Quantum key-based secure and authenticated message transmission in wireless networks using quantum pareto optimal routing

Mrs.P.Pavani Sri Katyayini
Assistant Professor, Department of Computer Science and Engineering. Seshadri Rao Gudlavalleru Engineering College
Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru
Gudlavalleru, India
Katyayini33979@gmail.com

Manikanta Rajulapati
Student, Department of Computer Science and Engineering. Seshadri Rao Gudlavalleru Engineering College
Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru
Gudlavalleru, India
19481a05k5@gmail.com

Rajyalakshmi Yarlagadda
Student, Department of Computer Science and Engineering. Seshadri Rao Gudlavalleru Engineering College
Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru
Gudlavalleru, India
rajiyarlagadda2002@gmail.com

Triveni Yandrathi
Student, Department of Computer Science and Engineering. Seshadri Rao Gudlavalleru Engineering College
Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru
Gudlavalleru, India
triveni.y96@gmail.com

Siva Naga Raju Tirumalasetti
Student, Department of Computer Science and Engineering. Seshadri Rao Gudlavalleru Engineering College
Seshadri Rao Gudlavalleru Engineering College, Gudlavalleru
Gudlavalleru, India
tsnraju2002@gmail.com

Abstract— Securing communication networks is a big problem in the modern world because information needs to be sent from one place to another without being hacked. The system's secrecy relies on the shared key algorithm and public key. Quantum cryptography is a technique that provides a more secure communication network based on photon polarisation and the principle of uncertainty. In order to guarantee secure communication, diverse encryption techniques are employed to withstand different forms of attack. An effective method has been proposed to enhance message authentication by combining the Blowfish algorithm (used for encryption) with a new quantum hash function (used for message authentication).Wireless multi-hop networks (WMHNs) are used for secure communication. These networks require a trade-off between different Quality of service requirements (QOS).By means of the Pareto optimality criterion, the resulting solutions are incorporated into the Pareto frontier .Nevertheless, locating all the pareto-optimum paths in WMHNs can be difficult, and the suggested approach entails utilising a dynamic optimization i.e. Evolutionary Quantum Pareto Optimization (EQPO) algorithm..

Keywords: Quantum Key Distribution(QKD), Quantum Hash Function (QHF), Quantum Encryption (QE),Quantum Crptography,Wireless Multihop Network(WMHN),Quantum Walk,BlowFish Algorithm,Evolutionary Quantum Pareto Optimization(EQPO).

I. INTRODUCTION

A. Quantum Cryptography

Cryptography has become an essential aspect of modern-day communication systems. It involves the use of mathematical algorithms and protocols to secure the exchange of information between parties. With the rapid advancement of technology, traditional cryptographic techniques such as symmetric-key cryptography and public-key cryptography are becoming increasingly vulnerable to attacks from quantum computers. Therefore, there is a need for a more secure cryptographic technique that can withstand the processing power of quantum computers, and this is where quantum cryptography comes into play.

Quantum cryptography is a field of study that utilizes the principles of quantum mechanics to develop secure communication protocols. Unlike traditional cryptography, which relies on mathematical algorithms, quantum cryptography utilizes the unique properties of quantum mechanics to ensure secure communication between parties. In particular, it uses the phenomenon of quantum entanglement and the Heisenberg uncertainty principle to establish secure communication channels.

The basic principle of quantum cryptography is that any attempt to observe or measure a quantum system will inevitably disturb it. This is known as the Heisenberg uncertainty principle, which asserts that the more precisely one knows the position of a particle, the less precisely they can know its momentum, and vice versa. In the context of quantum cryptography, this means that any attempt to eavesdrop on a quantum communication channel will inevitably disturb the quantum state of the particles being transmitted, thus alerting the communicating parties to the presence of an intruder.

Quantum entanglement is another fundamental principle of quantum cryptography. When particles are entangled, their states become interdependent, regardless of the distance between them. This means that any modification made to one particle will instantly affect the other particle, irrespective of the distance between them. By leveraging this property, quantum cryptography enables the establishment of secure communication channels between two parties.

Quantum cryptography has several advantages over traditional cryptographic techniques. Firstly, it provides unconditional security, which means that it is mathematically impossible for an eavesdropper to intercept or modify a message without being detected. Secondly, it is not vulnerable to attacks from quantum computers, which have the potential to break traditional cryptographic algorithms. Thirdly, it offers a higher level of privacy, as any attempt to eavesdrop on a quantum communication channel will inevitably disturb the system, alerting the communicating parties to the presence of an intruder.

There are several applications of quantum cryptography in real-world scenarios. For example, it can be used to secure communication between two parties, such as financial transactions, military communication, and government communication. It can also be used to secure communication between a user and a server, such as in online banking or e-commerce transactions. In addition, it can be used to secure communication in a network, such as in a corporate environment or a government agency.

In conclusion, quantum cryptography offers a promising approach to secure communication in an era where traditional cryptographic techniques are becoming increasingly vulnerable to attacks from quantum computers. By utilizing the unique properties of quantum mechanics, it provides unconditional security, immunity to attacks from quantum computers, and a higher level of privacy. With its potential applications in various real-world scenarios, quantum cryptography is poised to revolutionize the way we communicate and exchange information securely.

B. Routing Algorithm

The phenomenon of Wireless Multi-Hop Networks (WMHN) makes it possible for nodes located far apart to connect with one another by forwarding data packets via a collection of mobile relays. Since the selection of relays has such an important role in the performance of WMHN, optimizing the routing is an important factor to take into account. In order to achieve optimal routing, it is necessary to strike a delicate balance between various Quality-of-Service (QoS) requirements, which can often be in direct opposition to one another. These QoS requirements can include things like the overall Bit-Error-Ratio (BER), power consumption, end-to-end latency, achievable rate, system sum-rate, and lifetime.

Many studies have been carried out with the goal of optimizing the routing of wireless multi-hop networks (WMHN) by taking into account various quality-of-service (QoS) requirements. Many studies use objective functions (OF) with a single component, such as Network Lifetime (NL) and Network Utility (NU), to accomplish this goal. Although NL takes into consideration the power consumption of the routes and the battery levels of the nodes, NU also takes into account the maximum rate that the routes are capable of achieving. In some research, aggregate functions are constructed and quality of service is included as a constraint in order to offer a more all-encompassing perspective on the routing issue. For example, in order to improve network throughput, an optimization was conducted on an aggregate function that takes into account the cost of Dirichlet routing and the average network latency under specified operational conditions.

In order to discover the most efficient routes while still depending on single-component aggregate functions, dynamic programming has been used. One reason why Dijkstra's algorithm has been used is that it can come close to finding the best routes. On the other hand, it adds a level of complexity to $O(E^3)$, where E represents the total number of edges in the graph that represent the network. Also, the Viterbi decoding method has been changed so that it works better with the challenges of single-component routing optimization. In a manner similar to the Bellman-Ford method, this method transforms the routing problem into a decoding problem by modelling the route discovery procedure as a trellis graph.

Quantum computing gives us a way to solve Pareto-optimal routing problems using quantum parallelism (QP). Quantum annealing is one method that makes use of this to maximize the throughput of a wireless network with the least amount of interference possible. Quantum annealing simplifies things significantly when compared to traditional approaches like simulated annealing. With a complexity of $O(N \surd N)$, the NDQO algorithm has successfully identified the whole collection of pareto-optimal paths using QP for pareto-optimal routing utilizing quantum computing. Taking advantage of the synergistic relationship between QP and hardware parallelism, a new method dubbed "non-dominated quantum iterative optimization" may find a whole set of pareto-optimal paths with a parallel complexity of $O(NOPF \sqrt{N})$ and a sequential complexity of $O(N2OPF \sqrt{N})$. It's worth noting that the number of pareto-optimal paths is represented by NOPF.

The Non-Dominated Quantum Optimization and Non-Dominated Quantum Iterative Optimization algorithms for multi-objective problems are still impractical for networks with a large number of nodes despite the simplification they offer. Assuming that the values in the database have nothing to do with each other, Zalka's proof suggests that the complexity is at least O(N). But social networking can make it possible to link Pareto-optimal route combinations, which can make things even simpler. Based on this idea, we propose a new method called Evolutionary Quantum Pareto Optimization. EQPO creates trellis graphs that direct the search process in a manner akin to Viterbi decoding by using the commonalities between several Pareto-optimal pathways. We also use a combination of QP and HP to minimize the number of database entries while maintaining a near-full search-based speed, resulting in additional complexity reduction.

In contrast to the sequential complexity, which ignores the possibility of parallelism altogether, the parallel complexity is defined as the degree of difficulty that takes into consideration the amount of parallelism involved.

C. Quantum Key Distribution

Two parties, typically called sender and receiver, can talk to each other in a completely secure way. For secure communication, you need to be able to share a key, which is a set of random bits that have already been chosen. Classical cryptography, on the other hand, lets people listen in on the key distribution channel without the real users noticing. This is not the case with Quantum Key Distribution (QKD), which offers a solution to this issue and is based on quantum mechanics. It has been suggested as a new way to solve this problem in quantum cryptography. It is a promising way to send secret bits based on quantum mechanics. The fundamental tenet of QKD is that measuring a quantum system disturbs it, so any attempt by an eavesdropper, commonly known as Eve, to intercept the quantum signal will introduce detectable errors that the sender and receiver can detect. Key authentication is a very important part of security, especially since it's often not true that the traditional communication channel can't be jammed. A key generated through QKD is known as a "one-time pad," which is a sequence of random bits that are used to encrypt a message. Because the key is random and used only once, it provides an unbreakable encryption scheme that is resistant to all known attacks, including those that exploit weaknesses in classical computers. In the QKD protocol, a single-photon source, such as a laser, is utilized to generate a quantum signal that is transmitted through a quantum channel, which can be either an optical fiber or free space. The photons in the signal are typically polarized in one of two orthogonal states, such as vertical or horizontal. The sender randomly selects the polarization of each photon and sends it to the receiver for measurement. The receiver measures the polarization of each photon using a quantum detector, and he records his measurement results. The sender and receiver then compare a subset of their measurement results to check for errors introduced by eavesdropping, and if no errors are detected, they use the remaining measurement results to generate a shared secret key. QKD has the potential to revolutionize cryptography by providing a way to generate unbreakable encryption keys that are resistant to all known attacks. However, the technology is still in its early stages, and there are many challenges that must be overcome before it can be widely deployed, such as the need for high-quality quantum detectors and the difficulty of transmitting quantum signals over long distances.

D. Quantum Hash Function

The confidentiality of messages is treated with the utmost seriousness. There are several approaches that may be taken to increase the privacy of conversations over the airwaves. Message authentication is used to protect sensitive information in a number of different types of communication networks against active assaults. It is via the use of a technique known as Message Authentication Code that we are able to accomplish message authentication (MAC). The authentication mechanism for the message authentication code is often a hash function. A quantum hash function is a cryptographic technique that, using quantum mechanical principles, produces a hash value of a fixed size from an input of arbitrary size. Discrete quantum walks may be used as a means of enforcing quantum hash functions. Like classical random walks, in which a particle randomly hops from one point to the next, discrete quantum walks may be thought of as a quantum analogue. A qubit stands in for the particle in a discrete quantum walk, with the random motion being dictated by a specific kind of quantum operator known as the shift operator. First, the input data must be transformed into a quantum state before the discrete quantum walk can be used to generate a quantum hash function. The hash value is calculated by measuring the final quantum state, which is the outcome of a sequence of quantum walk operations performed on the initial quantum state. Any effort to tamper with the input data or the quantum walk processes will result in a different hash value, making QHF based on discrete quantum walks a secure method of hashing. Due to this feature, known as collision resistance, it is very hard for an attacker to construct two separate inputs that result in the same hash value. The creation of quantum hash functions using discrete quantum walks has showed promise, although it is still in its infancy. There is still a long way to go before their speed is optimized and their security is guaranteed for real-world use.

II. LITERATURE SURVEY

*Akwasi Adu-Kyere et.al[1]*presented their design and implementation of a communication architecture for QKD that utilizes the BB84 protocol. The main objective of this architecture is to facilitate secure communication through the distribution of keys and the detection of eavesdroppers using quantum cryptography, in line with the principles of quantum mechanics.

Shafiqul Abidin et.al[2] proved the unconditional security of quantum cryptography as well as the detection of sniffing, making it particularly ideal for future internet applications.

*Muhammad Aamir Panhwar et.al[3]*have proposed that, there is an ongoing threat or challenge of increasing the processing capabilities of existing or impending technology for current work on cryptographic algorithms such as AES, SHA, and other forms of cryptographic systems. This technological innovation also provides a technique for securing communication networks.

According to *Raj Chakrabarti et.al[4]*, the current algorithms enable multi-observable optimization by pursuing direct pathways to the Pareto front and are capable of constantly tracking the Pareto front after it is discovered in order to explore families of possible solutions. The numerical and experimental techniques introduced are also relevant to other problems that require the simultaneous handling of enormous numbers of observables, such as quantum optimum mixed state preparation.

Yalan Wang et.al[5] proposed a message authentication approach that uses the AES for encrypting and a new quantum hash function for authenticating the message. By incorporating the QHF into the message authentication technique, its security is notably improved, considering the issues that conventional encryption and authentication methods encounter as a result of the progress in computing capabilities.

Qing Zhou et.al[6] have proposed a hash function called QHFM that is built using quantum walks with one- and two-step memories on circles. Its statistical features, as well as its time and space complexity, are assessed and compared to those of the current QW-based hash functions.

*Yu-GuangYang et.al[7]*aim to create a quantum hash function (QHF) by making subtle modifications to the quantum walk (QW) mode. This QHF will enhance the security of quantum key distribution systems during the privacy amplification process by utilizing the principles of quantum mechanics. Moreover, Quantum hash function’s chaotic dynamics make it capable of generating pseudo-random numbers and supporting a new quantum hash function-based image encryption algorithm.

Yuxiang Yang et.al[8] have studied the cost of communication across non-use quantum channels. They have offered a method for transmitting the conventional channel description. Also, they have developed a lower limit on the more general task known as "remote channel simulation," which comprises a client telling a server to run a certain quantum channel.

Patrick Dreher et.al[9] presented the creation and deployment of a Docker container for the IBM Q Experience. This prototype offers a more adaptable and versatile container environment than the standard options provided by IBM, enabling researchers to develop quantum computing algorithms with greater flexibility and agility. Moreover, the prototype can be extended to other quantum computing platforms by modifying the vendor-specific software that needs to be installed and downloaded into the container.

Chris Erven[10] has demonstrated the creation of a QKD system that utilizes photon pairs to establish a secure key. The system can distribute entanglement and successfully distribute secret keys. The project's objective is to develop an independent system which is capable of uninterrupted operation.

Manju Suresh et.al[11] investigated security risks and security measures for the IoT. Their analysis showed that a significant portion of security issues on the Internet of Things occur due to the insecure channels that link wearables or mobile devices. To ensure secure transmission of information at the network layer, cryptology techniques can be employed. The researchers found that, among all cryptology algorithms, symmetric-key block cipher algorithm i.e., Blowfish algorithm is most suitable for IoT, as it offers excellent performance in terms of latency, BER, throughput, etc.

Dimitrios Alanis et.al [12] aimed to simplify routing complexity by utilizing correlations to develop pareto-optimal paths. To this end, they initially developed an optimal dynamic programming method that transformed the multi-objective routing problem into a decoding problem. But the above method was complicated, prompting them to modify it and propose the Evolutionary Quantum Pareto Optimal Routing Algorithm, which is supported by the Preinitialized-NDQIO algorithm. The EQPO algorithm allows for pareto-optimal path planning with reduced complexity.

Yong-Sheng Zhang et.al[13] introduced the quantum encryption concept and proposed a QKD scheme that uses quantum encryption. This technique may be used to encrypt any quantum state and has been used in quantum authentication.

Ramesh Bhandari[14] provided an explanation for why quantum-key distribution (QKD) is being researched, outlined the fundamental steps involved in generating a private, random key using QKD, and discussed the widely adopted BB84 QKD protocol. Additionally, they addressed the issue of key security provided by this protocol.

*Cherry Mangla et. al[15]*investigated all current issues with these technologies in 5G networks in order to resolve them for 6G secure networks. The old security system is collapsing over time as a result of the integration of quantum mechanics rules into encryption, known as quantum cryptography. We have mapped the known quantum solutions for security concerns in order to construct a secure network.

Ashwak ALabaichi et.al[16] have stated that the Blowfish algorithm presents good avalanche text. The plaintext and ciphertext have a stochastic connection according to the Blowfish algorithm.

Michael Mc Gettrick [17] has defined a new kind of quantum walk with two-step memory and investigated its properties. They discovered that it shares some similarities with the classical random walk and others with the quantum (Hadamard) walk. They have also proven that the remarkable feature of localization at the origin.

The requirement for quantum cryptography poses some challenges, so it is critical to test and analyze such systems. In order to address this, *J.R. Sahoo et.al[18]* suggest a technique that is based on a model for security analysis of the most widely used protocol, BB84.

*Peter W. Shor et.al[19]*suggested the safety of the QKD by using BB84. They presented an entanglement-based key distribution mechanism and showed that it is secure using the method employed in Lo and Chau's similar protocol. Additionally, they demonstrated that the safety of BB84 can be inferred from the security of their proposed protocol.

A two-step memory Hadamard's quantum walk was defined by *Songfeng Lu et.al[20],* and its distributions were examined numerically. Additionally, they derived a general formula for the amplitudes of this quantum walk analytically. Moreover, they introduced additional models of quantum walks with historical dependence by utilizing several coins.

III. PROPOSED WORK

Message transmission over a network with confidence requires the use of encryption techniques to ensure that the message is only accessible to authorized parties. The message is encrypted using an encryption algorithm and a secret key. To generate a secret key, we proposed the concept of quantum key distribution (QKD). It is a technique that uses the principles of quantum mechanics to generate a shared secret key between two parties that can be used to encrypt and decrypt messages securely. QKD offers a high level of security since any attempt to intercept or measure the quantum states used to generate the key would change the state of the quantum system, alerting the legitimate parties to the presence of an eavesdropper. QKD can be implemented using various physical systems, such as photons, superconducting circuits, or ions, and has been demonstrated in laboratory experiments and commercial applications. While QKD offers strong security guarantees, it is not a complete solution to secure communication, and other security measures such as message authentication and encryption algorithms may also be necessary to ensure complete security. Hash functions are used to ensure data integrity, authenticate the origin of a message, provide non-repudiation, and store passwords securely. We proposed a quantum hash function based on discrete quantum walk (DQW) to strengthen our communication. It is a specific kind of quantum algorithm that produces a message's hash value using the concepts of quantum theory. Unlike classical hash functions, which are vulnerable to attacks such as collisions and pre-image attacks, the DQW quantum hash function is believed to be resistant to these attacks due to the quantum properties used in its construction. The network routing algorithm is responsible for determining the path that a message takes through the network to reach its destination. The routing algorithm's goal is to find the most efficient and reliable path for the message while using as little network resources as possible, such as bandwidth and processing power. Using evolutionary algorithms, a category of optimization algorithm motivated by natural selection, is a typical approach to network routing. In an evolutionary routing algorithm, a population of candidate solutions is generated, and each candidate solution represents a possible path through the network. These candidate solutions are evaluated based on their fitness, which is a measure of how well they perform in terms of efficiency, reliability, and other criteria. The evolutionary algorithm then applies operators such as selection, crossover, and mutation to the population of candidate solutions to generate a new generation of solutions. This process continues until a satisfactory solution is found. We used the EQPO routing algorithm in our proposed work. The evolutionary quantum The Pareto optimal routing algorithm combines the principles of quantum mechanics and evolutionary algorithms to generate optimal routing solutions that satisfy multiple objectives simultaneously. The Pareto optimal solutions are those that are not dominated by any other solutions, meaning that they represent the best trade-offs between multiple conflicting objectives.

A. EQPO algorithm

The Evolutionary Quantum Pareto Optimal Routing Algorithm is a hybrid optimization algorithm that combines evolutionary computing and quantum computing concepts to solve multi-objective routing problems. It is designed to find a set of Pareto-optimum paths that optimize multiple objectives simultaneously, such as minimizing the total cost and maximizing the reliability of a communication network. EQPO algorithm depends on two approximations, Approximation 1 and 2.

Approximation 1: If a route is Pareto-optimum, it can only produce optimal routes according to Explication 1.

$$S_{(i)}^{surv} \triangleq S_{(i)}^{OPF} - S_{(i-1)}^{OPF}$$

Explication 1: A particular route X = {SN→ IN→DN} can be used to produce another route by adding the intermediate node IN between the preceding Intermediate Node (IN) and the Destination Node (DN).

Reliable routes: The set $S_{(i)}^{surv}$ of Reliable routes is obtained by solving the following optimization problem:

$$S_{(i)}^{surv} = \underset{x \in S_{(i)}^{gen}}{\operatorname{argmin}} \{u(x)\},$$
$$\text{s.t. } \nexists k \in S_{(i)}^{gen} \cup S_{(i-1)}^{OPF},\ : u(k) \geq u(x'),$$

Pareto-optimum routes: The collection of pareto-optimum routes in set $S_{(i)}^{OPF}$ is derived by resolving the subsequent optimization issue.

$$S_{(i)}^{OPF} = \underset{x \in S_{(i)}^{gen} \cup S_{(i-1)}^{OPF}}{\operatorname{argmin}} \{u(x)\},$$
$$\text{s.t. } \nexists k \in S_{(i)}^{gen} \cup S_{(i-1)}^{OPF},\ : u(k) \geq u\,(x'),$$

Approximation 1 limits the set of Reliable routes $S_{(i)}^{surv}$, to only include newly identified Pareto-optimum routes at ith trellis phase. This simplifies the process of identifying both the Reliable and Pareto-optimum route sets by solving a single optimization problem. Assertion 1 does not contradict Approximation 1 since the pareto-optimum routes always contain pareto-optimum sub-routes, which are dominating because the last hop is missing, resulting in increased evaluated Utility Functions. This implies that if a specified route from the initial node to the final node has no dominant routes, its sub-route will also be free of dominant routes.

Assertion 1: It ensures that if a sub-route x = SN → I_i is not the most efficient compared to other valid sub-routes, then the specific path x = {SN → I_i →DN} will not able to produce non-dominated paths by inserting an intermediate node IN between its preceding IN and the destination node. If there exist any routes x_d in the set S such that the utility function u(x_d) is better or equal to u(x) but worse than u(x), then it can be concluded that if x is suboptimal, then x will also be sub-optimal. This property may be used to reduce the search space needed to discover the complete OPF.

Approximation 2:

To make it easier to find all Pareto-optimum routes, Explication 1 is modified in the following way: a specific route x can produce another route $x_g^{(k,j)}$by adding a new IN I_j midst jth and (j+1)th nodes. Approximation 2 broadens the collection of $S_{(i)}^{gen}$produced routes, which are constructed from the collection of $S_{(i-1)}^{surv}$ reliable paths from preceding phase of the given graph. The above is accomplished by restoring a direct link formed by two INs or an IN and the Destination node via an indirect link including a suitable IN as an intermediary node.

Evolutionary Quantum Pareto Optimal Routing Algorithm:

1. Initialize three sets: $S_{(0)}^{gen}$, $S_{(0)}^{OPF}$, and $S_{(0)}^{surv}$, all containing only the route from SN to DN. Set i to 0.
2. Repeat the following steps until either the maximum number of nodes is reached or there are no more reliable routes to generate:
3. Increase i by 1.
4. Produce a new set of acceptable routes from the$S_{(i)}^{gen}$ collection of reliable routes, from the previous iteration, $S_{(i-1)}^{surv}$using Approximation 2 to add a single IN between two intermediate nodes.
5. Add the previously identified Pareto-optimal routes, $S_{(n-1)}^{OPF}$, to the new set $S_{(i)}^{gen}$.
6. Use the Preinitialized NDQIO algorithm to identify the Pareto-optimum routes in set $S_{(i)}^{gen}$ and initialize the identified optimal Pareto-front to $S_{(n)}^{OPF}$, where n is the current iteration.
7. Set the new set of reliable routes, $S_{(i)}^{surv}$, to the difference between the identified OPF $S_{(i)}^{OPF}$ and the previous OPF $S_{(n-1)}^{OPF}$.
8. End the loop if there are no more reliable routes to generate (i.e., the size of the set $S_{(i)}^{surv}$ is 0) or the maximum number of nodes has been reached.
9. Export the final identified OPF, $S_{(n)}^{OPF}$, and terminate the algorithm.

The initial step of the above algorithm starts by creating a collection of routes, which includes pareto-optimum and reliable routes, and setting the direct route {SN →DN} as the default route. Then, the algorithm proceeds to the trellis phases, using Steps 2–8 of the algorithm. After completing each trellis phase, Step 4 of the algorithm creates the collection of $S_{(i)}^{gen}$ of produced paths. This set is then combined with the set $S_{(i-1)}^{OPF}$of pareto-optimum routes from the preceding phase in Step 5. In Step 6, the P-NDQIO method is used to determine the collection of

$S_{(i)}^{OPF}$ non-dominated paths from the collection of $S_{(i)}^{gen}$. In Step 7, the collection $S_{(i)}^{surv}$ of reliable paths is obtained using Approximation 1.

Below are the steps involved in a single trellis phase of the Evolutionary Quantum Pareto Optimal Routing Algorithm.

Generation of Routes:

The EQPO algorithm creates a set of routes called $S_{(i)}^{gen}$using the reliable routes from the preceding trellis phase, as shown in Step 4 of above Algorithm, by applying Approximation 2. For example, consider the diagram with 5 nodes where the route $\{1 \rightarrow 2 \rightarrow 5\}$ can produce four alternative routes: $\{1 \rightarrow 2 \rightarrow 3 \rightarrow 5\}$, $\{1 \rightarrow 2 \rightarrow 4 \rightarrow 5\}$, $\{1 \rightarrow 3 \rightarrow 2 \rightarrow 5\}$, and $\{1 \rightarrow 4 \rightarrow 2 \rightarrow 5\}$.

Pareto-optimum and Reliable Routes:

After generating the network of routes$S_{(i)}^{gen}$, the above algorithm uses the Preinitialized NDQIO algorithm in Step 6 to identify new pareto-optimum routes from the generated set. However, the optimality of a route is determined by the set of suitable routes considered. Therefore, the network of pareto-optimum paths identified in all preceding trellis phases, denoted by $S_{(i-1)}^{OPF}$,must be concatenated with the set of generated routes, denoted by $S_{(i)}^{gen}$, in Step 5 of the algorithm. This ensures that the paths that are considered optimal by the Preinitialized NDQIO algorithm are actually pareto-optimum in terms of complete set of valid paths. The set $S_{(i)}^{OPF}$ comprises the pareto-optimum paths from all trellis phases up to the i-th stage. Therefore, the pareto-optimum routes found in the current trellis phase using Approximation 1 are regarded as reliable routes. However, Pareto-optimal routes found in previous phases are not considered since they would produce routes that have already been processed in earlier trellis phases.

Explication 2:

A specific path xi, linked with the UV $u(x_1)$, is pareto-optimum only if no other route exists that surpasses it, meaning there is no x_k where $u(x_k)$ is greater than $u(x_i)$. Alternatively, if there is no x_k that weakly dominates xi, meaning $u(x_i)$ is not less than or equal to $u(x_k)$, then x_i is considered to be strongly Pareto-optimum.

The above Explication explains the requirements for the optimality of this particular path. Pareto Fronts are clusters of routes that share a large number of dominant routes (PF). Explication 2 specifies that the Optimal Pareto Front (PF) consists of the pareto-optimum paths that are not dominated by other paths (OPF). Our goal is to discover all weakly Pareto-optimum routes for the Utility Functions under consideration so that we may better understand the routing trade-offs involved.

B. Quantum Key Distribution

Implementations of one-time pad cryptosystem included a flaw that made it difficult for two users to exchange a secret key. Quantum cryptography, which is another name for quantum mechanics, is the method that QKD employs to overcome this problem. In QKD, the quantum mechanics are responsible for providing the key with security. Several other proposals for QKD protocols have been made, such as BB84, BBM92, E91, and B92. BB84 is the only one of them that is a popular choice for a QKD scheme. The use of QKD may improve the network security of many different types of networks, including satellite, optical, and terrestrial wireless networks.

a. BB84 algorithm

The BB84 protocol in quantum key distribution involves encoding each bit of the secret key onto the quantum state of a single qubit. Measuring the state of a qubit in a particular basis (e.g., the Z-basis or X-basis) will provide one of two possible outcomes (0 or 1), but any measurement in an arbitrary basis will yield a random result. Therefore, if an eavesdropper tries to measure a qubit to determine its state, they will inevitably introduce errors into the system, which can be detected by the legitimate parties. To avoid detection, the eavesdropper will need to guess which basis to measure in, but this will result in a lower probability of successful interception. As a result, the key distribution protocol's security is based on the principles of quantum physics, which prohibit covert listening.

1. Qubits Transmissions

i. The sender chooses, accordingly, a random string of bits a € {0, 1}, and a random string of basis b^s€ {0, 1}.

ii. For each bit a_i in a and b^si in b^s, the sender prepares a qubit in the quantum state in q_{ij} as illustrated in Table 1, and delivers it to the recipient through the quantum channel. The qubit will only stay in its original state if bit an equal to 0 and basis b^s equals 0. The qubit will be encoded in an X gate if bit an equal 0 and basis b^s equals 1. The qubit will be encoded in the H gate if bit an equal 1 and basis b^s equals 0. A qubit will be encoded in both the X gate and the H gate if bit an equal 1 and basis b^s equals 1.

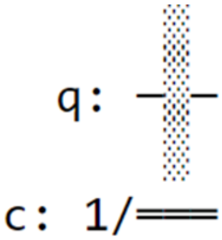


Figure 3.2.1.1: Qubit encoded in |0⟩ state

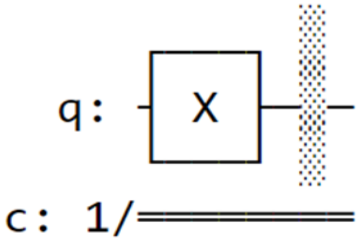


Figure 3.2.1.2: Qubit encoded in |+⟩ state

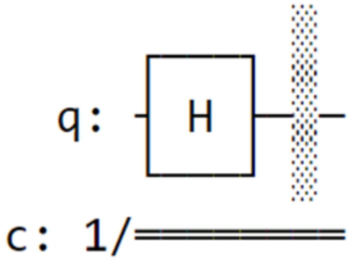


Figure 3.2.1.3: Qubit encoded in |1⟩ state

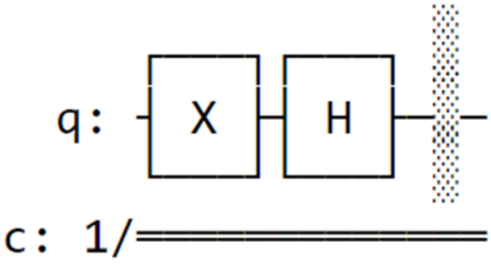


Figure 3.2.1.4: Qubit encoded in |-⟩ state

iii. With respect to the qubits sent by the sender, the receiver measures each qubit's q_i with his basis b^r_i. The receiver obtains the value string a' € {0, 1} from the measurements, and his selection of bases would take the form b^r € {0, 1}. Qubits are measured in the Z-basis if the basis is 0, otherwise they are measured in the X-basis.

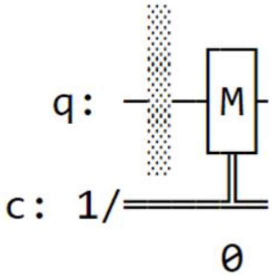


Figure 3.2.1.5: |0⟩ state measured in Z basis

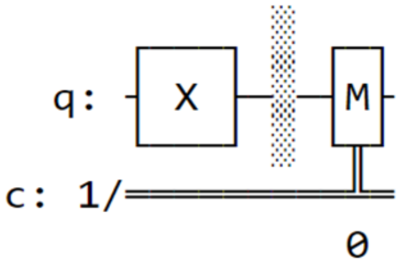


Figure 3.2.1.6: |+⟩ state measured in Z basis

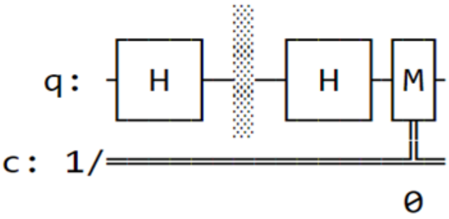


Figure 3.2.1.7: |1⟩ state measured in X-basis

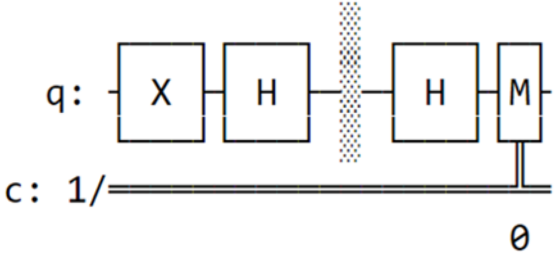


Figure 3.2.1.8: |-⟩ state measured in X-basis

2. Public Discussion

i. The sender and the receiver disclose their bases to each other.

ii. Receivers utilize qubit measurements as part of their shared secret key if they are performed using the same methodology as the sender; otherwise, they discard the information for that bit i.e., both a_i and a'_i are discarded if b^s_i not equals to $b^r_{i.}$

iii. The string of bits left in a and a' after the bits exposed in step 2 are eliminated is the shared secret key, QK € {0,1}.

What happens if, in between, there's an eavesdropper who listens in on their communication? In order to extract information from the communication, an eavesdropper would have to perform a measurement on a qubit. A quantum measurement generally changes the state of a quantum system unless the quantum state happens to be an eigenstate of the observable that you measure. In that case, you can perform a measurement without changing the state, so the eavesdropper, of course, could try to measure on a suitable basis in order to minimize the disturbance of the qubit, but there's no way that the eavesdropper can always do that because the qubit in transit from Alice to Bob is in one of the four states. 0 or 1, or + or -: if the qubit was only in one of the two standard basis states (0 or 1), then the eavesdropper could simply measure on the standard basis, and this measurement would not disturb the quantum state. But the qubit can also be in the + or - state. So, if the eavesdropper measures on the standard basis but the qubit happens to be in a plus or minus state, then the measurement will disturb the state of the qubit.

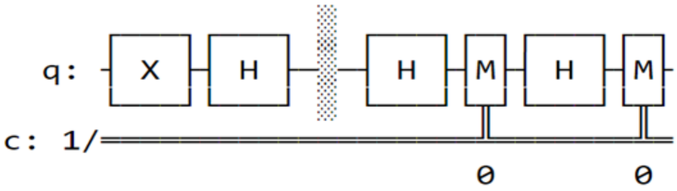


Figure 3.2.2.1: Qubit changes from one state to another

C. *Message Authentication*

1. **Blowfish Algorithm:**

The encryption mechanism for message authentication should be reversible. As a result, we consider the symmetric encryption strategy. During communication, we may encode and decode messages using the same keys. We made advantage of BB84's quantum keys in this situation. The block cypher and the stream cypher are the two different encryption techniques that make up symmetric encryption. We shall emphasize the block cypher in our plan. First, the information will be transformed into a series of binary numbers in a block cypher method. The binary digits will then be divided into a number of blocks with equal sizes. Last but not least, each data block should be encrypted using key calculations. This is how the block-cipher system operates as a whole.

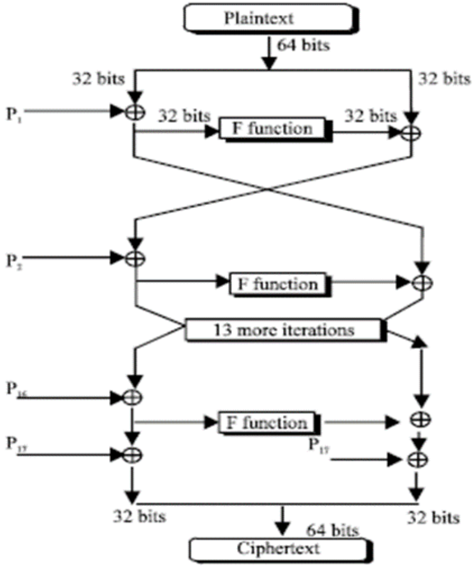


Figure 3.3.1: Whole process of Blowfish algorithm

We chose the Blowfish encryption technique to serve as the encryption process in our system in consideration of the security need. Blowfish supports key sizes from four bits to 56 bits in 8-bit increments and works on 64-bit blocks of data. The algorithm creates a number of subkeys that are employed in the encryption and decryption procedures via the key-expansion process. The Blowfish algorithm has a fixed number of rounds of 16. Each cycle modifies the material being encrypted or decrypted using a new 32-bit subkey. Several rounds with diverse subkeys serve to increase the algorithm's security and resistance to different kinds of assaults. The round function in Blowfish performs a number of actions on the input data using the round subkey, including replacements, permutations, and XOR operations. Sixteen Blowfish iterations are performed using a Feistel network architecture, in which the input block is divided in half and the round function is applied to each half in turn. Blowfish's 16-round architecture, which successfully balances security and speed, makes it a popular option for a variety of applications. Figure 3.3.1 explains how the Blowfish algorithm works as a whole.

2. **Discrete Quantum Walk**

The quantum walk concept is a key part of our paper's discussion of the quantum hash function. As much interest as possible has been shown in the quantum walk, which is like a traditional random walk. There are two main types of quantum walk, continuous-time and discrete. We place a high value on the discrete-time quantum walk, regarded as the one-way controlled quantum walk with one- and two-step memory (CQWM). The discrete-time quantum walk uses coin operators to control the walkers. The CQWM occurs within the Hilbert space $\mathscr{H}_w \otimes \mathscr{H}_{d2} \otimes \mathscr{H}_{d1} \otimes \mathscr{H}_c$, which is composed by vectors $|a, d2, d1, c\rangle$. Here, c is designated as the coin state, while d1 and d2 stand for the first and second preceding steps' directions (where 0 represents the left and 1 is the right), and 'a' depicts the present position. The walker's location may be expressed as 'a' $\in$ Z (the set of all integers) if it follows a linear route. However, if the walker is moving on a cyclic path consisting of n nodes, its position 'a' can be represented from 0 to n-1.

Mathematically, a k length bit string message, denoted by M = (msg1, msg2, ⋯, msg_k) $\in$ {0, 1}and k is the length of the message, regulates the CQWM's development. The composite of k unitary transformations is how the evolution is defined.

$$\hat{U}_{msg}=\hat{U}(^{msg_k}) \cdot \hat{U}(^{msg_{k-1}}) \ldots \hat{U}(^{msg_2}) \cdot \hat{U}(^{msg_1}) \quad (1)$$

The one-step translation of jth bit of the message represents $\hat{U}(^{msg_j})$ (where $1 \leqslant j \leqslant t$), which is signified by the and is defined as:

$$\hat{U}^{(msg_j)} = Z \cdot \left(I_n \otimes D^{(msg_j)}\right) \cdot \left(I_{4n} \otimes C^{(msg_j)}\right) \quad (2)$$

If m_j equals zero, the parameterized angle α is used to define $C^{(msg_j)}$, which is a 2×2 matrix coin operator controlled by msg_j. If m_j equals one, the parameterized angle β is used.

$$C^{(0)} = \begin{bmatrix} cos \propto & sin \propto \\ sin \propto & -cos \propto \end{bmatrix} C^{(1)} = \begin{bmatrix} cos\beta & sin\beta \\ sin\beta & -cos\beta \end{bmatrix}$$

The operator I_z, with z equal to either n or 4n, is an identity operator with dimensions of k × k. Additionally, msg_j controls the 8 × 8 direction decide operator $D^{(msg_j)}$, whereas the next direction controls the conditional shift operator Z.

$$Z = \sum(|a+1\rangle\langle a| \otimes |0\rangle\langle 0| + |a-1\rangle\langle a| \otimes |1\rangle\langle 1|)$$

QHF Algorithm:

The mentioned hash method is built by executing CQWM on an n-node cycle, in which every node provides m bits towards the message digest value QH(M) based on the input message M. The method of a hash function based on CQWM is explained as follows:

1. Choose the parameter values of (n, m, l, α, β, θ) with the following conditions are met: the number of nodes in a cycle is denoted by n and it is an odd number, and n × m is the hash value's bit length, θ is the parameter that determines the initial coin state, while the parameters of two coins operator governing the quantum walk are α, and β and (0<α, β, θ < Π/2). C1 and C2 are the two-coin operators. If the message bit is 1 then it decides C1 else it determines C2.

$$C_1=\begin{bmatrix} cos \propto & sin \propto \\ sin \propto & -cos \propto \end{bmatrix} C_2=\begin{bmatrix} cos\beta & sin\beta \\ sin\beta & -cos\beta \end{bmatrix}$$

2. Set up the walker to start in the state$|\psi_0\rangle = \cos\theta|0\rangle + \sin\theta|1\rangle$.
3. Compute the resultant probability by applying $\hat{U}_{msg}$ to $|\psi 0\rangle$.
4. To create a binary string that gives as the hash value, amplify every value in the resultant probability by 10^l and keep exactly their decimal portion modulo 2^m where l≥m. The hash value contains nm bits in it.
5. Repeat step 2 nine times, changing the alpha and beta values for each hash value produced at step 4 to ensure that the results are unique.
6. The hash value generated at the ninth step is the final hash value for the message.

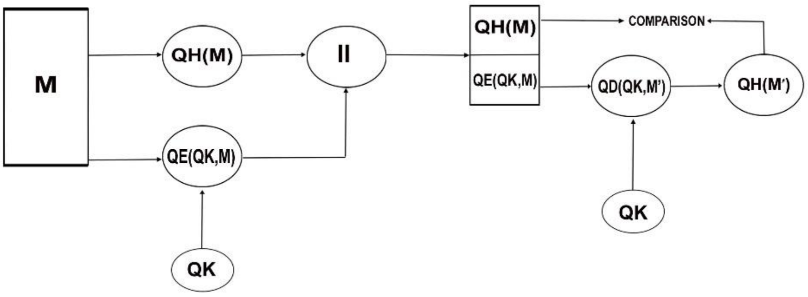


Figure 3.3.2: The is the explicit message authentication procedure: M is the message that will be sent from the sender to the recipient. The message digest of M is found by using the authentication method called the quantum hash function QH(M). By using the quantum keys QK and the Blowfish algorithm to make the ciphertext, the message M is encrypted. "||" is an operation used to cascade the message digest and the ciphertext of M. QE is the encryption of the message done by using QE, and QD is the decryption of the message done by using QE.

Process of Message Authentication:

- Before to anything further, confirm that the quantum keys (QK) and parameters (n, m, l, α, β, θ) between the sender and the receiver have the appropriate values. The coin operators that will operate the quantum walk are chosen by the first message M. Therefore, we may get the transformation of the probability distribution into the quantum hash value QH(M).
- With the original message M's security in mind, the block cypher method Blowfish is used to encrypt it, using the quantum key QK to produce the ciphertext QE (QK, M). The sender used the quantum hash value QH(M) to cascade the ciphertext, yielding the result QE(QK, M)||QH (M). The result is sent via the channel by the sender.
- The receiver initially uses the same quantum key QK to decode the ciphertext QE(QK,M) using the Blowfish method after obtaining the result QE(QK, M) || QH(M). The receiver then applies the parameters (n, m, l, α, β, θ) to the decrypted text M' in order to run the quantum hash function and get QH(M'). The receiver then analyses QH(M) readings with QH(M'). It is acceptable to assume that the original message M was transmitted by the sender and not falsified or altered if QH(M) = QH(M'). The message M, however, is likely to be forged, altered, or not have been sent by the sender if QH(M) is not equal to QH(M').

This is how the explicit message authentication approach works overall.

IV. RESULTS

In our scheme, we created ten nodes in the network for the simulation, labelled from n1 to n10. These nodes are connected by the links as follows:

- n1 is connected to n2, n8, and n9
- n2 is connected to n3
- n3 is connected to n4
- n4 is connected to n5
- n5 is connected to n6 and n10
- n6 is connected to n7
- n7 is connected to n8
- n8 is connected to n1
- n9 is connected to n10
- n10 is connected to n5

The figure.4.1 that follows provides an illustration of this network. Each link has different attributes that could affect the routing decisions, such as bit error rate (BER), delay, and throughput. For example, n1 to n2 may be low BER but have high delay, whereas n2 to n4 may be high BER but have high throughput. The links at each node in our simulation are shown in the figure 4.2.

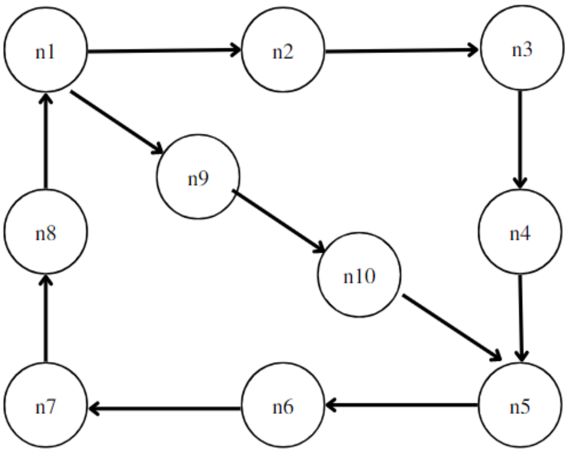
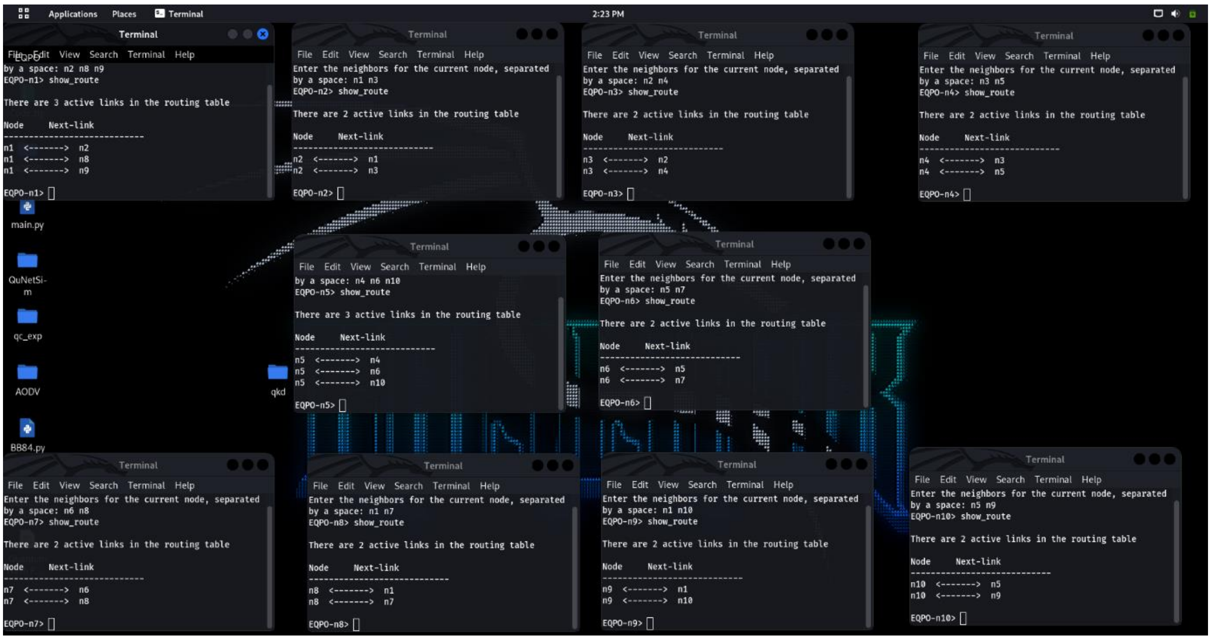


Figure: 4.1

In our case, the EQPO routing algorithm could be used to find the best routing paths for communication in this network. Each routing path would be shown as a string or list of binary digits in EQPO, each of which would stand for a network node. For example, a routing path from node n1 to node n5 could be represented as 0101, where "0" represents the choice to take the upper path (if available) at a decision point and "1" represents the choice to take the lower path (if available). This shows that the path goes from n1 to n5 via n9 and n10.



```
by a space: n2 n8 n9
EQPO-n1> show_route

There are 3 active links in the routing table

Node     Next-link
----------------------------
n1  <------->  n2
n1  <------->  n8
n1  <------->  n9

EQPO-n1> 
```

```
Enter the neighbors for the current node, separated
by a space: n1 n3
EQPO-n2> show_route

There are 2 active links in the routing table

Node     Next-link
----------------------------
n2  <------->  n1
n2  <------->  n3

EQPO-n2> 
```

```
Enter the neighbors for the current node, separated
by a space: n2 n4
EQPO-n3> show_route

There are 2 active links in the routing table

Node     Next-link
----------------------------
n3  <------->  n2
n3  <------->  n4

EQPO-n3> 
```

```
Enter the neighbors for the current node, separated
by a space: n3 n5
EQPO-n4> show_route

There are 2 active links in the routing table

Node     Next-link
----------------------------
n4  <------->  n3
n4  <------->  n5

EQPO-n4> 
```

```
by a space: n4 n6 n10
EQPO-n5> show_route

There are 3 active links in the routing table

Node     Next-link
----------------------------
n5  <------->  n4
n5  <------->  n6
n5  <------->  n10

EQPO-n5> 
```

```
Enter the neighbors for the current node, separated
by a space: n5 n7
EQPO-n6> show_route

There are 2 active links in the routing table

Node     Next-link
----------------------------
n6  <------->  n5
n6  <------->  n7

EQPO-n6> 
```

```
Enter the neighbors for the current node, separated
by a space: n6 n8
EQPO-n7> show_route

There are 2 active links in the routing table

Node     Next-link
----------------------------
n7  <------->  n6
n7  <------->  n8

EQPO-n7> 
```

```
Enter the neighbors for the current node, separated
by a space: n1 n7
EQPO-n8> show_route

There are 2 active links in the routing table

Node     Next-link
----------------------------
n8  <------->  n1
n8  <------->  n7

EQPO-n8> 
```

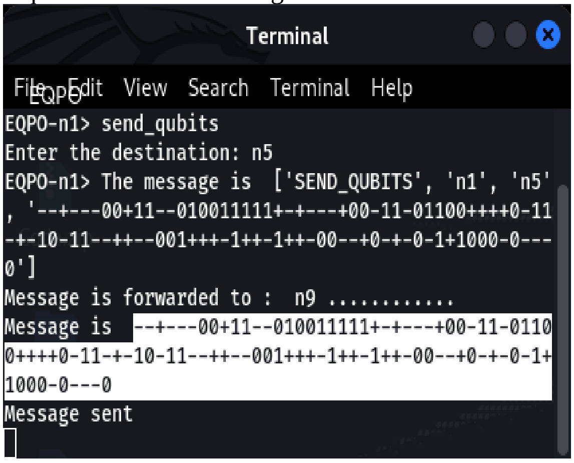
```
Enter the neighbors for the current node, separated
by a space: n1 n10
EQPO-n9> show_route

There are 2 active links in the routing table

Node     Next-link
----------------------------
n9  <------->  n1
n9  <------->  n10

EQPO-n9> 
```

```
Enter the neighbors for the current node, separated
by a space: n5 n9
EQPO-n10> show_route

There are 2 active links in the routing table

Node     Next-link
----------------------------
n10  <------->  n5
n10  <------->  n9

EQPO-n10> 
```

Figure:4.2

Then, we used the EQPO routing algorithm to make a population of possible solutions (routing paths) for communication in the network, evaluated their utility function based on multiple goals, used quantum operators to make the solutions change, and chose the fittest individuals for genetic operations like crossover and mutation. Through the iterative application of EQPO routing algorithm, we found Pareto optimal solutions that balance multiple objectives, such as minimizing the BER and latency of the routing paths while maximizing the overall throughput of the network.

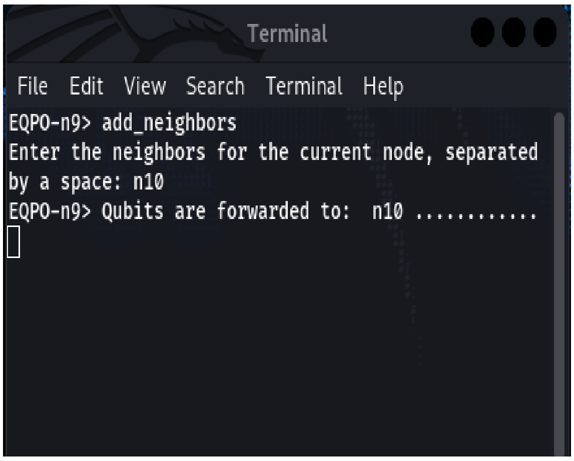
We used quantum key distribution to create a variable-length quantum key between the network's sender and receiver nodes before they could communicate with each other or send messages. This was done for security reasons. In this instance, n1 and n5 served, respectively, as the transmitter and recipient nodes. Node n1 makes a 100-bit list of random 0s and 1s. Each bit can be either 0 or 1. For each bit, node n1 will arbitrarily select either the standard basis or the computational basis to encode it in. The standard basis is represented by {|0⟩, |1⟩} and the computational basis is represented by {|+⟩, |-⟩}, where |+⟩ = 1/ √ 2 (|0⟩+|1⟩) and |-⟩ = 1/ √ 2 (|0⟩-|1⟩). Node n1 then sends the sequence of 100 qubits to node n5 through the channel. This was shown in the figure 4.3.



```
EQPO-n1> send_qubits
Enter the destination: n5
EQPO-n1> The message is  ['SEND_QUBITS', 'n1', 'n5'
, '--+---00+11--010011111+-+---+00-11-01100++++0-11
-+-10-11--++--001+++-1++-1++-00--+0-+-0-1+1000-0---
0']
Message is forwarded to :  n9 ...........
Message is  --+---00+11--010011111+-+---+00-11-0110
0++++0-11-+-10-11--++--001+++-1++-1++-00--+0-+-0-1+
1000-0---0
Message sent
```

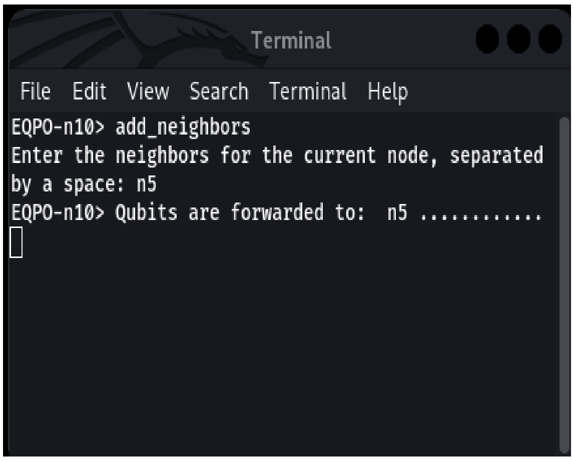
Figure:4.3

The qubits in question were traded with one another in the network channel. Due to the fact that there is no direct link from node n1 to node n5, the EQPO routing algorithm utilizes the path that goes via nodes n9 and n10. First, the qubits were sent from the node n1 to the node n9, as illustrated in the figure 4.4, and then, from the node n9, the qubits were sent to the destination node n5, as shown in figure 4.5.



```
EQPO-n9> add_neighbors
Enter the neighbors for the current node, separated
by a space: n10
EQPO-n9> Qubits are forwarded to:  n10 ...........
```

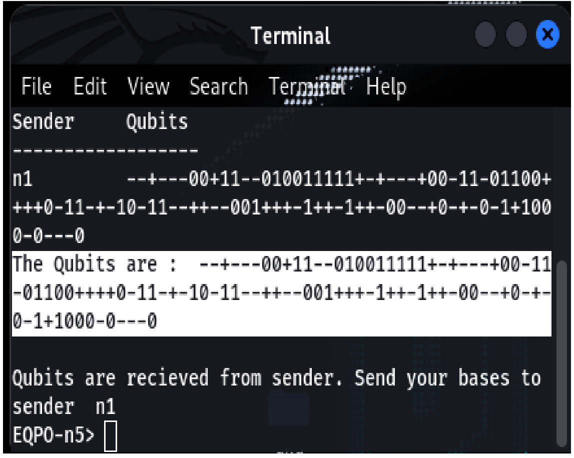
Figure:4.4



```
EQPO-n10> add_neighbors
Enter the neighbors for the current node, separated
by a space: n5
EQPO-n10> Qubits are forwarded to:  n5 ...........
```

Figure:4.5

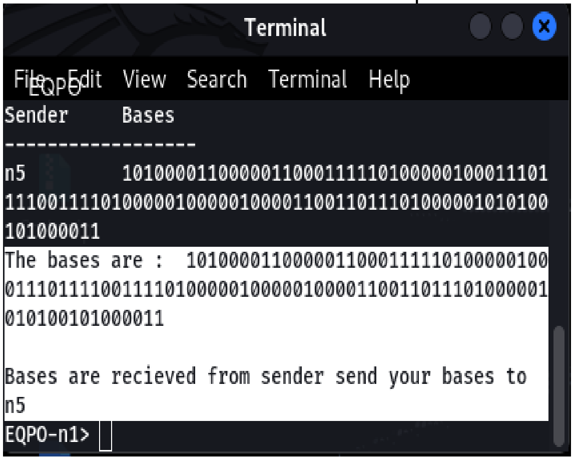
In the figure 4.6, we can see that, node n5 received the qubits from node n1. After the receiving of qubits from node n1, node n5 also makes a list of 100 random bits and chooses a random basis for each qubit it gets from node n1. Node n5 measures each qubit on the selected basis and records the result, which is either 0 or 1.



```
Sender      Qubits
------------------
n1          --+---00+11--010011111+-+---+00-11-01100+
+++0-11-+-10-11--++--001+++-1++-1++-00--+0-+-0-1+100
0-0---0
The Qubits are :  --+---00+11--010011111+-+---+00-11
-01100++++0-11-+-10-11--++--001+++-1++-1++-00--+0-+-
0-1+1000-0---0

Qubits are recieved from sender. Send your bases to
sender  n1
EQPO-n5>
```

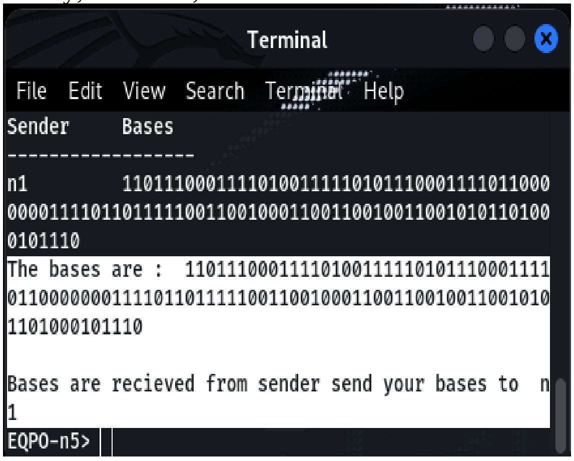
Figure:4.6

After each qubit has been measured, Node n5 keeps the data about the measurements secret. Next, nodes n1 and n5 were able to disclose the basis they had chosen to use for each qubit; this could be seen from the figures 4.7 and 4.8. If node n5's measurement of a qubit was based on the same basis as node n1's preparation of the qubit, then the information for that bit was utilized as part of the shared secret key; otherwise, it was discarded.



```
Sender      Bases
------------------
n5          1010000110000011000111110100000100011101
1110011110100000100000100001100110111010000001010100
101000011
The bases are :  1010000110000011000111110100000100
0111011110011110100000100000100001100110111010000001
010100101000011

Bases are recieved from sender send your bases to
n5
EQPO-n1>
```

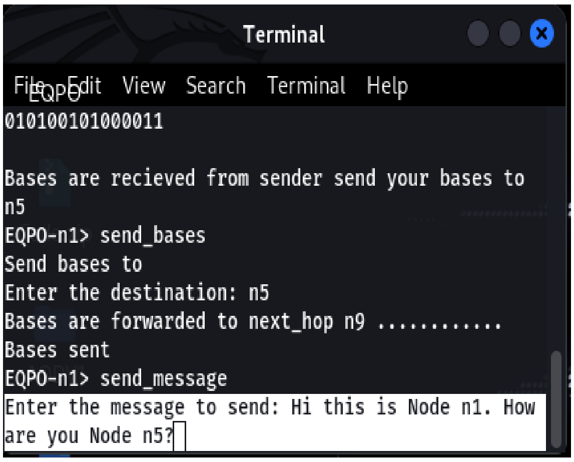
Figure:4.7



```
Sender      Bases
------------------
n1          1101110001111010011111010111000111101100 0
0000111101101111110011001000110011001001100101011010 0
0101110
The bases are :  1101110001111010011111010111000111 1
0110000000111101101111100110010001100110010011001010
1101000101110

Bases are recieved from sender send your bases to  n
1
EQPO-n5>
```

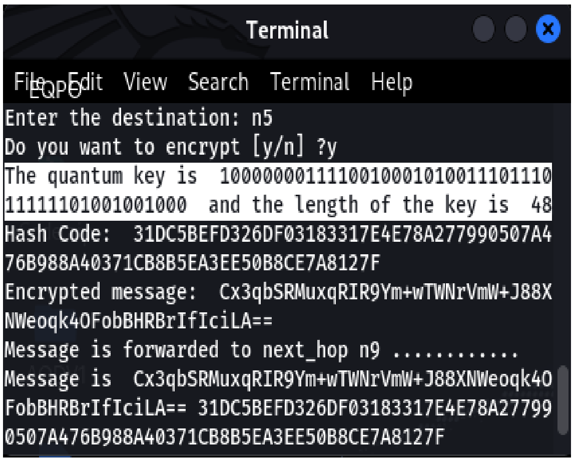
Figure:4.8

When one party has successfully sent the qubits and bases to the other, it is now time for the nodes to communicate with one another. The node labelled "n1" in Figure 4.9 is going to transmit the message "Hi this is Node n1. How are you Node n5?". Node n1 desires to transmit the information in a safe manner, and in order to achieve this goal, n1 has computed a quantum key with a length of 48 bits. The length of the quantum key as well as the quantum key itself were both configurable. As can be seen in figure 4.12, the message was successfully encrypted with the assistance of this quantum key and by making use of the general idea behind the Blowfish encryption technique. Figure 4.10 allows us to see the quantum key that was used throughout the encryption procedure, and we were able to see it. A hash code was computed in order to provide the communication a higher level of protection against being read by malicious hackers. The hash code was computed by using the quantum hash function in conjunction with the notion of discrete quantum walking. The quantum hash function produced a value of 296 bits, which is equivalent to a number of 74 hexadecimal digits. Figure 4.13 demonstrates this hash value for the reference. When the computations for the message digest and encryption had been completed, the encrypted message and the hash value were combined together with a space in between them, and then they were sent to node n5 through a network channel.



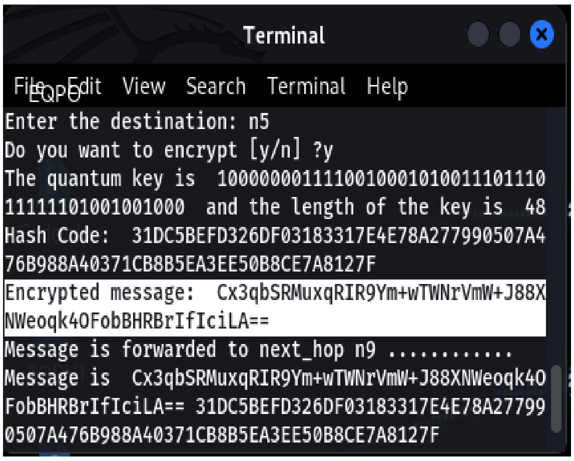
```
010100101000011

Bases are recieved from sender send your bases to
n5
EQPO-n1> send_bases
Send bases to
Enter the destination: n5
Bases are forwarded to next_hop n9 ...........
Bases sent
EQPO-n1> send_message
Enter the message to send: Hi this is Node n1. How
are you Node n5?
```

Figure:4.9



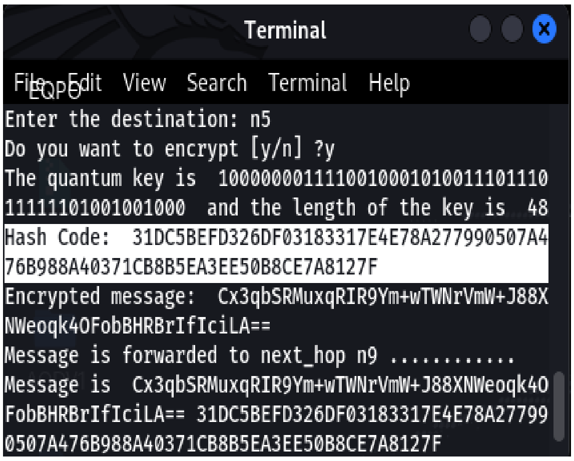
```
Enter the destination: n5
Do you want to encrypt [y/n] ?y
The quantum key is  1000000011110010001010011101110
11111101001001000  and the length of the key is  48
Hash Code:  31DC5BEFD326DF03183317E4E78A277990507A4
76B988A40371CB8B5EA3EE50B8CE7A8127F
Encrypted message:  Cx3qbSRMuxqRIR9Ym+wTWNrVmW+J88X
NWeoqk4OFobBHRBrIfIciLA==
Message is forwarded to next_hop n9 ...........
Message is  Cx3qbSRMuxqRIR9Ym+wTWNrVmW+J88XNWeoqk4O
FobBHRBrIfIciLA== 31DC5BEFD326DF03183317E4E78A27799
0507A476B988A40371CB8B5EA3EE50B8CE7A8127F
```

Figure:4.10



```
Enter the destination: n5
Do you want to encrypt [y/n] ?y
The quantum key is  1000000011110010001010011101110
11111101001001000  and the length of the key is  48
Hash Code:  31DC5BEFD326DF03183317E4E78A277990507A4
76B988A40371CB8B5EA3EE50B8CE7A8127F
Encrypted message:  Cx3qbSRMuxqRIR9Ym+wTWNrVmW+J88X
NWeoqk4OFobBHRBrIfIciLA==
Message is forwarded to next_hop n9 ...........
Message is  Cx3qbSRMuxqRIR9Ym+wTWNrVmW+J88XNWeoqk4O
FobBHRBrIfIciLA== 31DC5BEFD326DF03183317E4E78A27799
0507A476B988A40371CB8B5EA3EE50B8CE7A8127F
```

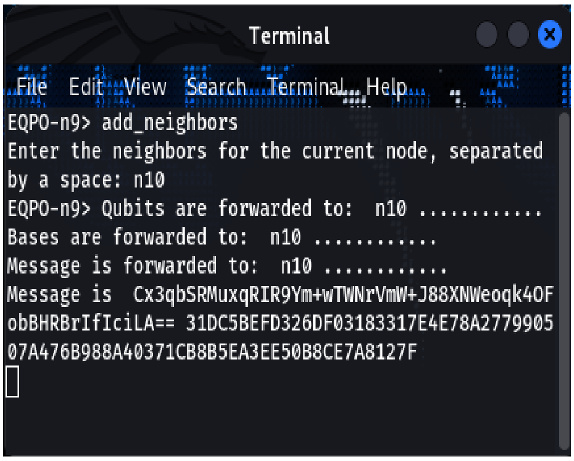
Figure:4.11



```
Enter the destination: n5
Do you want to encrypt [y/n] ?y
The quantum key is  1000000011110010001010011101110
11111101001001000  and the length of the key is  48
Hash Code:  31DC5BEFD326DF03183317E4E78A277990507A4
76B988A40371CB8B5EA3EE50B8CE7A8127F
Encrypted message:  Cx3qbSRMuxqRIR9Ym+wTWNrVmW+J88X
NWeoqk4OFobBHRBrIfIciLA==
Message is forwarded to next_hop n9 ...........
Message is  Cx3qbSRMuxqRIR9Ym+wTWNrVmW+J88XNWeoqk4O
FobBHRBrIfIciLA== 31DC5BEFD326DF03183317E4E78A27799
0507A476B988A40371CB8B5EA3EE50B8CE7A8127F
```

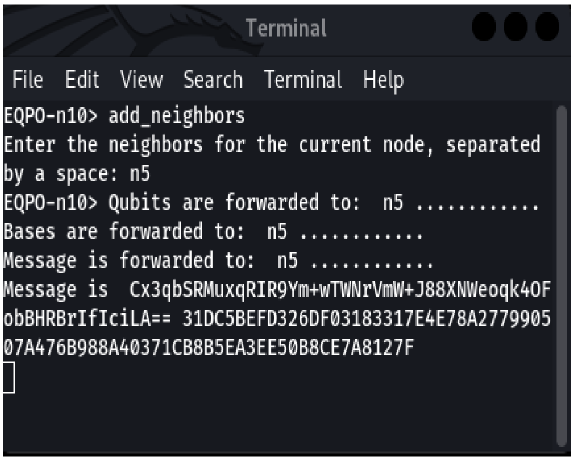
Figure:4.12

As it mentioned before, there was no connection that ran directly from node n1 to node n5. Once again, the EQPO routing algorithm chose the route that traverses via nodes n9 and n10 in this scenario. After then, the concatenated message was transmitted from node n1 to node n9, where it was received. It is clear to us that node n9 received the combined message from node n1, and that node n9 then passed the combined message to node n10 without engaging in any activities that would have been considered suspicious at the time. A representation of this might be seen in Figure 4.13. Node n10 was the recipient of the message that was sent from node n9, and it transmitted the combined message to node n5 without doing any steps that were suspicious. A depiction of this could potentially be seen in Figure 4.14.



```
EQPO-n9> add_neighbors
Enter the neighbors for the current node, separated
by a space: n10
EQPO-n9> Qubits are forwarded to:  n10 ...........
Bases are forwarded to:  n10 ...........
Message is forwarded to:  n10 ...........
Message is  Cx3qbSRMuxqRIR9Ym+wTWNrVmW+J88XNWeoqk4OF
obBHRBrIfIciLA== 31DC5BEFD326DF03183317E4E78A2779905
07A476B988A40371CB8B5EA3EE50B8CE7A8127F
```

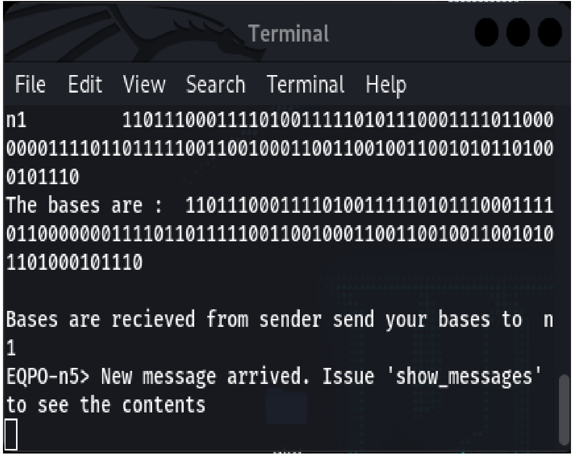
Figure:4.13



```
EQPO-n10> add_neighbors
Enter the neighbors for the current node, separated
by a space: n5
EQPO-n10> Qubits are forwarded to:  n5 ...........
Bases are forwarded to:  n5 ...........
Message is forwarded to:  n5 ...........
Message is  Cx3qbSRMuxqRIR9Ym+wTWNrVmW+J88XNWeoqk4OF
obBHRBrIfIciLA== 31DC5BEFD326DF03183317E4E78A2779905
07A476B988A40371CB8B5EA3EE50B8CE7A8127F
```

Figure:4.14

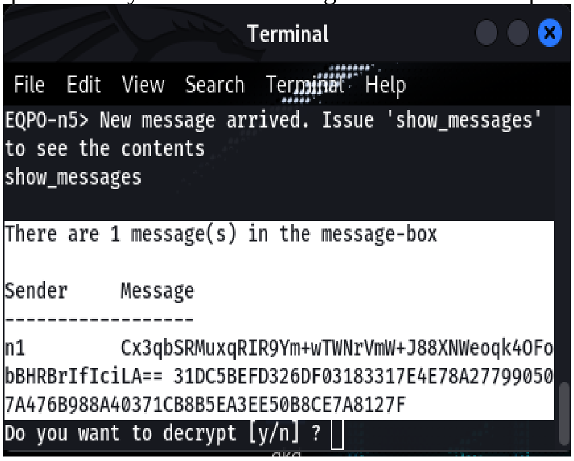
Figure 4.15 makes it possible for us to see that the combined message, which had been transmitted by node n10, was received by the target node n5, as shown.



```
n1          11011100011110100111110101110001111011000
00001111011011111001100100011001100100110010101101000
0101110
The bases are :  110111000111101001111101011100011110
01100000001111011011111001100100011001100100110010101
1101000101110

Bases are recieved from sender send your bases to  n
1
EQPO-n5> New message arrived. Issue 'show_messages'
to see the contents
```

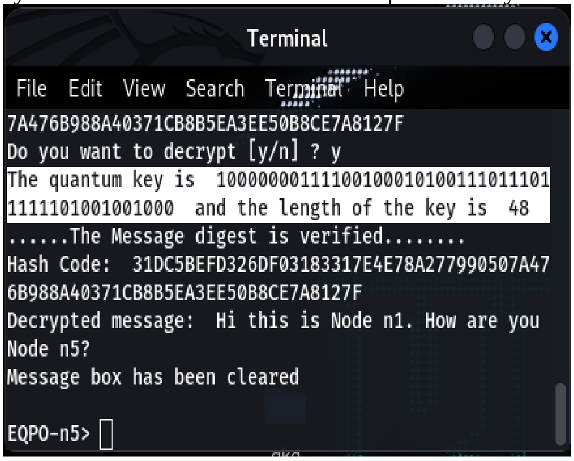
Figure:4.15

The contents of the message were revealed when the receiver typed the command "show messages," which included the sender's ID as well as the message that was sent by the sender. Figure 4.16 depicts the information that was received in the message. To begin, the recipient extracted the message that it had received from node n1 and divided it into two pieces. These parts were an encrypted message and a message digest. The recipient of an encrypted message must possess the quantum key in order to decipher the message. Node n5 computed the quantum key by utilizing the results of the measurements and the bases of n1 and n5. As can be seen in Figure 4.17, the calculated quantum key had the same length as the sender's quantum key and was identical to the sender's quantum key.



```
EQPO-n5> New message arrived. Issue 'show_messages'
to see the contents
show_messages

There are 1 message(s) in the message-box

Sender     Message
------------------
n1         Cx3qbSRMuxqRIR9Ym+wTWNrVmW+J88XNWeoqk4OFo
bBHRBrIfIciLA== 31DC5BEFD326DF03183317E4E78A27799050
7A476B988A40371CB8B5EA3EE50B8CE7A8127F
Do you want to decrypt [y/n] ?
```

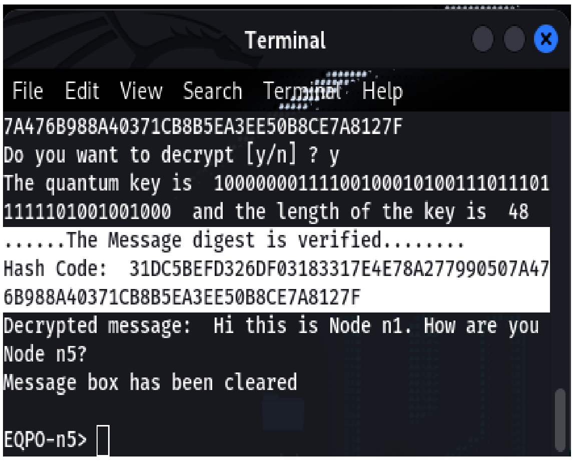
Figure:4.16



```
7A476B988A40371CB8B5EA3EE50B8CE7A8127F
Do you want to decrypt [y/n] ? y
The quantum key is  100000001111001000101001110111101
1111101001001000  and the length of the key is  48
......The Message digest is verified........
Hash Code:  31DC5BEFD326DF03183317E4E78A277990507A47
6B988A40371CB8B5EA3EE50B8CE7A8127F
Decrypted message:  Hi this is Node n1. How are you
Node n5?
Message box has been cleared

EQPO-n5>
```

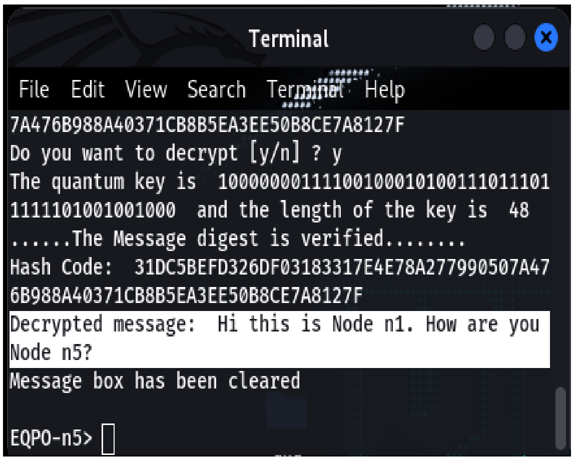
Figure:4.17

The message was decrypted by the receiver using this quantum key, and the hash value was obtained by using the same hash function that was used by the sender node, which is known as the quantum hash function. The determined hash value of the recipient is shown in figure 4.18. The receiver is able to draw the conclusion that there were no eavesdroppers present in the network if the hash value that was supplied by the sender node n1 is identical to the hash value that was computed by the recipient. Figure 4.19 displays the message after it has been decrypted.



```
7A476B988A40371CB8B5EA3EE50B8CE7A8127F
Do you want to decrypt [y/n] ? y
The quantum key is  10000000111100100010100111011101
1111101001001000  and the length of the key is  48
......The Message digest is verified........
Hash Code:  31DC5BEFD326DF03183317E4E78A277990507A47
6B988A40371CB8B5EA3EE50B8CE7A8127F
Decrypted message:  Hi this is Node n1. How are you
Node n5?
Message box has been cleared

EQPO-n5>
```

Figure:4.18



```
7A476B988A40371CB8B5EA3EE50B8CE7A8127F
Do you want to decrypt [y/n] ? y
The quantum key is  10000000111100100010100111011101
1111101001001000  and the length of the key is  48
......The Message digest is verified........
Hash Code:  31DC5BEFD326DF03183317E4E78A277990507A47
6B988A40371CB8B5EA3EE50B8CE7A8127F
Decrypted message:  Hi this is Node n1. How are you
Node n5?
Message box has been cleared

EQPO-n5>
```

Figure:4.19

V. CONCLUSION

In this project, we have created a network with 10 nodes connected as per the Figure 4.1. To solve the multi-objective routing problem in wireless networks, we used the EQPO routing algorithm. To secure the communication between the sender and the receiver, we performed QKD to generate the quantum key, which consists of qubits and bases transmitted through the network using optimal routes produced by the EQPO routing algorithm. We encrypted the plaintext into ciphertext using the Blowfish algorithm with the quantum key. To strengthen the communication, we calculated the hash code using the quantum hash function for the original message and appended it to the ciphertext, which was then transmitted to the receiver in the network. On the receiver's side, the ciphertext was decrypted into plaintext, and the hash code for the plaintext was calculated and compared to the hash value sent by the sender. We noticed that, if eve tries to measure the quantum key being transmitted in the network, it will disturb the quantum state of the qubits, and this disturbance will be detectable by the sender and receiver. Then sender and receiver restart the communication.

Since quantum cryptography has the potential to change information security by providing encryption that can't be broken and secure protocols for communication, its future scope will be safer and stronger against attacks. Also, its future score includes large-scale implementations of quantum key distribution, quantum-resistant cryptography, quantum blockchain, quantum random number generators, quantum sensors, and quantum metrology.

REFERENCES

[1] Adu-Kyere, A.; Nigussie, E.; Isoaho, J. Quantum Key Distribution: Modeling and Simulation through BB84 Protocol Using Python3. Sensors 2022, 22, 6284. https://doi.org/ 10.3390/s22166284.

[2] Shafiqul Abidin, Amit Swami, Edwin Ramirez-Asís, Joseph Alvarado-Tolentino, Rajesh Kumar Maurya, Naziya Hussain,Quantum cryptography technique: A way to improve security challenges in mobile cloud computing (MCC),Materials Today: Proceedings,Volume 51, Part 2022,Pages 508-514,ISSN 2214-7853,https://doi.org/10.1016/j.matpr.2021.05.593.

[3] Panhwar, Muhammad & Ali, Sijjad & Mazhar, Tehseen & Zhongliang, Deng & Qadir, Nabeel & Panhwar, M & Khuhro, S & Mazhar, T & Qadir, Nya. (2021). Quantum Cryptography: A way of Improving Security of Information. International Journal of Mathematics and Computer Science. 16. 9-21.

[4] Chakrabarti, R., Wu, R., & Rabitz, H. (2008). Quantum Pareto optimal control. Physical Review A, 78(3), 033414.

[5] Y. Wang, Y. Chen, H. Ahmad and Z. Wei, "Message authentication with a new quantum hash function," Computers, Materials & Continua, vol. 59, no.2, pp. 635–648, 2019.

[6] Zhou, Qing, and Songfeng Lu. "Hash function based on controlled alternate quantum walks with memory." arXiv preprint arXiv:2105.14788 (2021).

[7] Yang, YG., Xu, P., Yang, R. et al. Quantum Hash function and its application to privacy amplification in quantum key distribution, pseudo-random number generation and image encryption. Sci Rep 6, 19788 (2016). https://doi.org/10.1038/srep19788.

[8] Yang, Yuxiang, Giulio Chiribella, and Masahito Hayashi. "Communication cost of quantum processes." IEEE Journal on Selected Areas in Information Theory 1.2 (2020): 387-400.

[9] Dreher, Patrick & Ramasami, Madhuvanti. (2019). Prototype Container-Based Platform for Extreme Quantum Computing Algorithm Development. 1-7. 10.1109/HPEC.2019.8916430.

[10] Chris Erven (2007). On Free Space Quantum Key Distribution and its Implementation with a Polarization-Entangled Parametric Down Conversion Source. UWSpace. http://hdl.handle.net/10012/3021.

[11] Manju Suresh, M. Neema, "Hardware Implementation of Blowfish Algorithm for the Secure Data Transmission in Internet of Things" Procedia Technology, Volume 25, 2016, Pages 248-255, ISSN 2212-0173, https://doi.org/10.1016/j.protcy.2016.08.104.

[12] Alanis, Dimitrios, et al. "A quantum-search-aided dynamic programming framework for pareto optimal routing in wireless multihop networks." IEEE Transactions on Communications 66.8 (2018): 3485-3500.

[13] Zhang, Yong-Sheng, Chuan-Feng Li, and Guang-Can Guo. "Quantum key distribution via quantum encryption." Physical Review A 64.2 (2001): 024302.

[14] Bhandari, Ramesh. "Quantum error correcting codes and the security proof of the bb84 protocol." arXiv preprint arXiv:1409.1452 (2014).

[15] Cherry Mangla, Shalli Rani, Nawab Muhammad Faseeh Qureshi, Aman Singh,Mitigating 5G security challenges for next-gen industry using quantum computing,Journal of King Saud University -Computer and Information Sciences,2022,ISSN 1319-1578,https://doi.org/10.1016/j.jksuci.2022.07.009.

[16] Alabaichi, Ashwaq & Ahmad, Faudziah & Mahmod, Ramlan. (2013). Security analysis of blowfish algorithm. 12-18. 10.1109/ICoIA.2013.6650222.

[17] Yang, Yuxiang, Giulio Chiribella, and Masahito Hayashi. "Communication cost of quantum processes." IEEE Journal on Selected Areas in Information Theory 1.2 (2020): 387-400.

[18] Sahoo, J & Satapathy, Sandeep. (2011). SIMULATION AND ANALYSIS OF BB84 PROTOCOL BY MODEL CHECKING. International Journal of Engineering Science and Technology. 3.

[19] Shor, Peter W. and Preskill, John,"Simple Proof of Security of the BB84 Quantum Key Distribution Protocol",Phys. Rev. Lett.,85-2,441-444,2000,doi:10.1103/PhysRevLett.85.441.

[20] Zhou, Q., Lu, S. One-dimensional quantum walks with two-step memory. Quantum Inf Process 18, 359 (2019). https://doi.org/10.1007/s11128-019-2475-3.

[21] S. Chaturya, Y. Adilakshmi, MBDS: Message Authentication Code (MAC) Based Black Hole Detection System in Manets, International Journal of Innovative Technology and Research, ISSN: 2320-5547, pp. 7980-7987, Feb-Mar 2018, indexed by Google Scholar.

[22] Yannam, Adilakshmi & Prasad, Dr. (2019). "Cooperative Intrusion Detection System to Enhance the Security in MANET." Journal of Advanced Research in Dynamical and Control Systems. 11. 100-109.