proposed QKD process. First, they agree on the basis they going to use for each bit value. [Figure 1] presents the table of these polarisation states. Client02 sends out an initial request to switch over the public channel with a destination mac address of client04. QKD Switch receives the request and checks its QKD session state table for any existing session for both clients. The switch makes this decision based on the quantum mac addresses. If client04 is already in the process of QKD exchange, the switch denies the request and respond with a specific header. Client02 receive the message and terminate the session and wait for a particular amount of time to retry if Client04 is not in the process of key exchange. The switch makes an entry in its session state table with status "Request" and sends the request to Client04. Client 04 receives the request from the switch and acknowledge the request. When the switch receives the acknowledgement, it updates the session state table. It opens the subinterface 4 of quantum interface two and subinterface 2 of quantum interface four. Moreover, it closes all the other sub-interfaces on these quantum channels.

$$\text{Client02} \leftrightarrow \text{Sub Interface 04} \leftrightarrow \text{Sub Interface 02} \leftrightarrow \text{Client04}$$

After this, the QKD switch sends the acknowledgement to the client02 to start the transmission. Client02, on receipt of an acknowledgement, it generates a stream of random bits and sends them over to the quantum states generator module for encoding these binary digits into the polarisation or spin of light particles (photons). Quantum states generator randomly selects the filters based on agreed polarisation states mentioned in (see Table 1). Each binary digit gets encoded into the polarisation of photons and transmitted over the quantum channel, which connects to the QKD supported network switch. During the session establishment process, switch already closed all the subinterfaces of quantum interface 02 and quantum interface 04 except connected to these two clients. So, on arrival of these quantum states, they automatically received on Q02 and send over the sub-interface Q04 of interface Q02, which then received by Quantum Interface 04 on sub interface q02 and exit through Quantum interface 04. Client 04 receives them on its quantum interface. Client04 receives the quantum states on its quantum interface and randomly pass every single photon from one of its straight or diagonal polarization filters. The quantum state's generator module decodes them. After decoding, it performs the key sifting process and discards the non-matching photons and hand over the decoded binary values with the filter usage to the client processor for further processing. The client processor sends the filter usage to client02 over public channels. When the switch receives these packets filter usage, it removes the session entry from its session state table, which allows other clients to communicate. Client02 receives the filter usage from client04 and compares with its usage and discards the non-matching binary digits, while leaves the final key (see Figure 6). This generates the same key as client04 has after its key sifting process. Hence both clients have the same encryption key made with complete randomness and distribution with laws of quantum physics.

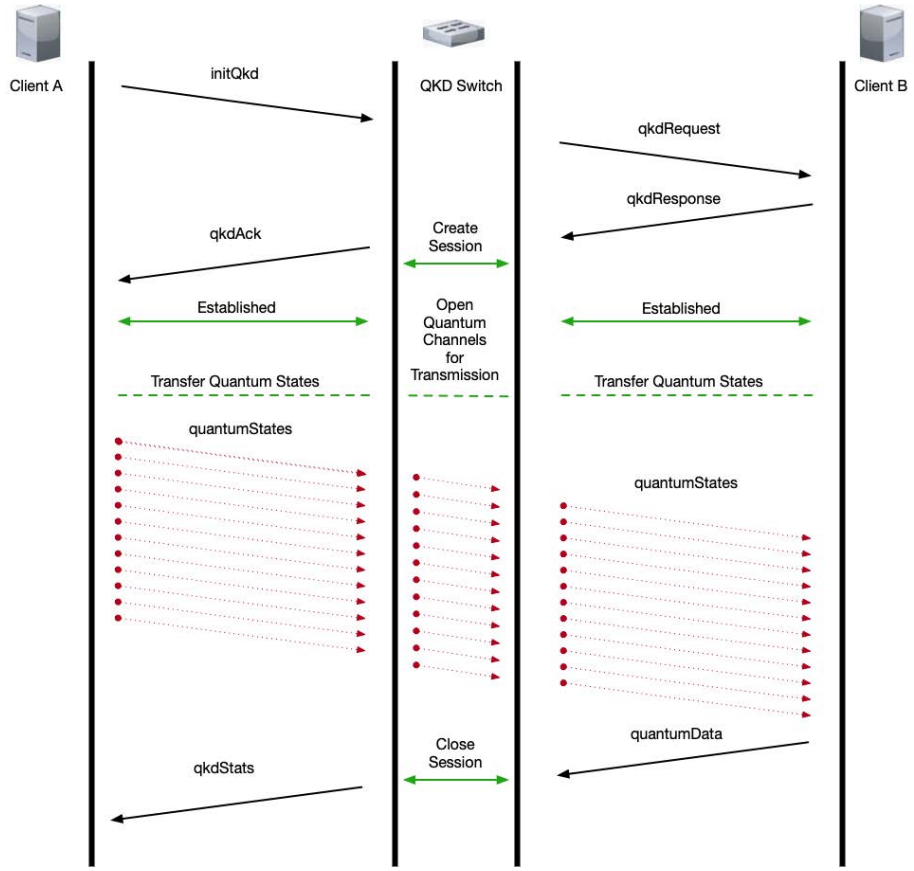

Fig. 6. Multi Access QKD Process

6 Results

6.1 2 Clients Exchanging Quantum Key

In this scenario, only two clients exchange quantum keys among each other, and no other client made any request to transmit the quantum key. E.g., client02 exchange quantum keys with client04.

Figure 7 exhibits that client 02 generates a stream of random bits and transmitted to the quantumStates-

```
[*] client02 RANDOM BITS & PHOTON POLARIZATION
*********************************************************************************************
RANDOM BITS               : 0010000011111110101001001001101010111011011011010011111001000
POLARIZATION FILTERS USED : \\/\\\|\-///--/|-\/\\/\|/\|/-|/\/|---|//\-/|/--\/\|---///|\-||\
*********************************************************************************************
```

Fig. 7. Random Bits Generation & Photon Polarization (OMNet++ Log)

```
[*] client04 PHOTONS RECEIVED & PROCESSED
*********************************************************************************************
POLARIZATION FILTERS USED    : ++x+xx++xxx+++x+++x+xx+xx+xx+xx+x+xxx+x++x+x+++xxxxx++++xx++x++
POLARIZATION RECEIVED        :   / \\|  // --/|- / \/  /  /- / /|   |/      --\/\  --  /  - |
SIFTED KEY                   :   1 000  11 11101 1 01  1  11 1 10   01      11010  11  1  1 0
---------------------------------------------------------------------------------------------
FINAL KEY                    : 1000111110110111111001101011110
*********************************************************************************************
```

Fig. 8. Quantum Key Processing at Receiving End (OMNet++ Log)

```
[*] client04 Quantum Key Table
============================================================================
  ID       MacAddress                        Key                    Status
============================================================================
  1     AA:AA:AA:AA:AA:02     10001111101101111110011101011110     Active
============================================================================

[*] client02 Quantum Key Table
============================================================================
  ID       MacAddress                        Key                    Status
============================================================================
  1     AA:AA:AA:AA:AA:04     10001111101101111110011101011110     Active
============================================================================
```

Fig. 9. Final Key (OMNet++ Log)

Generator module. quantumStatesGenerator module received the bits and randomly encoded them in into photon polarization where (0 is encoded into \and —) (1 is encoded into / and -). These quantum states than transmitted over a quantum channel to Client04. Figure 8 exhibits the processing of quantum states received at Client04. Each polarised photon (qubit) randomly passed through one of two polarisation filters on Client04. It also shows the processing of quantum bits when than decoded back to binary digits. After key sifting processing, a random key generated by Client04. The details of which filter used for each photon on arrival shared with Client02 over a public channel. Client02 compared these filters with its filter usage and disregarded the non-matching bits after comparing filter usage. Hence both parties have the same key generated completely random by using QKD mechanism. Figure 9 illustrates the quantum key table on both clients after completion of the QKD process.

6.2 All four clients exchange quantum keys in pairs

In this scenario, two pairs of 2 clients exchange quantum keys with each other without requesting a client already in the key exchange process.

(Client02 ↔ Client 04) and (Client01 ↔ Client03)

```
[*] client01 RANDOM BITS & PHOTON POLARIZATION
*********************************************************************************************
RANDOM BITS               : 1000001111111010100100100110101011101101101110100111111100100000
POLARIZATION FILTERS USED : /\|\\\-------|-|-\|/\\/\\|///|/|/|/-/|--\/-\///|/|\/-/--/\\/||\|\
*********************************************************************************************

[*] client02 RANDOM BITS & PHOTON POLARIZATION
*********************************************************************************************
RANDOM BITS               : 0000101000110110000001001011000111110001010110001111000101110 10
POLARIZATION FILTERS USED : |||\-|/|\|-/|-/\|\\\|-\|-|--|\|////\\|-\-|//\\\-/-/||\/|///|/\
*********************************************************************************************
```

Fig. 10. Random Bits Generation & Photon Polarization (OMNet++ Log)

6.3

In this scenario, two multiple clients exchange the quantum keys simultaneously.

Client02 ↔ Client04 Client03 ↔Client01

Both clients (Client02 and Client04) generated the complete random stream of bits and transferred over to the quantumStatesGenerator module for quantum encoding. quantumStatesGenerator module randomly encoded the bits into photon polarisation and transmitted them over the quantum channel to quantum switch. Figure 10 illustrates this because switch maintains the session state of quantum interfaces, so quantum switch allowed the transfer of quantum states without any issue. Figure 11 and Figure 12 present the processing of quantum states on the receiver side and getting the final key after key sifting processing and exchange of filter usage information, respectively.

```
[*] client03 PHOTONS RECEIVED & PROCESSED
*********************************************************************************************
POLARIZATION FILTERS USED    : x+xxxxx+xx+x+++xxx++++xxx++++x++x+xxxx+x++xxxx+xx++x++xxx++++xx
POLARIZATION RECEIVED        : /  \\\ -  - -|-  \|   /\   |   |/-/    /- ///   \ -/--/\\ ||  \
SIFTED KEY                   : 1  000 1  1 101  00   10   0   0111    11 111   0 1111100 00  0
---------------------------------------------------------------------------------------------
FINAL KEY                    : 10001110100100011111111101111100000
*********************************************************************************************

[*] client04 PHOTONS RECEIVED & PROCESSED
*********************************************************************************************
POLARIZATION FILTERS USED    : +xxx++x+xx+x++xx+xx+++xxx+++x+xxxxx++xx+++x+++xxxxxxx+x+++x++x+
POLARIZATION RECEIVED        : |  \-|/|\ -/|-/\|\\ |-\  |--   ////  \ - -    \\ / / |\ | / |/
SIFTED KEY                   : 0  010100 110110000 010  011   1111  0 1 1    00 1 1 00 0 1 01
---------------------------------------------------------------------------------------------
FINAL KEY                    : 00101001101100000100111111101100110000101
*********************************************************************************************
```

Fig. 11. Quantum Key Processing at Receiving End (OMNet++ Log)

```
[*] client03 Quantum Key Table
==========================================================================================
  ID       MacAddress                        Key                                Status
==========================================================================================
  1     AA:AA:AA:AA:AA:01     10001110100100011111111101111100000          Active
==========================================================================================

[*] client04 Quantum Key Table
==========================================================================================
  ID       MacAddress                        Key                                Status
==========================================================================================
  1     AA:AA:AA:AA:AA:02     00101001101100000100111111101100110000101     Active
==========================================================================================

[*] client01 Quantum Key Table
==========================================================================================
  ID       MacAddress                        Key                                Status
==========================================================================================
  1     AA:AA:AA:AA:AA:03     10001110100100011111111101111100000          Active
==========================================================================================

[*] client02 Quantum Key Table
==========================================================================================
  ID       MacAddress                        Key                                Status
==========================================================================================
  1     AA:AA:AA:AA:AA:04     00101001101100000100111111101100110000101     Active
==========================================================================================
```

Fig. 12. Final Key (OMNet++ Log)

7 Use of OMNet++ framework

OMNeT++ is a discrete event network simulation library and framework which is being used to simulate various problem areas which include [7]

- Network Protocols
- Queueing Networks
- Distribution Systems
- Microprocessors
- Performance Analysis of complex networks and software systems
- Communication Networks.

It is a significant C++ oriented modular, component-based, very extensible library and framework. OMNeT++ is used to simulate our proposed network that will support QKD with extended network protocols to accomplish multi access communication among different network nodes. OMNeT++ provides an architecture where compound and straightforward modules work together to build the network and pass messages among

them. [7].

OMNet++ provides the support to develop individual components. These components are called simple modules. Simple modules can be used to create more complex called compound modules and finally joining all the simple and compound modules together to build and simulate a network [7].

8 Related Workd

Chip Elliott [11] proposed the implementation of QKD protocols in real networks with internet architecture. He proposed three different architectures for QKD implementation. In the first architecture, he presented a QKD mechanism among two private enclaves. These two enclaves have various communication channels (internet/public for the transmission of encrypted traffic). Where all computer systems of one enclave connected to one central point, which connects to the QKD endpoint to which send and receive quantum states to the second enclave. The second system architecture of a trusted network to overcome the drawback of geographic distance by using trusted relays that work as a mesh of QKD nodes. QKD endpoints are connected to these trusted relays to transmit quantum keys remote enclaves, which are geographically not very close. The prime weakness of this using trusted relays is that if a trusted relay compromises, it will affect the whole system. The third system architecture bases were an untrusted network that Elliott [11] discussed. In this approach, he proposed the use of an untrusted QKD switch, which is used to set up the optical paths to transmit photons from one QKD endpoint to another endpoint without participating in the QKD protocol. This approach also has the weaknesses that these untrusted QKD network switches cannot spread the topographical reach of a QKD network.

9 Future Work

Future work will extend the QKD model in a WAN environment, where a client wants to exchange quantum keys with another client residing in another network. The other network can be a VLAN (virtual local area network) or a server or service available over the internet where packets have to leave the local network and exit out of network gateway. In this context, further empirical research is needed to explore this area in greater detail. The QKD model needs to be extended, as appropriate in order to be applicable in a typical LAN environment when a network switch consists of 24 and 48 port access switches. This is due to inability of this model cannot be applied if there is a considerable number of clients and more than one network switch required. This limitation also needs additional analysis and research to exchange quantum states among different network switches or routers. As an immediate next step of this work, these suggestions might be incorporated into future prototypes of the proposed QKD model.

10 Conclusion

Quantum cryptography is a new and vital occurrence in the history of cryptography. It is a procedure that can be used to safeguard the privacy of data communicated among two parties, by exploiting the counterintuitive conduct of essential elements such as photons. A comprehensive exploration of the QKD protocols was carried out and proposed a QKD model for LAN environment. This proposed model reduces the cost and complexity of the network when the number of clients in the network increases (c.f., Section 4.1). The proposed QKD model in this research is also based on clients using a single application, where the operating system handles the key exchange process over the quantum channel. This proposed model also has the restriction of multiple applications. For example, this QKD model doesn't support the scenario where if clients have various applications needing to exchange quantum keys with the same application of another client; thus, this area of research needs an in-depth exploration. This QKD network model will not work in WAN environment as well because of the laws of quantum physics and other technological difficulties and need more exploration and experiments.

References

[1] S. Wiesner, "Conjugate coding," *ACM SIGACT News*, vol. 15, pp. 78–88, jan 1983.

[2] C. H. Bennett, "Quantum cryptography using any two nonorthogonal states," in *Physical Review Letters*, pp. 3121–3124, APS, 1992.

[3] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theoretical Computer Science*, vol. 560, no. P1, pp. 7–11, 2014.

[4] D. Bruss, G. Erdélyi, T. Meyer, T. Riege, and J. Rothe, "Quantum Cryptography: A Survey," in *ACM Comput. Surv.*, vol. 39, (New York, NY, USA), ACM, jul 2007.

[5] A. A. Berezin, "Quantum computing and security of information systems," in *WIT Transactions on the Built Environment*, vol. 94, pp. 149–159, 2007.

[6] S. Wang, W. Chen, Z.-Q. Yin, H.-W. Li, D.-Y. He, Y.-H. Li, Z. Zhou, X.-T. Song, F.-Y. Li, D. Wang, H. Chen, Y.-G. Han, J.-Z. Huang, J.-F. Guo, P.-L. Hao, M. Li, C.-M. Zhang, D. Liu, W.-Y. Liang, C.-H. Miao, P. Wu, G.-C. Guo, and Z.-F. Han, "Field and long-term demonstration of a wide area quantum key distribution network," *Optics Express*, vol. 22, no. 18, p. 21739, 2014.

[7] L. O. Mailloux, J. D. Morris, M. R. Grimaila, D. D. Hodson, D. R. Jacques, J. M. Colombi, C. V. McLaughlin, and J. A. Holes, "A Modeling Framework for Studying Quantum Key Distribution System Implementation Nonidealities," in *IEEE Access*, vol. 3, pp. 110–130, IEEE, 2015.

[8] S. Loepp and W. K. Wootters, "Protecting information: From classical error correction to quantum cryptography," in *Protecting Information: From Classical Error Correction to Quantum Cryptography*, vol. 9780521827, pp. 1–287, Cambridge University Press, 2006.

[9] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," in *Reviews of Modern Physics*, pp. 1301–1350, APS, 2009.

[10] D. R. Hjelme, L. Lydersen, and V. Makarov, "Quantum cryptography," in *A Multidisciplinary Introduction to Information Security*, vol. 74, pp. 73–92, APS, 2011.

[11] C. Elliott, "Building the quantum network," *New Journal of Physics*, vol. 4, p. 46, jul 2002.