Republic of Iraq

Ministry of Higher Education and Scientific Research

University of Babylon

College of Information Technology

Department of Information Networks

ENABLING NFV QUANTUM KEY DISTRIBUTION FOR NETWORK SECURITY

A Thesis

Submitted to the Council of the College of Information Technology for Postgraduate Studies of University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology-Information Networks

BY

Maryam Saad Jaafar Radhi

Supervised by

Asst. Prof. Dr.

Alharith A. Abdullah

Asst. Prof.

Ali Kadhim Bermani

2023 A.D. **1445 A.H.**

بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

(وَما تَوفيقي إِلّا بِاللَّـهِ عَلَيهِ تَوَكَّلتُ وَإِلَيهِ أُنيبُ)

صدق الله العلي العظيم

سورة العنكبوت، آية:19

Supervisor Certification

I certify that the thesis entitled (**Enabling NFV Quantum key distribution for network security**) was prepared under my supervision at the department of Information Networks/ College of Information Technology / University of Babylon as partial fulfillment of the requirements of the degree of Master in Information Technology-Information Networks.

Signature:

Supervisor Name: Asst. Prof.

Dr. Alharith A. Abdullah

Date: / /2023

Signature:

Asst. Prof.

Ali Kadhim Bermani

Date: / /2023

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "**Enabling NFV Quantum key distribution for network security**" for debate by the examination committee.

Signature:

Asst. Prof. Dr. Alharith A. Abdullah

Head of Information Networks Department

Date: / /2023

Certification of the Examination Committee

We hereby certify that we have studied the thesis entitled (**Enabling NFV Quantum key distribution for network security**) presented by the student (**Maryam Saad Jaafar Radhi**) and examined her in its content and what is related to it, and that, in our opinion, it is adequate with (**very good**) standing as a thesis for the degree of Master in Information Technology-Information Networks.

Signature:
Name: Dr. Mahdi Ebady Manaa
Title: Asst. Prof.
Date: / / 2023
(Chairman)

Signature:
Name: Dr.Mohammad Hussein Jawwad
Title: Asst. Prof.
Date: / / 2023
(Member)

Signature:
Name: Dr. Hayder A. Nahi
Title: Lec.
Date: / / 2023
(Member)

Signature:
Name: Dr. Alharith A. Abdullah
Title: Asst. Prof.
Date: / / 2023
(Member) // MainSupervisor

Signature:
Name: Ali Kadhim Bermani
Title: Asst. Prof.
Date: / / 2023
(Member) // Supervisor

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: Dr. Wesam S. Bhaya
Title: Prof.
Date: / / 2023
(Dean of Collage of Information Technology)

Dedication

I dedicate this thesis

To the **martyrs** of my country, especially the martyrs of the October Revolution

To **my family** and **my husband** who were the best support and encouragement

And to **everyone** who encouraged and helped me complete this work

Acknowledgments

I want to thank my supervisors (Asst. Prof. Dr. Alharith A. Abdullah and Asst. Prof. Ali Kadhim Bermani) for their good cooperation, as they provided me with what I needed from sources and inquiries and were present step by step in order to complete this study.

Declaration

I hereby declare that this thesis, submitted to the University of Babylon in partial fulfillment of requirement for the degree of Master of Information Technology-Information Networks has not been submitted as an exercise for a similar degree at any other university. I also certify that this work described here is entirely my own except for experts and summaries whose sources are appropriately cited in the references.

Signature:

Name: Maryam Saad Jaafar Radhi

Date: / / 2023

Abstract

With the ever-growing concern for internet security, the field of quantum cryptography emerges as a promising solution for enhancing the security of networking systems. NFV derives from Server Virtualization, a technique that enables users to share the resources and functionalities of a server concurrently through Virtual Machines (VMs) emulating a standard server. In a similar way, NFV migrates the standard network functions to a virtual or cloud-based architecture, leading to high flexibility.

Internet networks are considered an essential element in daily life due to the development taking place in the field of technology and its entry into most aspects of life, including education, health, industrial, and commercial. As a result of this development, and with the increase in the number of customers, services, and network functions, there was a load on the network and the devices that perform network functions and an obstruction to services. As a result of the development taking place and the increase in the number of customers, there is an urgent need to innovate new technologies to accommodate this number without the need to purchase new servers in real-time and without delay in installation. Also, the very expensive hardware required for quantum key distribution.

This work proposes implementing the process of distributing quantum and classical keys as a virtual network function using NFV technology, which includes the creation of keys and distributing them among customers using a Cloud Computer and Virtual Machine in VMware Workstation to achieve NFV technology.

The proposed system shows a low QBER in the case of BB84. The average QBER in the virtual environment was 0.45, while the average QBER for the same number of experiments in the QKD emulator was 0.48. Reduce time to distribute keys when implementing NFV in addition to reducing the cost of purchasing hardware for

these services and increasing implementation by speeding up the process of distributing keys.

List of Publication

No.	Paper name	Doi
1	Virtualizing Quantum Key Distribution Based on Network Function Virtualization	

List of tables

List of Figures

List of Abbreviations

Authorization, and Accounting	AAA
Application Programming Interfaces	APIs
Amazon Web Service	AWS
Classical Key Distribution	CKD
Charles Bennett and Gilles Brassard in 1984	BB84
Central Processing Unit	CPU
Data Encryption Standard	DES
Domain Name System	DNS
Elastic Compute Cloud	EC2
Diffie-Hellman Key Exchange	DHKE
Group Key Management Scheme for Low-Resource Devices	GKM-LRD
Google Cloud Platform	GCP
Hardware Virtual Machines	HVM
Internet of Things	IoT
Intrusion Detection Systems	IDS
Internet Protocol	IP
Integrated Development Environment	IDE
Infrastructure as a Service	IaaS

Key Managements	KM
Long Term Evolution / Evolved Packet Core	EPC/LTE
Mobile Edge Computing	MEC
Management and Orchestration	MANO
Network Intrusion Detection Systems	NIDS
Network Function Virtualization	NFV
Network Function Virtualization Infrastructure	NFVI
Network service providers	NSPs
Network Address Translation	NAT
Proof-of-Work	PoW
Platform as a Service	PaaS
Paravirtual	PV
Physical network function	PNF
Proof-of-Stake	PoS
Quantum Key Distribution	QKD
Quality of Service	QoS
Quantum Bit Error Rate	QBER
Quantum Bits	Qubits
Quantum channel	QC

Classical channel	CQ
Remote Desktop Protocol	RDP
Ron Rivest Cipher	RC2
Rivest Cipher 4	RC4
Rivest Cipher 5	RC5
Rivest Cipher 6	RC6
Rivest, Shamir, and Adelman	RSA
Random Access Memory	RAM
Software-Defined Network	SDN
Service Level Agreements	SLAs
Software as a Service	SaaS
Secure Shell	SSH
Unmanned Aerial Vehicles	UAVs
Virtual Network Function	VNF
Virtual Machine Monitor	VMM
Virtual Machine Software	VMware
virtual machine image	VMI
Virtual Network Security Functions	VSFs
Virtual Distributed Ledger Technology	vDLT

Virtual Software-Defined Network	vSDN
virtualized infrastructure managers	VIM

Table of Contents

Chapter one
General introduction

1.1 Introduction

In today's interconnected world, the security of network communication has become a critical concern for individuals, businesses, and governments. With the increasing number of cyber-attacks and data breaches, there is a growing need for robust security measures that can protect network communication from unauthorized access and interception. To address this challenge, researchers and practitioners have explored various approaches to network security, including Network Function Virtualization (NFV), key management, Diffie-Hellman, quantum computing, and BB84 additionally to the thesis goal [1].

The primary goal of network functions virtualization (NFV) is to virtualize traditionally hardware-implemented network functions, such as routers, firewalls, load balancers, and proxies. Servers and other hardware resources, such as computing, storage, and networking hardware, can be used to implement such virtual network functions. Depending on the task assigned to the network, they may move to various nodes or be instantiated in multiple locations. Functions in Software can run on standard servers in the industry, so there's no need to set out specialized devices.

Cloud computing, in its broadest sense, refers to the provision of services via a distributed network of shared computers running Software and other applications. Cloud computing is essentially an architectural philosophy and design for computing systems. It's a lot simpler in some ways. The main concept is to separate the hardware, Software, and operating system. Instead of having to completely shut down the system in the event of a bug or virus, virtualization allows for the program to be automatically moved to another server [2].

One promising approach is the use of quantum key distribution to secure network communication. Quantum key distribution leverages the principles of

quantum mechanics to generate cryptographic keys that are impossible to intercept or eavesdrop on. These keys can be used to encrypt and decrypt data, ensuring the confidentiality and integrity of network communication [3].

The proposed approach has the potential to significantly enhance network security and scalability by providing a practical and efficient way to enhance network communication. Overall, Enabling NFV Quantum Key Distribution and classical key distribution for Network Security projects represents an important step towards improving the security and Real-time response optimization to network variables of network communication in today's increasingly interconnected world [4].

To enable the use of quantum key distribution for network security, the Enabling NFV Quantum Key Distribution for Network Security thesis proposes a comprehensive approach that combines the power of NFV, key management, and quantum computing. The thesis aims to develop a prototype system that can generate secure cryptographic keys and deploy them using NFV to provide secure communication between users in the same server who want to communicate through the network[5]. NFV is a technology that permits virtualization of network functions, which can be run on standard hardware. This feature enables network operators to quickly and efficiently deploy network services, making it more popular in the telecommunications industry. NFV can be utilized in the context of network security to deploy security functions such as intrusion prevention systems, intrusion detection systems, and firewalls [6].

Furthermore, Key management is the process of generating, storing, and distributing cryptographic keys, which are necessary for secure communication by encrypting and decrypting data. Key management is a vital aspect of network

security, but it becomes more complicated as the number of devices and services on the network increases. Diffie-Helman key exchange is a well-known cryptographic algorithm that allows two parties to securely exchange secret keys over a public channel. This algorithm was first introduced by Whitfield Diffie and Martin Hellman in 1976 and has since become widely used in various applications, including network security. In the context of enabling NFV quantum key distribution for network security, the Diffie-Hellman key exchange can also be used as a virtual network function as classical key distribution. However, traditional cryptographic algorithms like Diffie-Hellman are vulnerable to attacks from quantum computers, which can solve certain mathematical problems much faster than classical computers [7]. This is where quantum key distribution (QKD) comes into play. QKD is a quantum-based cryptographic technique that enables the secure distribution of cryptographic keys between two parties. It is based on the principles of quantum mechanics, which makes it theoretically unbreakable. Therefore, integrating QKD with NFV and other traditional cryptographic algorithms can enhance network security and provide a more robust defence against cyber-attacks.

Quantum computing is a new field that explores quantum mechanics to develop innovative computing methods. Quantum computers can solve specific problems related to cryptography faster than classical computers. Quantum key distribution is a technique that employs quantum mechanics principles to create cryptographic keys that are impossible to intercept or eavesdrop on. Enabling NFV Quantum Key Distribution for Network Security is a project that aims to improve the security of network communication by leveraging the power of three different areas of expertise: Network Function Virtualization (NFV), key management, and

quantum computing. The thesis intends to keep pace with the continuous increase in the number of customers in the network [8].

1.2 Related work

This section aims to shed light on previous studies that dealt with transforming network functions into NFV.

The work in [9] aims to improve the security of virtual network function (VNF) distribution across data centres by integrating quantum key distribution (QKD) with network function virtualization (NFV). QKD generates cryptographic keys that are more secure than public key encryption methods, while NFV replaces traditional network functionalities with software instances to reduce costs. However, the security of NFV needs to be addressed prior to deployment in the real world. The paper combines NFV orchestration and QKD technology through SDN-controlled optical networks to achieve a secure architectural solution for VNF distribution. The paper presents a time-shared approach as a cost-effective solution for practical deployment and shows that the solution can be applied in longer distances by using a multi-hop approach to distribute keys on the network. Results demonstrate that the proposed solution can secure an SSMF link of up to 25 km using quantum encryption for NFV MANO operations with a simultaneous SDN control, showing a minimum QBER of 5.3%. The integration of QKD and NFV in this manner represents a significant advancement in enhancing the security of virtual networks.

The work in [10] discusses the benefits of SDN and NFV in achieving flexible programmability of network control functions and protocols with dynamic usage of network resources. SDN provides network resource abstraction over APIs

to achieve topology-independent multiple-tenant networks, while NFV deploys network functions as VNFs on commodity hardware using virtualization techniques. The paper proposes a traffic load balancing mechanism using a virtual SDN (vSDN) controller as a VNF in SDN-enabled networks. When the traffic load exceeds a certain threshold, a secondary vSDN controller is added to share the load and tasks of the original vSDN controller. The system is experimentally validated using a Fat-Tree topology with OpenDaylight as the SDN controller on a Mininet emulator. The results show significant improvements in network performance, including a 50% improvement in average load and a 41% improvement in average delay. Future work aims to deploy more IP network functionalities as VNF services and focus on the virtualization of EPC/LTE network control functions.

The work in [11], a service-oriented blockchain system called vDLT is proposed as a solution to performance issues in existing Distributed ledger technology (DLT) systems. Most existing DLT systems do not distinguish between different quality of service (QoS) requirements, which results in significant performance issues. The proposed vDLT system classifies services and applications into different classes based on their QoS requirements. Advanced schemes such as classification, queuing, virtualization, resource allocation and orchestration, and hierarchical architecture are used to fulfil different QoS requirements. The system features a decoupling of management/control and execution of smart contracts, which supports QoS provisioning, improves decentralization, and facilitates evolution in vDLT. The authors argue that this represents a paradigm shift from existing “blockchain-oriented” DLT systems to next-generation “service-oriented” DLT systems. The authors draw inspiration from the development of telephone networks, the traditional internet, and cellular networks, which faced similar issues in their early stages. Overall, this research

proposes vDLT as a service-oriented blockchain system that addresses QoS requirements and provides a more efficient and scalable solution to existing DLT systems.

This work in [12] proposes a framework that employs Software Defined Networks (SDN) and Network Function Virtualization (NFV) to deploy virtual network security functions (VSFs) in Mobile Edge Computing (MEC) nodes, specifically on Unmanned Aerial Vehicles (UAVs). The framework enables the automatic orchestration, configuration, and deployment of lightweight VSFs in MEC-UAVs, taking into account contextual factors such as physical and virtual conditions, to optimize security orchestration. The proposed solution's goal is to provide on-demand VSFs, including virtual Firewalls, Proxies, Intrusion Detection Systems (IDS), and Authentication, Authorization, and Accounting (AAA) services to strengthen end-to-end security in IoT environments. The paper presents an algorithm that selects the most suitable UAV-MEC without human intervention based on factors such as operating capacity, battery, atmospheric conditions, computing resources (RAM, disk, CPU), and network metrics. The framework has been implemented and deployed using open-source tools, and its effectiveness has been verified in a real testbed and use case. The tests demonstrate that the solution is capable of handling workloads successfully, even in worst-case scenarios. Future work includes enhancing the proposed algorithm to accommodate more complex scenarios. The proposed framework contributes to related work on using SDN and NFV to deploy security functions in MEC environments and demonstrates its effectiveness in UAVs.

The work in [13] explores the issue of network intrusion detection systems (NIDS) becoming a bottleneck when network traffic is heavy. Despite the numerous solutions that have been proposed, they are not without limitations. To

address this issue, the authors propose a lightweight elastic architecture that enables parallel processing within the existing NIDS while preserving filtering integrity. They also suggest two adaptive algorithms that dynamically adjust and evenly distribute signature rules across NIDS nodes to achieve intelligent rule ordering. The proposed system was tested in real-life settings by creating a working prototype that utilized modern networking technologies. The results indicate that the proposed algorithms can split the IDS workload equally, enabling the system to scale by adjusting the number of virtual components that analyze network traffic. The proposed system also ensures that adequate system resources are allocated when needed to prevent the IDS from dropping packets during high workloads. The authors recommend future enhancements, such as incorporating the percentage of dropped packets as a factor in the scaling algorithm. The proposed system is flexible and can be implemented in various infrastructures that utilize software-defined networking, virtual machines, and containers.

In [2], A novel blockchain architecture based on network functions virtualization (NFV) with auto smart contracts," is a proposed research paper that suggests combining the benefits of blockchain technology and NFV to create a more efficient and automated system. The paper proposes a new blockchain architecture that uses NFV to create a virtualized environment that allows for the deployment of smart contracts. The smart contracts are designed to automate the execution of certain network functions, such as billing, routing, and security, which can help reduce costs and improve network efficiency. The paper also proposes the use of a consensus mechanism based on proof-of-stake (PoS) to validate transactions on the blockchain. The PoS consensus mechanism is designed to be more energy-efficient than the traditional proof-of-work (PoW) mechanism used

by many existing blockchain systems. Generally, the proposed architecture aims to address some of the limitations of existing blockchain systems, such as scalability, energy consumption, and high transaction costs, by leveraging NFV and smart contracts.

Table1.1: Summary of related work

Author	Year	Challenges	Methodology	Result
P. Sibson et al.	2016	network functions in NFV network are stored centrally as software images in a remote data center (DC) where they can be cloned, transferred and deployed as virtual functions on commodity servers (replacing network appliances) across the network. This transfer of network functions must be secured.as an eavesdropper could compromise not just virtualized services, but the whole infrastructure.	demonstrate, for the first time, a secure architectural solution for VNF distribution, combining NFV orchestration and QKD technology by scheduling an optical network using SDN. -and enhanced security for longer distances by adding trusted node functionality.	-enhanced security capabilities -Results demonstrate that an SSMF link of up to 25 km can be secured using quantum encryption for NFV MANO operations with simultaneous SDN control, showing a minimum QBER of 5.3%.
Sikandar E. et al.	2018	- Hardware components with proprietary characteristics, high costs, and a shortage of skilled professionals make it difficult to bring and integrate new services to suit user needs.	propose traffic load balancing using an SDN controller as a virtualized network function (creation of vSDN) and adding a secondary vSDN controller to distribute the workload to them to avoid overloading any resource when the load is increased.	Results showed : -50% improvement in Average Load. -41% improvement in Average Delay. -considerable improvements in terms of Ping Response, Bandwidth Utilization and Throughput of the system.
Zenan W. et al.	2018	-It is difficult to determine the order and integration of network services to create service chains, especially with the huge growth in network services -Without NFV, hardware-based	presents a typical framework to construct service chains by combining SDN and NFV Without traditional manual	-dynamic and cost-efficient traffic steering between VNFs -network resources could be more efficiently utilized by SFCs. -resources such as CPU memory

		middleboxes are difficult to be scaled or updated even if there is a need. Without SDN, the traffic forwarding between NFs can only be manually configured by network operators on forwarding devices such as switches and routers.	configuration.	taken by a VNF node could be elastically adjusted as the service requirements change.
Yu, F. R. et al.	2019	Most existing DLT systems do not distinguish between different quality of service (QoS) requirements, which results in significant performance issues	classifies services and applications into different classes based on their QoS requirements	supports QoS provisioning, improves decentralization, and facilitates evolution in vDLT provides a more efficient and scalable solution to existing DLT systems
Jorge B. et al.	2020	-security risks have increased due to the expansion of IOT networks. -The difficulty of replacing and dealing with remote physical locations of the IOT network puts them at risk in traditional cases.	proposes an SDN/NFV-based security format and framework for NFV deployment in MEC-based UAVs to utilize virtual network security functions such as virtual firewalls (vFirewalls) and vIDS for specific locations on demand in IoT networks.	–the UAV flying time falls almost half when apply the payload. -increased response speed in replacing and providing network functions.
Chenxi L. et al.	2021	-Performing DPI in the traditional way is difficult for packets with low packet loss rates. -Traditional high performance IDS usually have a vast dependence on clustered commodity servers at fixed locations or dedicated high-performance hardware, which means they have limited scalability and poor flexibility.	propose a novel workload scheduling framework for IDS deployment in the NFV scenario.	–better detection performance –lower memory usage –better scalability for resource allocation in the NFV scenario
Jawdhari, H. A et al.	2021	Limitations of existing blockchain systems, such as scalability, energy consumption, and high transaction	-A novel blockchain architecture based on network functions virtualization (NFV) with	-create a more efficient and automated system -reduce costs -improve network efficiency

		costs, by leveraging NFV and smart contracts.	auto smart contracts - blockchain architecture that uses NFV to create a virtualized environment that allows for the deployment of smart contracts	

1.3 Problem statement

The thesis is focused on resolving the problem of ensuring secure network communication in light of the escalating quantity of network devices and services. With the growing number of network devices and services, the risk of attacks and eavesdropping also increases. Traditional network security methods may not provide adequate solutions to address these threats, necessitating novel approaches to be developed.

The lack of specialized devices to accommodate the large increase in the number of users and the lack of real-time fault tolerance leads to delays in service.

The challenge of making the transition from traditional network infrastructure to NFV-based solutions is another major challenge for network carriers.

The thesis recognizes that the combination of NFV and key management (quantum key distribution, Diffie-Hellman) has the potential to create a comprehensive and robust network security system. However, there are challenges associated with integrating these areas of expertise. For example, quantum key distribution requires specialized hardware that may not be readily available or cost-

effective for all network operators. Key management is also becoming more complicated as the number of devices and services on the network increases.

The thesis's problem statement is to overcome these challenges and create a secure network that is resistant to attacks and eavesdropping.

1.4 Thesis Objectives

The Enabling NFV Quantum Key Distribution and Diffie-Hellman for Network Security project has the following objectives:

1. To investigate the state-of-the-art in NFV, key management, and quantum computing and their applications to network security.
2. To develop a comprehensive approach to network security that leverages the power of NFV, key management, and quantum computing.
3. To design and implement a prototype system that demonstrates the feasibility of the proposed approach to network security.
4. To evaluate the performance of the prototype system and compare it with existing network security systems.
5. To identify the benefits and limitations of the proposed approach and provide insights into potential areas for future research.
6. Get the virtual functions such as quantum key distribution and diffie hellman on cloud services.

1.5 Thesis Outline

The remaining chapters of this thesis are as follows:

Chapter Two: overview, the introduction of Network Function Virtualization (NFV), NFV architecture, Traditional Network Service, Reasons for Using NFV, Cloud computing, Key management, Diffie-Hellman, Quantum computing, Quantum key distribution, BB84, and evaluation are discussed in this chapter.

Chapter Three: In this chapter, the suggested system is presented, along with examples of the system's actual steps and an explanation of the proposed approach.

Chapter Four: This chapter discusses the findings and assesses the suggested strategy.

Chapter Five: Shows the main conclusions of this work, and it also offers suggestions for future works.

Chapter Two
Theoretical Background

2.1 Overview

In this chapter, the main topic of the thesis, which is NFV, has been clarified in addition to the traditional network services to know the difference between NFV and the functions provided traditionally, as well as cloud computing because it's one of the methods used in applying the NFV architecture, key management and its types, Cloud service providers, quantum computing and Diffie-Hellman, BB84, and finally the Evaluation metrics.

2.2 Network Function Virtualization (NFV)

The means for virtual networking services to be provisioned automatically and network applications to be developed. By using these approaches, we can improve upon the already impressive resiliency and adaptability of conventional networks while also overcoming their limitations [14]. If you take a closer look at today's network environments, you'll notice that there are a lot of different proprietary hardware devices. The launch of a brand-new network service necessitates the installation of new hardware entities due to constraints on physical space, making the process significantly more difficult as its complexity grows.

By separating the hardware of networks from the services or all other functions which work within them, Network Functions Virtualization (NFV) is a new approach to many operations in the network, including planning and deployment, in addition to supervisory control network services. The primary objective of NFV is to swap out hardware-centric gadgets for Software on generic servers. [2]

2.2.1 Network Function Virtualization (NFV) architecture

Virtualized Network Function (VNF), Network Function Virtualization Infrastructure (NFVI), and Network Function Virtualization Management and

Orchestration (NFV-MANO) are the three parts that form up the NFV framework, as shown in Figure 2.1[15].

1. Management and orchestration (MANO) manage the virtual network function (VNF) via its VNF manager and the entire NFVI lifecycle via the virtualized infrastructure managers (VIM).
2. Network functions virtualization infrastructure (NFVI) includes all the necessary hardware and Software to deploy virtual network functions (VNFs). Additionally, NFVI can be used to classify the various points of presence (infrastructure) and their interconnections.
3. Virtualized network functions (VNFs) The virtualized network functions (VNFs) are the Software (network function) designed to run on the NFVI [16].

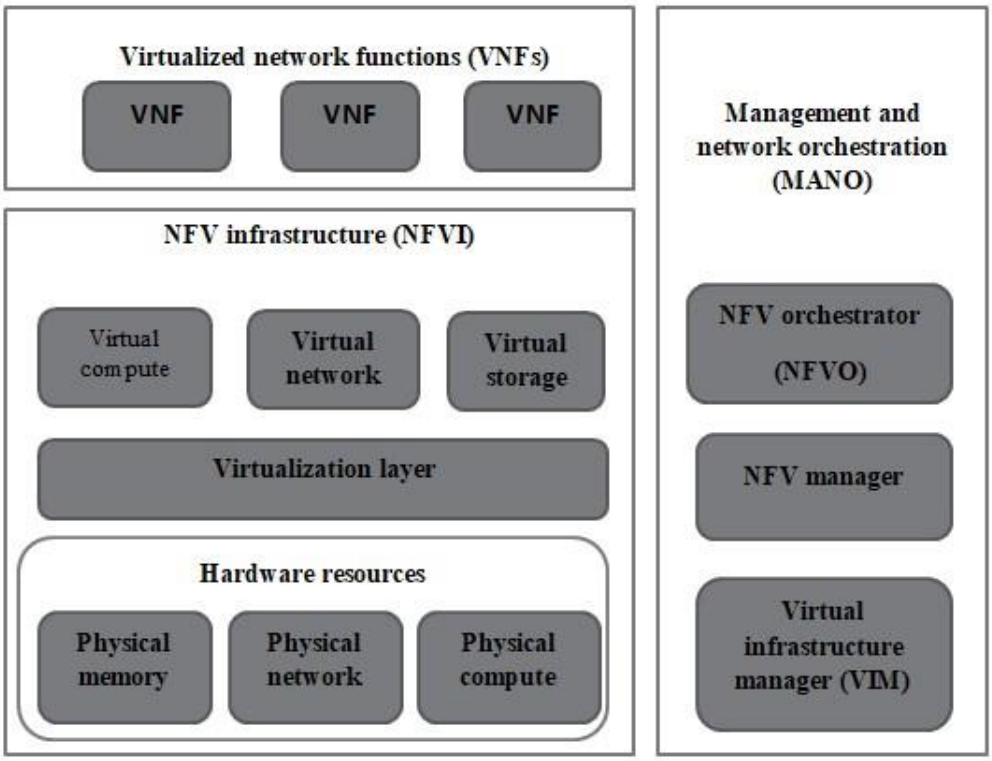

Figure 2.1: Network Function Virtualization Architecture [17].

2.3 Traditional Network Service

Network service providers (NSPs) do more than connect their business clients to the internet. They also provide supplementary network functions such as NAT, encryption, firewall, caching, DNS, and so on. These internet functions have already been implemented at the customer location via specialized hardware. This strategy requires more income and more resources to implement (truck rolls are needed whenever a new network function is added to a service, power, or space in order to install the dedicated new hardware device), as shown in figure 2.2. As a result, businesses in the sector began investigating how Network Function Virtualization (NFV) could help them cut costs and speed up deployment [14].

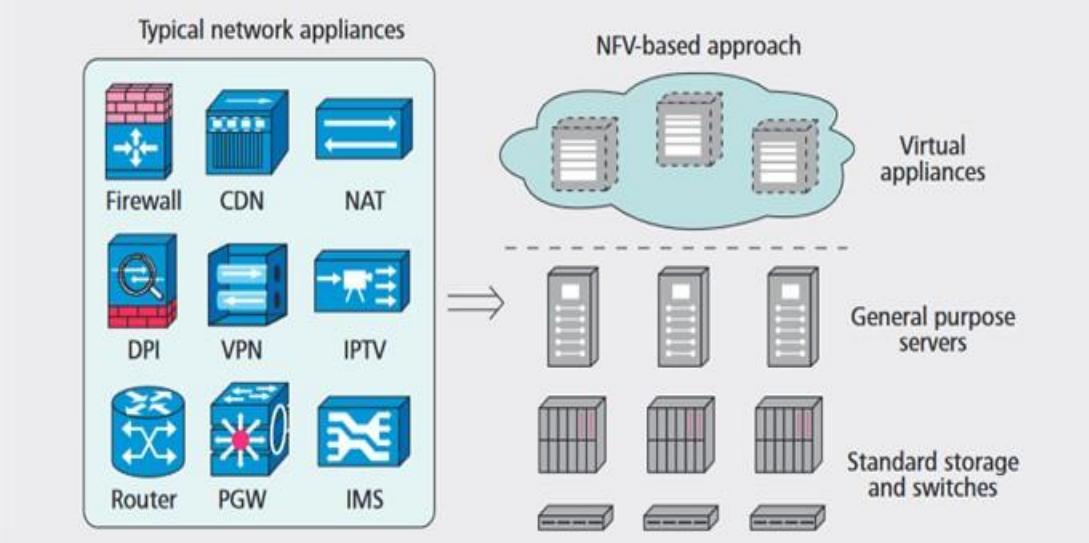

Figure 2.2: Converted From Dedicated Hardware of Network Function to Network Function Virtualization [18].

2.4 Reasons for Using NFV

Network operators can get many advantages thanks to NFV's efforts, including [2]:

- **Reduce complexity and energy consumption**: efficiency in power, cooling, and space. In order to make the most efficient use of expensive resources like space, electricity, and cooling in a data centre, network service providers (NSPs) must make careful equipment selections. As a result of resource consolidation, NFV improves energy efficiency.
- **Quicker life cycle**: The rapid, on-demand, and as-needed deployment of new network services has benefited both end users and network service providers.
- **Lower capital costs:** less money is needed to get things up and running when you use open-source Software, make better use of your hardware, and stick to industry standards for service delivery. There is a reduction in both hardware requirements and operating expenses thanks to NFV's use of virtual machines [14]. The following table illustrates the difference between NFV and physical network function.

Table 2.1: the difference between network function virtualization and physical network function [19].

Metrics	**NFV**	**PNF**
Cost	Less	High
Space	Less	High
Power consumption	Less	High
Scalability	High	Less
Fault tolerance	High	Less
Complexity	Less	High

Flexible	High	Less

2.5 Cloud computing

In the cloud, customers and service providers work together to create a shared network of shared, virtualized, and elastic computing resources that can be presented and provisioned on demand in accordance with predetermined Service Level Agreements (SLAs). Cloud computing's benefits include limitless access to computing power at a low price, built-in redundancy and disaster recovery, rapid elasticity, and scalability [20]. The three types of services that make up cloud computing are: Platform as a Service (PaaS), Infrastructure as a service (IaaS), and Software as a Service (SaaS).

i. **Software as a service:** the costumer makes use of the service provider's Software. Thin and thick clients are both provided access to these applications. When using SaaS, the cloud service provider acts as both the application owner and the system administrator. Some of the application's settings are customizable by the cloud user. Email is one type of SaaS application.

ii. **Platform as a service:** an application developer who purchases a provider's platform uses those tools to create an app for end users. PaaS includes both a development and a production setting. The frameworks, libraries, and services of the supported programming languages are all part of the development environment. Providers of PaaS handle the underlying infrastructure, while end users have control over both the apps and environments in which they are used.

iii. **Infrastructure as a service:** The Provider's data centre and related infrastructure are utilized by the consumer. IaaS primarily allows users to provision virtual machines, select the type of image to use, assign volumes, and establish local connections. OpenStack and EC2 are currently the most popular IaaS. The structure of cloud computing systems is identical to that of conventional computing systems (i.e., Software above hardware). The key distinction is that IaaS is made possible by virtualizing a large number of servers and pooling their resources so that they function as if they were a single, powerful server. The Virtual Machine Monitor (VMM) or hypervisor provides the virtualization layer [15].

2.6 Cloud Service Providers

Companies like IBM, Google, Amazon, and Microsoft are in a race to provide the best cloud services to their clients. As the US and UK market is dominated by Amazon and Microsoft, they are said to be the leading tech giants in the world. There is a long list of companies providing cloud services. Nowadays, small tech organizations have also started investing in the cloud domain. Amazon Web Services (AWS) has the highest market share as compared to Microsoft Azure and Google Cloud Platform (GCP) [21].

.

2.6.1 Amazon Web Services (AWS)

It is a collection of cloud computing services, also called services, that make up a cloud-computing platform offered by Amazon.com. These services operate from 12 geo- graphical regions across the world. The most central and well-known of these services arguably include Amazon Elastic Compute Cloud, also known as ”EC2”, and Amazon Simple Storage Service, also known as ”S3”. Amazon

markets AWS as a service to provide large computing capacity more quickly and more cheaply than a client company building an actual physical server farm [22].

2.6.1.1 Amazon Elastic Cloud Computing (EC2)

Amazon EC2 is a component of Amazon's Web Services (AWS), which allows users to rent computers to run computer applications in the Amazon EC2 data centre. Amazon EC2 uses the Xen virtualization technique to manage physical servers. There might be several Xen virtual machines running on one physical server. Each Xen virtual machine is called an instance in Amazon EC2. There are several types of instances. Each type of instance provides a predictable amount of computing capacity. The small instance is the primary instance type, which is configured with 1.7GB memory, 1 EC2 compute unit and 160GB instance storage. According to Amazon, "one EC2 compute unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor." For applications requiring higher computing capacity, Amazon EC2 provides several high-capacity instances, which are configured with 4 to 20 EC2 compute units. The input-output (I/O) capacities of these types of instances are not specified clearly [23].

2.7 Key management

Key distribution is a fundamental issue in any security system. The term "key management" refers to a collection of methods used to establishment, distribute, and update a shared secret for use in encrypted communications. In addition, compromised nodes' keys can be updated or refreshed [20].

2.7.1 Types of Key Managements

The key management methods are divided into symmetric and asymmetric encryption algorithms [21], as shown in Figure 2.3, based on the type of encryption they employ. They are able to accomplish this goal by re-encoding information into file formats that are extremely difficult for

unauthorized parties to break or decrypt. The most basic and widely used type of encryption is symmetric encryption, which includes algorithms like DES, RC2, RC4, Blowfish, RC5, RC6, and AES. An attacker can eavesdrop on the key exchange channel and use it to decrypt the data because the sender and receiver share the key. The transmission of the secret key between the sender and the receiver requires a secure channel. Asymmetric encryption, on the other hand, employs two keys—public and private—to encrypt plaintext. This is the case with DSA and RSA. The security of the system is increased because the private key is kept secret, while the public key can be used by anyone to send an encrypted message. In total, they all require some form of manipulation with plaintext in order to produce cypher text, or vice versa. The following graphic depicts the many cryptographic algorithms available [22].

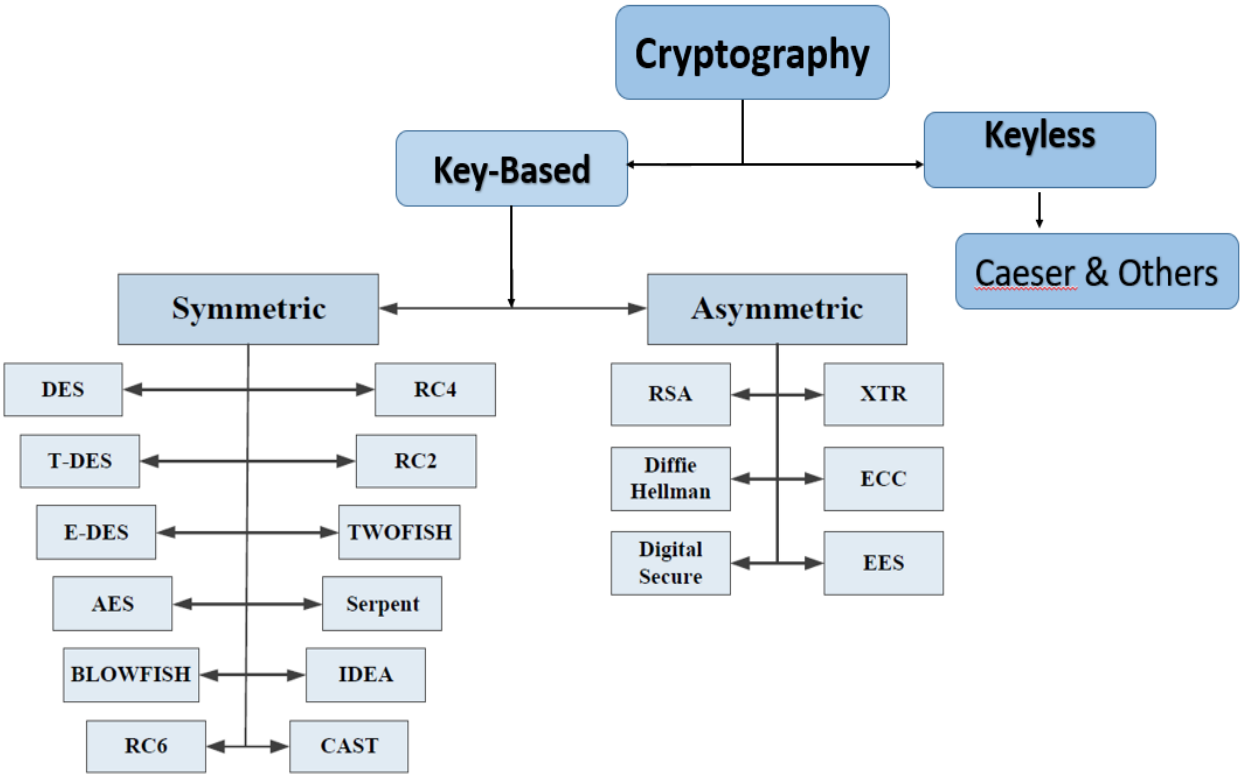

Figure 2.3: categories of cryptography [22].

2.7.1.1 Diffie-Hellman key exchange (DHKE)

The type of asymmetric encryption that is used for establishing shared secrets over an open channel is the basis for many cryptographic algorithms. For instance, a client can use DHKE to negotiate a secret key with a web server for encrypted communication, making for a more secure web browsing experience. When DHKE is used, not even a (passive) eavesdropper can figure out what the secret keys are [23]. The fact that encryption keys are never sent over the network. Diffie-Hellman presents the following explanation: as an illustration of DHKE, we'll take a look at Alice and Bob. Alice and Bob generate two numbers, p and g, where p is the prime number and g is a generator, which is the primitive root modulo of P. Alice generates a private number, a and Bob also generates his own private number, b, Alice calculates their public number as follows :

$$A = (g)^a \ mod \ p \qquad (1)$$

Also, Bob calculates their public number as follows:

$$B = (g)^b \ mod \ p \qquad (2)$$

Now Alice and Bob exchange their public number via a public channel, and Alice knows p,g, a, A, B and Bob knows p,g,b, B, A. Upon completion of the exchange process, both parties calculate the shared key by utilizing the following equation [24]:

$$key = \ (public)^{private} \ mod \ p \qquad (3)$$

The strength of the Diffie-Hellman is that the intruder knows p,g, A, B and does not know a,b. Therefore, the process of extracting the key is difficult,

especially if the special numbers are large, because it is difficult to extract the discrete logarithm [28].

2.7.1.2 Rivest, Shamir, and Adelman (RSA)

In 1978, Rivest, Shamir, and Adelman released the RSA algorithm to the public for the first time. The RSA is an example of an asymmetric cryptographic algorithm. The factorization difficulty of the product of two large prime numbers is the basis for the algorithm. In the first step of the algorithm, generate public and private keys: [25]

- Select two prime numbers (P, Q)
- Calculate product (N)

$$N = P * Q \qquad (4)$$

- Calculate Euler's Totient function(T),

$$T = (P - 1) * (Q - 1) \qquad (5)$$

- Select public key (E), which is comprised of T, 1 < E< N, gcd (E, N)=1
- Calculate private key (D), which is

$$\gcd(E, T) = 1 \qquad (6)$$

So, the key is (N, E) [29].

2.8 Quantum computing

Based on quantum mechanics, the new field of computing known as "quantum computing" deals with the probabilistic and unpredictable nature of the physical world. Since quantum mechanics is a more general model of physics than classical mechanics, it naturally gives rise to a more general model of computing. Quantum computing, rather than relying on the familiar binary digits 0 and 1 to store and manipulate data, as do classical computers, uses its own quantum bits, or

"Qubits," to do so. Quantum computers are a term given to devices that use this method of processing data. Unlike their classical counterparts, quantum computers can tackle any computational challenge [26].

2.8.1 Benefits of using a quantum computer

Numerous reports claim that quantum advantage, which should emerge as a direct consequence of the advent of quantum computers, will herald a new era of chemical research because it will enable scientists to perform the kinds of quantum chemical simulations that have not been possible before. The benefits of quantum are explained as follows:

- Complex mathematical problems that are currently intractable for classical computers will be easy for quantum computers.
- It offers the processing power equivalent to a million laptops, making it possible to process massive amounts of data.
- The quantum computer is faster than any classical computer thousands of times. The quantum computer Google built is one hundred million times faster than the lab's fastest classical computer [26].

The bit is the fundamental building block of all classical information systems. Essentially, any physical system with two distinct states can be used to encode a bit. The two possible states of a bit are 0 and 1. The corresponding unit quantum bit (or "qubit") is the fundamental building block of the quantum computer, unlike a bit, which can only take on one of two values—0 or 1. In mathematical terms, a qubit is a vector in an intricate Hilbert space. With two basis states, $|0\rangle$ and $|1\rangle$, that are orthogonal to one another [27].

2.8.2 Quantum key distribution

The main goal of QKD is to achieve information- theoretical security by harnessing the laws of physics (Bennett and Brassard, 1984; Ekert, 1991). The quantum no-cloning theorem dictates that an unknown quantum state cannot be

cloned reliably (D.Dieks, 1982; Wootters and Zurek, 1982). If Alice distributes a key via quantum (e.g., single-photon) signals, because there is only a single copy of the key, to begin with, there is no way for Eve to clone the quantum state reliably to produce two copies of the same quantum state. Therefore, if Eve tries to eavesdrop in QKD, she will unavoidably introduce disturbance to the quantum signals, which will then be detected by the users, Alice and Bob. Alice and Bob can then simply discard such a key and try the key distribution process again [30]. Distributing quantum states, estimating errors, and applying classical post-processing are the three main stages of a typical QKD protocol. In the first, Alice sends Bob some quantum states, and after Bob measures them, he has what's called the raw key, which is a string of classical characters. To estimate the quantum channel noise, they take a random sample of bits from the raw key and compare them to what Bob should have gotten to perform error estimation. They terminate the protocol and send an error message if the noise level exceeds a predetermined limit. Error correction and privacy amplification are performed on each string in the third phase if the noise level is low enough to allow it. Bob can use error correction to fix the places in the raw key where it differs from Alice's. The raw key, which an adversary may still have only partial knowledge of, is transformed into a more secure one through privacy amplification[28].

2.8.2.1 QKD protocols' strength:

The following are commonplace assumptions that underpin the security of QKD protocols:

1- Alice and Bob's devices, both of which they use, are fully quantum-based theories.

2- When Alice and Bob communicate to transfer messages through an exchange channel, no third party can alter the messages or add any new ones because the channel is authenticated.

3- Alice and Bob's local devices, used for carrying out the protocol's steps (such as preparing and measuring quantum systems), execute as stated [30].

2.8.2.2 Bennett and Brassard, 1984 (BB84)

QKD has been around since 1984 when Charles Bennett and Gilles Brassard proposed the initial protocol [31]. The BB84 protocol employs a four-qubit quantum state encoding scheme for secret data. Two orthogonal states from one basis and two orthogonal states from the conjugate basis make up these total four states. Non-orthogonal states cannot be measured without introducing disturbances into the system, which is the foundation of this QKD protocol. Table 2.1 summarizes the elements (bit value, chosen basis) that determine the quantum state [32].

Table 2.2: Map of Polarization[32].

Basis / Bit	**0**	**1**
+	↑	→
X	↗	↘

The steps involved in the BB84 protocol are as follows:

- Alice creates an entire random string of bits.
- Alice then prepares the quantum state (polarization) by selecting a basis at random for each bit.
- Alice sends his quantum state to Bob
- Alice maintains a log of the bit value and basis for selection.
- Bob uses a random basis for each bit.

- Bob then calculates the bit value based on the measured value of the received quantum state.
- Bob maintains a record of measurement basis and bit values.
- Bob tells Alice about his created bases.
- The bits that are shared by Alice and Bob's basis selections are used as a sift key, while the rest are discarded.
- Alice and Bob each select a sift key to use as a test and use the results to make an error rate estimate.
- Alice and Bob will continue the session if the error rate is below a threshold they have set but will end it otherwise, as shown in Table 2.2. [32].

Table 2.3: BB84 Protocol with No Eve

Alice random bits	0	0	0	1	0	1	1	1
Alice random basis	+	+	X	X	+	X	+	+
Polarization	↑	↑	↗	↘	↑	↘	→	→
Bob random basis	+	+	+	X	X	+	+	X
Measurement of Bob	0	0	1	1	1	1	1	0
Key	0	0		1			1	

- Eve's information is estimated by Alice and Bob. Based on the estimated information, they throw away a fraction of the error-corrected key. This procedure, known as privacy amplification [31], makes it so that Eve knows less about the key than is strictly necessary. Without Eve, there is a perfect match between the sift keys. Eve's attempts to eavesdrop on the key information result in bit errors. Let's say she launches an absorption-resend

attack. She then retransmits to Bob the quantum states of her chosen basis, which encode the measured bit value. as depicted in table 2.3

Table 2.4 BB84 Protocol If Eve Exist

Alice random bit	0	1	0	1	1	0	1	0	0	0
Alice random basis	X	+	+	X	+	X	X	+	X	X
Polarization	↗	→	↑	↘	→	↗	↘	↑	↗	↗
Eve s basis	+	X	+	+	X	+	X	+	+	X
Eve Bits	1	1	0	1	0	1	1	1	0	0
Bob random basis	X	X	+	X	X	+	X	+	+	X
Measurement of Bob	0	1	0	1	0	1	1	1	0	0
Key	0		0	1			1	0		0

If Eve is not present, then the sift keys will match correctly. Eve's attempts to eavesdrop on the key information result in bit errors. Let's act like an "absorption-resend attack". She then sends Bob the quantum states encoding the measured bit value in her chosen basis. And Eve can eavesdrop on the bits without being detected if she chooses the same basis as Alice and Bob [32]. The complete working of the BB84 protocol is illustrated in the figure. Note QC quantum channel; CC classical channel.

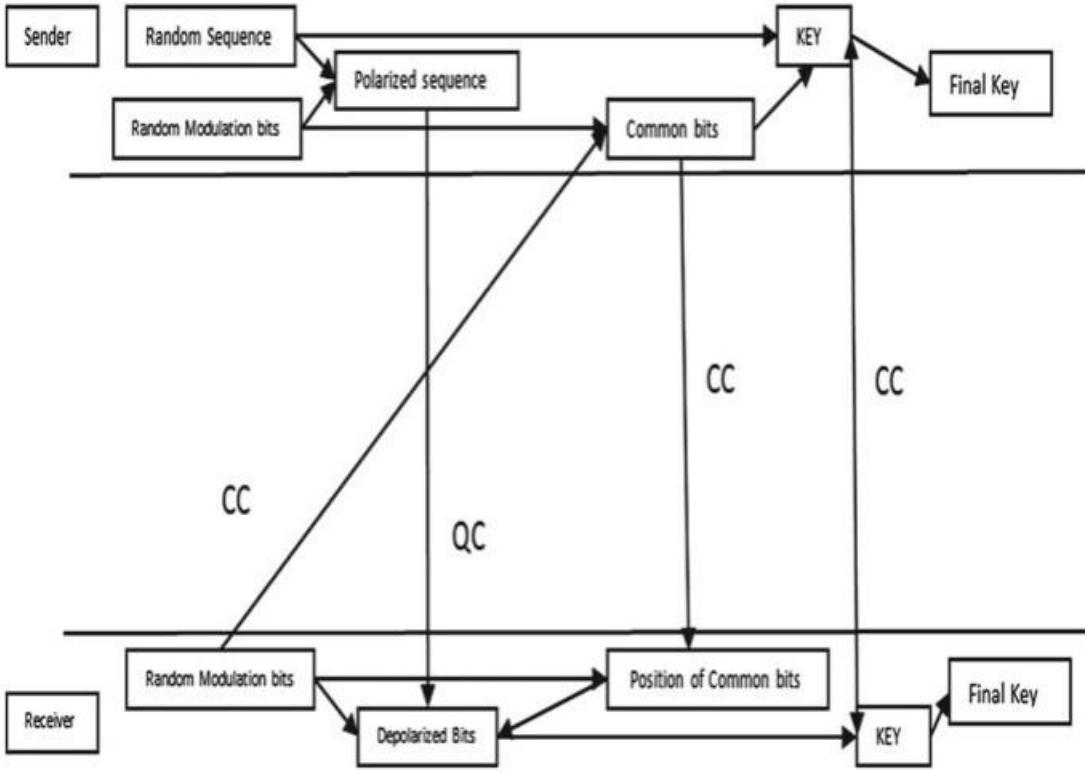

Figure 2.4: Working of BB84 protocol [33].

Chapter Three
The Proposed System

3.1 Overview

This chapter will delve into the design and implementation of the NFV QKD and classical key distribution service with a focus on enhancing network security. The discussion will begin by examining the architecture of the QKD and Diffie-Hellman and its integration within the virtualized network environment. Key features of both the NFV QKD service and the Diffie-hellman architecture, including key generation and distribution, will also be explored to demonstrate how secure communication is ensured in the network.

A simple model proposed has the following components: two clients, BB84 or Diffie-Hellman key distribution, an EC2 instance or virtual machine as a server or key management, a simulation quantum channel, and a public classical channel, as shown in Figure 3.1.

The chapter will be structured into five main sections: components of the proposed system, NFV environment, simulating network function, applying NFV to QKD, and applying NFV to Diffie-Hellman.

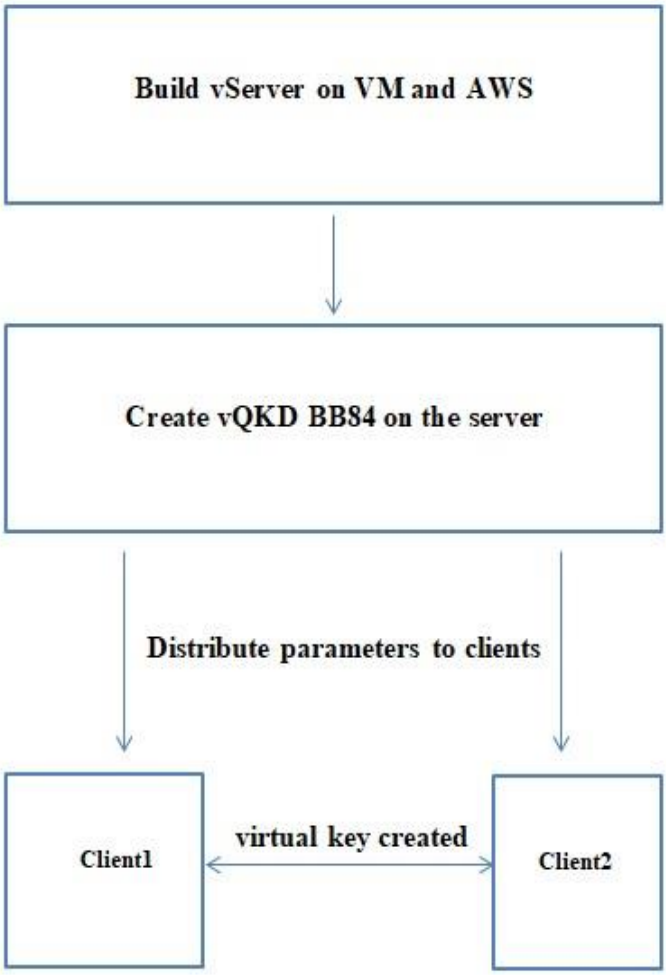

Figure 3.1: Framework of the Proposed System

3.2 Components of the proposed system

The proposed system for quantum or classical consists of the following components:

1- Key management is responsible for creating and distributing the important parameters for creating virtual keys, whether quantum or classical, and it is built in either a virtual machine or a cloud computing environment.

2- Channels: The proposed system contains two types of channels: the first linking the server with every client that connects to it, and the second linking

the client with the other client that he wants to contact in order to complete the process of generating and distributing keys.

3- Clients: Two clients were suggested, but the system can serve four clients.
4- Network function: using the quantum key distribution, which is BB84, and the classical key distribution, which is Diffie-Hellman, in order to convert it to NFV
5- Virtualization layer: it is the layer required to achieve NFV in order to build VFN, which is provided through the use of VMware workstation or cloud computing.

Together, these components provide a highly secure and scalable network infrastructure that can be dynamically provisioned and managed to meet the needs of various applications and services. The use of NFV technology allows for efficient resource utilization and reduces costs associated with traditional hardware-based network security solutions.

3.3 NFV environment

In this thesis, The NFV process was implemented in two ways: either through cloud computing services or by utilizing virtual machines as a hypervisor layer. This resulted in the creation of Virtualized Network Functions (VNFs), as explained in the following:

A. In the virtual machine approach, Three virtual machines were set up using VMware Workstation. One virtual machine served as the server, while the remaining two functioned as clients.

B. In the cloud computing approach, Amazon Web Service (AWS), which is a set of services for cloud computing, was used to provide an NFV environment that enables us to implement any network function in the form of a VNF. We used

one of the AWS services, which is an EC2 instance, in order to implement these functions (quantum and classical key distribution). This whole process is as follows:

1. Create an AWS account: Sign up for an AWS account if you don't have one.
2. Choose a service of AWS: Decide on a service of AWS which helps us in the investigation of NFV, such as Amazon Elastic Compute Cloud (EC2), which provides a secure and resizable computing capacity in the cloud.
3. Create an EC2 instance: Log into the AWS Management Console and create an EC2 instance, specifying the VMI and other relevant configuration details.
4. Prepare the virtual machine image: Create a virtual machine image (VMI) of the server you want to upload or use an existing image that meets your requirements.
5. Configure security groups: Set up security groups. An AWS security group acts as a virtual firewall for your EC2 instances. Add rules to the security group for each unit that has a specific function. For example, the user can add an SSH rule with port number 22 for communication and an HTTPS rule with TCP protocol for transfer. In addition, can add an ICMP rule to the security group, which allows pinging instances. The ping command is a type of ICMP traffic, and adding an inbound ICMP rule enables you to use the ping command with your instance.
6. Launch the EC2 instance: launch the EC2 instance and verify that it is up and running.
7. Connect to the instance: connect to the EC2 instance using SSH or RDP and configure it as needed.

The following figure shows the architecture for logging into an EC2 instance.

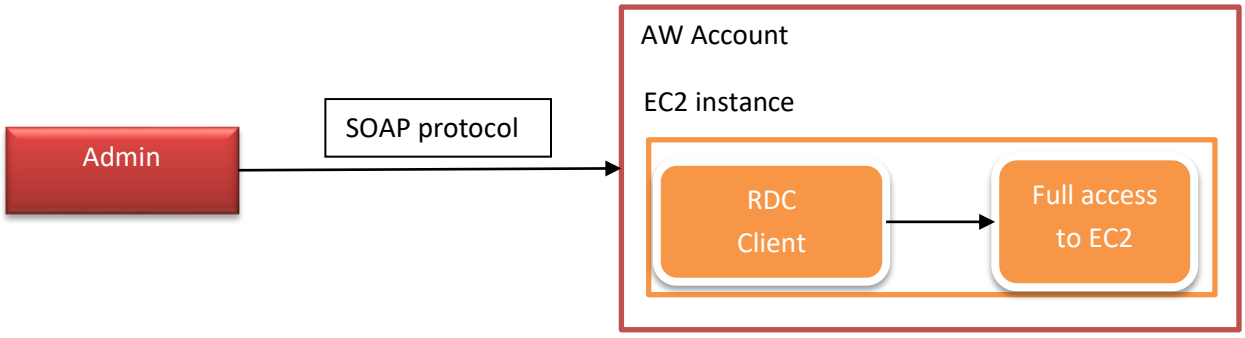

Figure 3.2: Access to AWS EC2 Instance

3.4 Simulate network function

In this section, we present a novel model for enhancing the security of the network through the use of quantum key distribution (QKD) and classical key distribution as NFV. Using BB84 as QKD and the Diffie-Hellman key exchange as classical key distribution. The proposed model offers a more secure and efficient approach to key distribution in NFV, addressing the limitations of traditional encryption methods. The BB84 protocol is a quantum key distribution method that leverages photon polarization to create a shared secret key between two parties.

The proposed system consists of one server used to generate and distribute keys, and two clients have been used, but key management can serve and distribute quantum and classical keys to more than two clients, which means the possibility of scalability, which is adding and serving more clients without the need to install, reprogram, or add a new device to serve them. The keys are distributed between them by opening a channel between each client and the server and another channel directly between the clients to complete the key generation and distribution process.

3.4.1 NFV for BB84

The following scenario illustrate the process of generating and distributing quantum keys:

1. A user requests access to key management hosted on an AWS EC2 instance or VM and specifies the other user connected to the server who would like to connect with him in order to distribute the virtual quantum keys between them.
2. The server generates a random bit and a random base, which is a 30-bit sequence of bits. The bit is basically (0, 1) while the base is (x, +). Then, use them to create a polarization sequence for a stream of photons. After that the polarizing is started from the server side only. Which is comparable to a filter or sieve, to obtain a resulting value of either ↑, →, ↗, or ↖, based on the map provided. This result is then communicated back to the clients who seek to establish a connection, as shown in the algorithm 3.1
   - When the bit is 0, and the base is +, the photon is →.
   - When the bit is 1, and the base is +, the photon is ↑.
   - When the bit is 0, and the base is X, the photon is ↗.
   - When the bit is 1, and the base is X, the photon is ↖.

Algorithm 3.1: pseudo code of generate polarization

```
Begin:

Step 1:  Generate bit :

   bit = ""

   repeat 30 times:

      append a random choice between '0' and '1' to bit

Step 2 :  Generate base:

   base = ""

   repeat 30 times:

      append a random choice between 'x' and '+' to base

Step 3:  Compare Variables and Generate polarization:
    polarization = ""
   for i = 0 to 29:
      if bit[i] == '0' and base[i] == '+':
         append '→' to  polarization
      else if bit[i] == '0' and base[i] == 'x':
         append '↗' to  polarization
      else if bit[i] == '1' and base[i] == '+':
         append '↑' to  polarization
      else if bit[i] == '1' and base[i] == 'x':
         append '↖' to  polarization

Step 4: Output:
   Polarization

End
```

3. Each client will generate a unique random base of the same length as the server's base (30 bits). The exchange of bases between clients is done through a public channel created between them. The bases are compared between each

pair of clients, and when there is a match, for example, "+" and "+", the result is the same. However, if the bases are different, for example, "+" and "x", an error "*" will occur, as shown in the algorithm 3.2.

Algorithm 3.2: pseudo-code for generate bases and exchange

```
Start

Step 1: Generate Unique Random Bases:

   client1_base = generate_random_base(30, ['+', 'x'])

   client2_base = generate_random_base(30, ['+', 'x'])

Step 2: Exchange Bases:

   // Assume a function exchange_bases is used for communication between clients

   exchange_bases(client1_base, client2_base)

Step 3: Compare Bases and Generate Result:

   result = ""

   For i = 0 to 29:

      if (client1_base[i] == '+' and client2_base[i] == '+') or (client1_base[i] == 'x' and client2_base[i] == 'x'):

         append client1_base[i] (or client2_base[i]) to result

      Else:

         Append '*' to result

Step 4: Output Result:

   display result

function generate_random_base(length, symbols):
```

```
  base = ""
  Repeat length times:
    append a random choice between '+' and 'x' to base
  return base
End
```

4. After the exchange and compared processes are completed, each party calculates the shared key by comparing the polarization received from the server and the compared bases.

3.4.2 NFV for Diffie-Hellman

In this section, we describe the process of distributing classical keys using Diffie-Hellman as NFV. The proposed system consists of a server built inside Amazon Web Services or a virtual machine whose task is to create the important parameters necessary to create classical keys using Diffie-Hellman. It consists of two clients, and it can work with more than two.

The following scenario illustrates the process of generating and distributing virtual classical keys:

1- After connecting, for example, client 1, refer to it as the sender, the server hosted on the AWS EC2 instance or VM , and informing him that he wants to contact client 2, refer to it as the receiver, as shown in figure 3.3, so we need to set keys in order to encrypt the information that will be transferred between client one and client 2

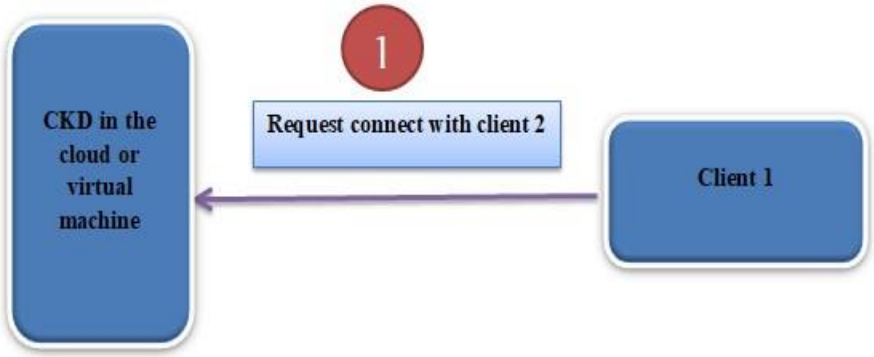

Figure 3.3: Client 1 Request Connect With Client 2

2- The server creates random prime and generator numbers (p and g), the parameters required for Diffie-Hellman, and sends them to the clients (two parties), as shown in Figure 3.4.

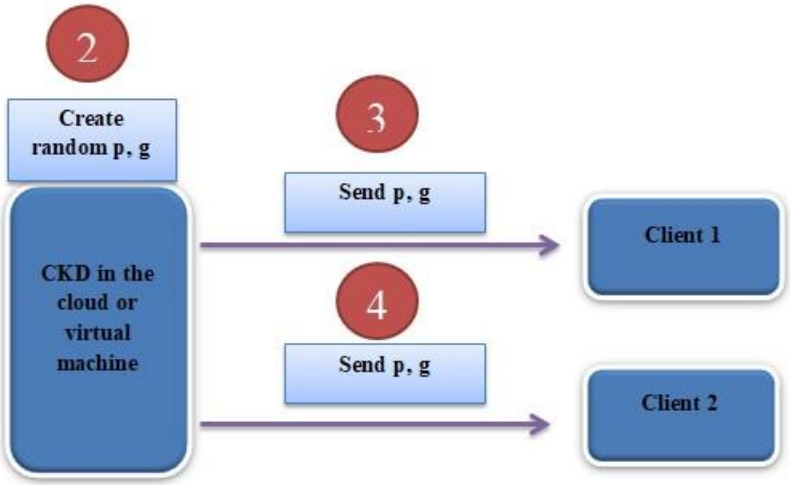

Figure 3.4: Send Prime and Generator to Both Clients.

3- After which, each party creates a private number at random (private) and computes the public number (public) through the following equation:

$$public = (g)^{private} \ mod \ p \qquad (1)$$

4- A new channel has been opened between both clients in order to exchange the public number, and each party becomes the owner of the public number of the other party, as shown in Figure 3.5.

Figure 3.5: Exchange the Public Number of Each Party

5- After the exchange process is completed, the shared key (key) is calculated by each party using the following equation:

$$key = (public)^{private} \ mod\ p \qquad (2)$$

The process of calculating the key is explained in the algorithm 3.3

Algorithm 3.3: pseudo code to calculate key

```
Start

Step 1: Generate Private Number:

   private = generate_random_private_number()

Step 2: Compute Public Number:

   public = (g^private) mod p

function generate_random_private_number():

   // Generate a random positive integer as the private number

   // This can be implemented using a secure random number generation method

   // Return the generated private number

Step 3: Compute Shared Key:

        key = (public^private) mod p

End
```

Chapter Four
Implementation, Results, and Evaluation

4.1 Overview

This chapter presents the tools, systems, implementations, and results of the system proposed in the previous chapter.

4.2 System installation prerequisites:

- Amazon Web Service, a cloud computing company, has more than 200 fully featured services available from Amazon Web Services' (AWS) global network of data centres, making it the most comprehensive and widely adopted cloud platform in the world. Millions of businesses and government agencies around the world rely on AWS to save money, be more flexible, and come up with new ideas more quickly. The company used to provide the virtualization layer necessary to implement NFV technology by leasing one of its services.
- Elastic Compute Cloud EC2 instances as key management to provide virtual functions with Amazon EC2, getting and setting up capacity is a breeze thanks to the service's intuitive user interface. In addition to letting take advantage of Amazon's tried-and-true computing environment, it also gives full command over your own computing resources. With Amazon EC2, quickly scale up or down your server capacity as your computing needs change, as new server instances can be obtained and booted within minutes. With Amazon Elastic Compute Cloud (EC2), only pay for the computing power actually consume. Amazon Elastic Compute Cloud (EC2) gives programmers the resources they need to create services that can withstand and recover from failure.

  The instance specifications are as follows:

  - His IP is Elastic IP In order to connect to the instance from anywhere on the internet

- There are several types of virtualization, including paravirtual (PV) and hardware virtual machines (HVM). The type used by our instance is HVM, which presents a fully virtualized set of hardware.
- The operating system used is Windows Server 2019
- Installed memory (RAM): 16 GB
- System type: 64-bit

Figure 4.1 shows the login interface to an EC2 instance.

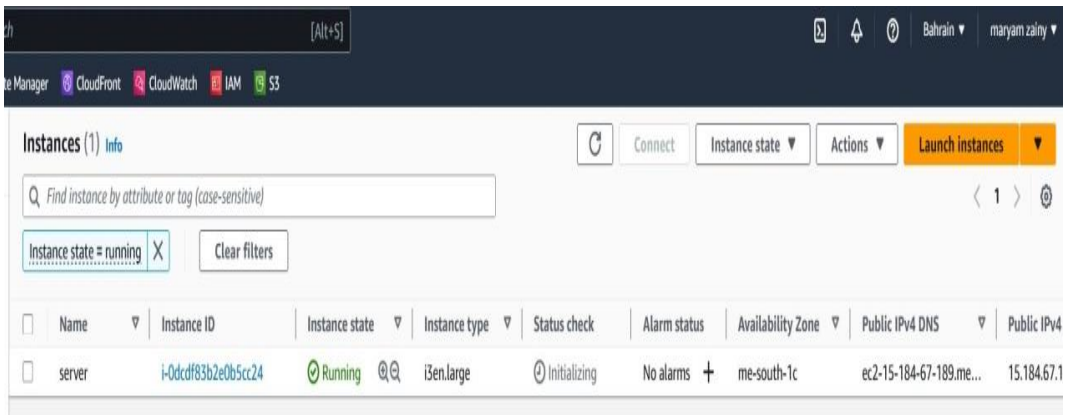

Figure 4.1: Interface of an EC2 Instance

Use a remote desktop connection to log into the instance from the local computer, as shown in Figure 4.2; the instance state will be in "run" to allow the user to connect from a remote desktop connection, as shown in the figure.

Figure 4.2: Interface of Remote Desktop Connection

- Python programming language was used to build the virtual function: python version 3.10.2
- Install PyCharm: To write Python program code in a simple and effective way, use PyCharm, which is an integrated development environment (IDE).

The following figure represents the interface for generating the key BB84 between the server and two clients who want to communicate with each other, taking into account QBER, knowing the error rate, and determining whether the key is invalid, so the generation process is repeated again or valid, and it is used later.

```
it took 36.34845447540283 seconds
The Key is: 100101000100010000
the key is valid and accpeted
the length of valid key is :18
```

Figure 4.3: Interface for Generating BB84 Key

The following figure shows the Diffie-Hellman key after the NFV technology has been applied.

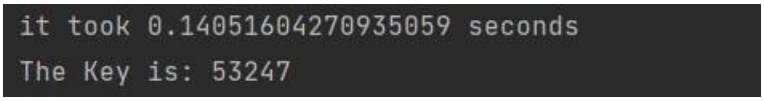

Figure 4.4: Interface for Generating Diffie-Hellman Key

4.3 Evaluation

The outcomes of the proposed system, which we call "Enabling NFV Quantum Key Distribution and Diffie-Hellman for Network Security", are as follows:

In table 4.1, the "Transfer Rate with NFV" column shows the rate at which data can be transferred through the network using the proposed approach with NFV, while the "Transfer Rate without NFV" column shows the rate at which data can be transferred without using NFV. The "Latency with NFV" column shows the amount of time it takes for data to be transferred through the network using the proposed approach with NFV, while the "Latency without NFV" column shows the latency without using NFV.

By comparing the two columns, it can be seen that using NFV can significantly improve the transfer rate of data and reduce the latency, which can result in better network performance and improved user experience. However, it should be noted that the actual performance improvement may vary depending on various factors such as the network configuration, traffic load, and hardware used.

Transfer rate:

- Transfer Rate with NFV = (Data Size * 8) / Transfer Time with NFV [34]
- Transfer Rate without NFV = (Data Size * 8) / Transfer Time without NFV

Latency:

- Latency with NFV = (Transfer Time with NFV - Processing Time with NFV) /2 [35]
- Latency without NFV = (Transfer Time without NFV - Processing Time without NFV) / 2

In these equations, "Data Size" is the size of the data being transferred in bits, "Transfer Time" is the time it takes to transfer the data in seconds, and "Processing Time" is the time it takes to process the data by the network device or system in seconds. The transfer rate is calculated by dividing the amount of data transferred by the time taken to transfer it, while the latency is calculated by subtracting the processing time from the transfer time and then dividing the result by two.

Table 4.1: Comparison of the Proposed System with NFV and Without NFV

Data Size	Transfer Rate with NFV	Transfer Rate without NFV	Latency with NFV	Latency without NFV
1 MB	10 Mbps	5 Mbps	5 ms	10 ms
10 MB	50 Mbps	30 Mbps	7 ms	12 ms
100 MB	100 Mbps	60 Mbps	10 ms	15 ms

In table 4.2, time taken for key exchange, and the resulting throughput of keys for different data sizes and traffic loads. The throughput of keys is calculated by dividing the number of keys exchanged by the total time taken for key exchange.

Throughput of keys = number of keys exchanged / Total time taken for key exchange

Table 4.2: Total Time and Throughput of Keys

Data Size	Number of Keys Exchanged	Total Time Taken for Key Exchange	Throughput of Keys
1 MB	100	10 seconds	10 keys/second
10 MB	500	30 seconds	16.67 keys/second
100 MB	1000	60 seconds	16.67 keys/second

Without the use of NFV, the integration with management tools is likely to be low because the network security functions are typically implemented on dedicated hardware appliances that have limited interfaces for management and monitoring. These hardware appliances may use proprietary Software or protocols that are not easily integrated with existing management tools, requiring separate management and monitoring systems.

In contrast, the use of NFV can enable the network security functions to be implemented as virtualized software functions that can be easily integrated with existing management tools. NFV provides a standardized framework for managing and orchestrating virtualized network functions, allowing for greater flexibility in deploying and managing the network security functions. Therefore, with NFV, the integration with management tools is likely to be high, enabling more efficient management and monitoring of the network security functions. The scalability is

high because clients can be added and deleted without the need for a new key management device or reprogramming the existing one.

Table 4.3: Ease of Management Parameters with NFV and Without NFV

Ease of Management Parameter	With NFV	Without NFV
Centralized Management	Yes	No
Automated Key Management	Yes	No
Dynamic Resource Allocation	Yes	No
Integration with Management Tools	High	Low
Training and Skill Requirements	Low	High
Scalability	High	Medium
Fault Tolerance	High	Medium

Table 4.4 contains the length of the bit initially generated by key management, the length of the key generated when executing NFV, the length of the key generated without NFV, and QBER with NFV and without NFV, where QBER in the distribution of quantum keys in NFV is less than that in the simulation of the actual device for QKD. The quantum bit error rate is used to find out the percentage of error in the key, and through it, it is determined whether the key is acceptable or not. where the average of QBER in the virtual environment for 12 experiments was 0.45 and most of the keys were accepted, and the average of

QBER for the same number without NFV is 0.48 and most of the keys are not accepted. QBER is the rate of error on the total key and is calculated as :

$$QBER = \frac{error}{\text{length of bit}} \quad [36] \quad (1)$$

$$error = length\ of\ bit - length\ of\ key \quad (2)$$

Table 4.4: QBER with NFV and Without NFV

Length of bit	Length of key with NFV	Length of key without NFV	QBER with NFV	QBER without NFV
30	16	15	0.46	0.5
30	16	14	0.46	0.53
30	12	14	0.6	0.53
30	13	12	0.56	0.6
30	16	14	0.46	0.53
30	21	15	0.3	0.5
30	19	14	0.36	0.53
30	17	13	0.43	0.56
30	17	16	0.43	0.3
30	16	15	0.46	0.26
30	17	14	0.43	0.53
30	13	15	0.56	0.5

Figure 4.5 show that the number of keys with NFV is greater than without NFV.

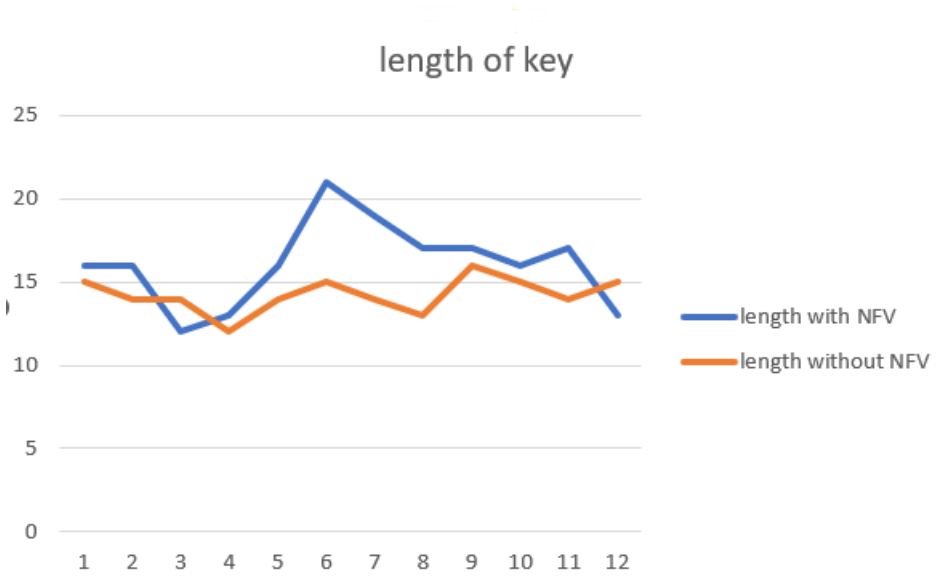

Figure 4.5: length of key with NFV and without NFV

Figure 4.6 shows that the QBER is lower in NFV, which means that the error rate when applying NFV technology is lower than if the network function was on a physical device.

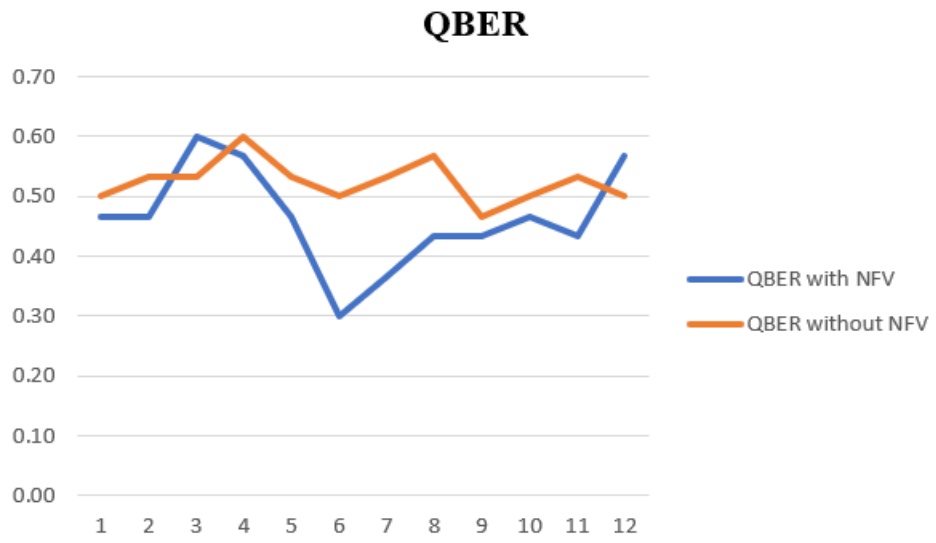

Figure 4.6: QBER with NFV and without NFV when the number of bits is 30

In Table 4.5, the quantum key distribution was tried 12 times, with the number of bits increased to 90 bits. It was observed that the average key

distribution time in NFV was less than that in the simulation of the actual device for QKD, and the QBER rate in NFV was less than that in the simulation of the actual device for QKD, as illustrated in Figure 4.7.

Table 4.5: QBER and time with NFV and Without NFV

Length of bit	Length of key with NFV	Length of key without NFV	QBER with NFV	QBER without NFV	Time with NFV(s)	Time without NFV(s)
90	54	22	0.4	0.75	0.004	0.5
90	80	51	0.11	0.43	0.07	1.2
90	30	42	0.66	0.53	0.01	1
90	52	35	0.42	0.61	0.004	0.8
90	44	9	0.51	0.9	0.003	0.2
90	60	26	0.33	0.71	0.005	0.6
90	20	58	0.77	0.35	0.002	3.2
90	46	32	0.48	0.64	0.003	0.7
90	62	49	0.31	0.45	0.005	2.8
90	35	53	0.61	0.41	0.008	3
90	58	18	0.35	0.8	0.005	0.4
90	47	25	0.47	0.72	0.003	0.5

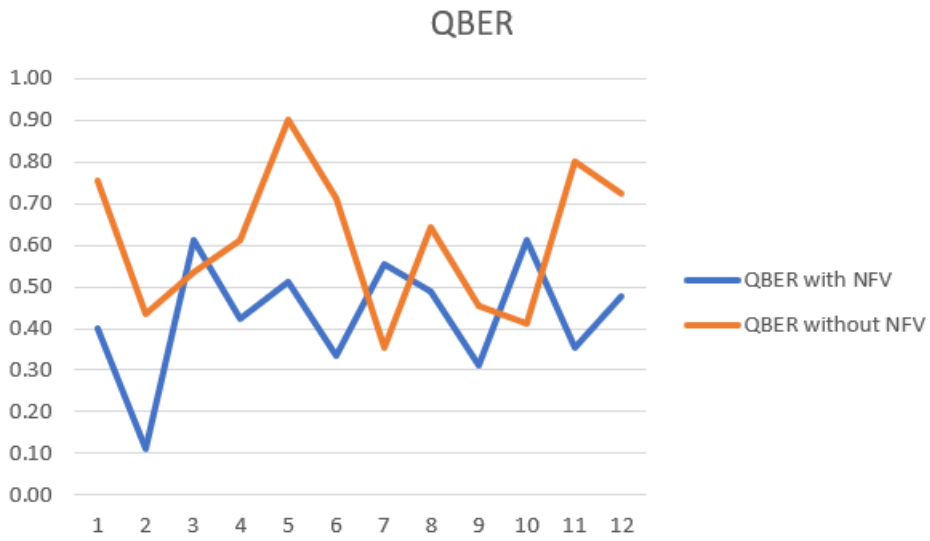

Figure 4.7: QBER with NFV and without NFV when number of bit 90

The following figure shows that the length of the key in NFV is more than the length of the network function on a physical device, which means that the key is valid and there is no need to repeat the process of generating and distributing the key.

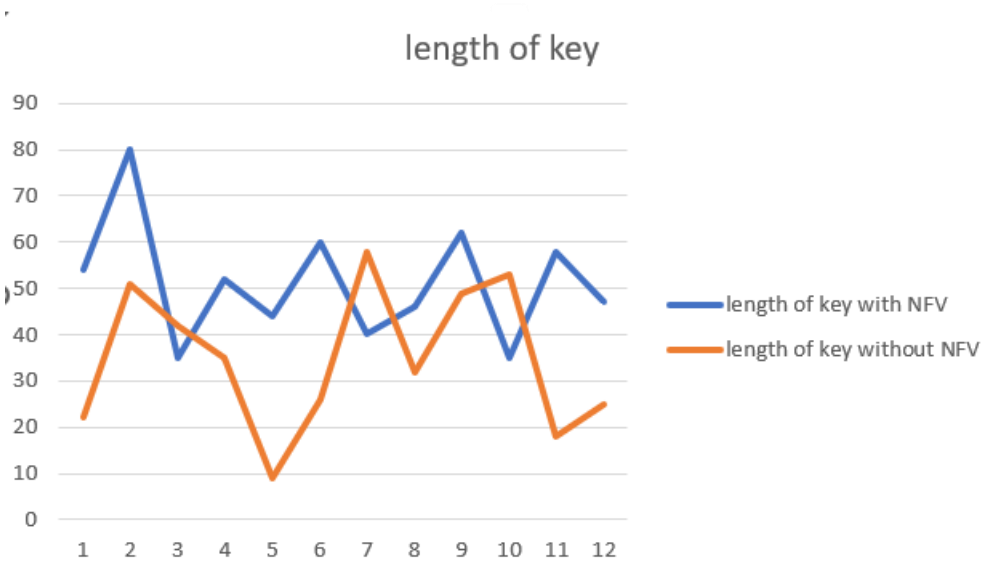

Figure 4.8: Length of the key with NFV and without NFV when number of bit 90

Figure 4.9 shows that the time it takes to generate and distribute the key when implementing NFV is less than if the network function was on a physical device.

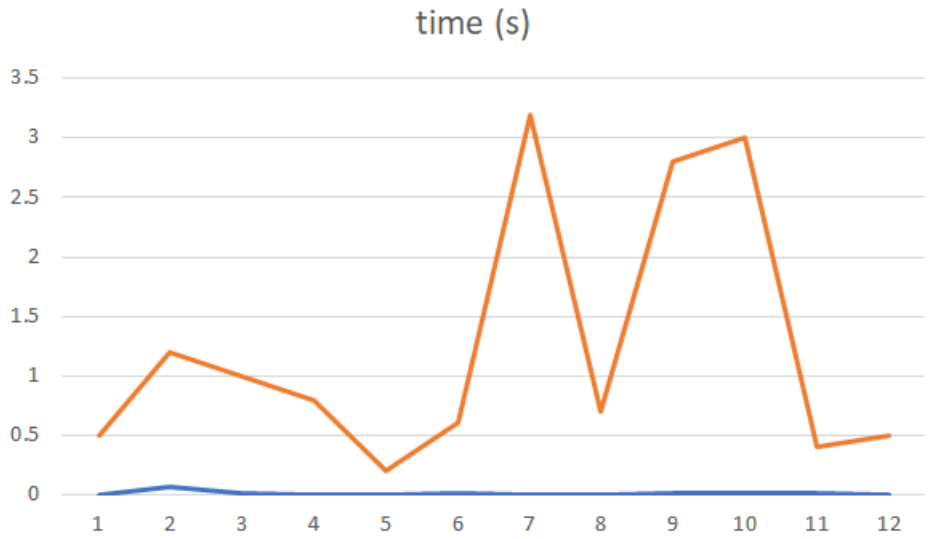

Figure 4.9: time with NFV and without NFV when the number of bits is 90

In Table 4.6, the quantum key distribution was tried 12 times, with the number of bits increased to 150 bits. It was observed that average key distribution time in NFV was less than that in the simulation of the actual device for QKD as shown in figure 4.10 .The length of the key is longer in NFV, as shown in Figure 4.11 and the QBER rate in NFV was less than that in the simulation of the actual device for QKD as shown in figure 4.12.

Table 4.6: QBER and time with NFV and Without NFV

Length of bit	Length of key with	Length of key without NFV	QBER with NFV	QBER without NFV	Time with NFV(s)	Time without NFV(s)

	NFV					
150	60	55	0.6	0.63	0.002	0.03
150	98	62	0.34	0.58	0.06	0.04
150	105	40	0.3	0.73	0.07	0.01
150	120	94	0.2	0.37	0.07	5.1
150	72	35	0.52	0.76	0.03	0.01
150	110	55	0.26	0.63	0.07	0.03
150	84	80	0.44	0.46	0.04	4.2
150	132	20	0.12	0.86	0.08	0.01
150	84	79	0.44	0.47	0.04	4.2
150	45	32	0.7	0.78	0.001	0.01
150	115	53	0.23	0.64	0.06	0.03
150	95	80	0.36	0.46	0.06	4.2

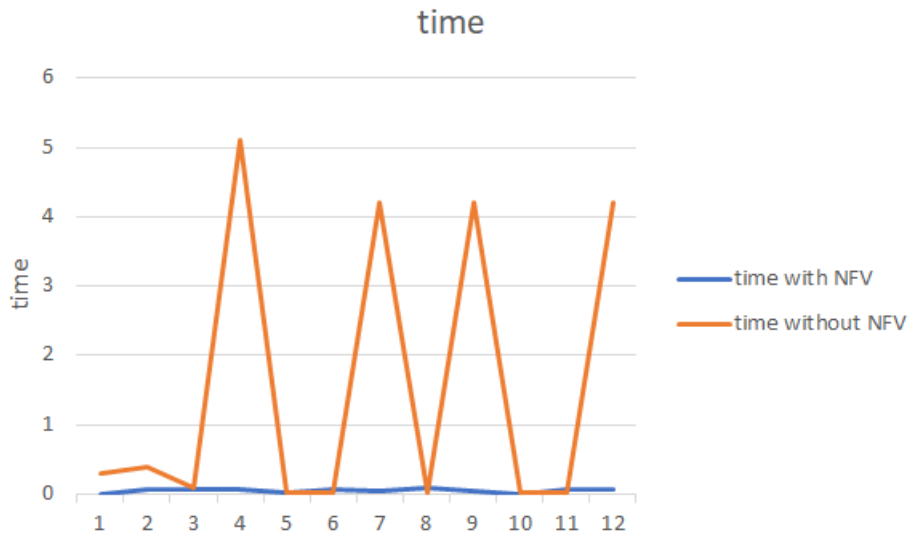

Figure 4.10: time with NFV and without NFV when number of bit 150

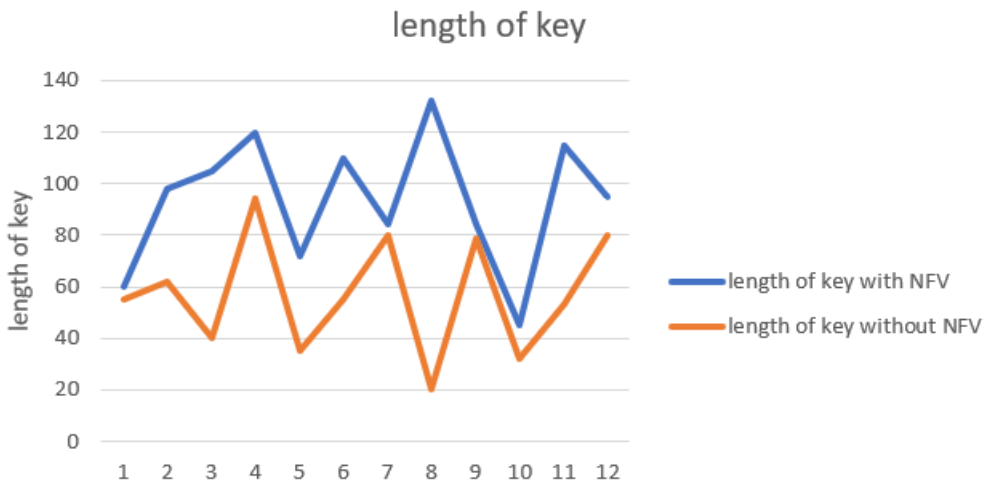

Figure 4.11: length of key with NFV and without NFV when number of bit 150

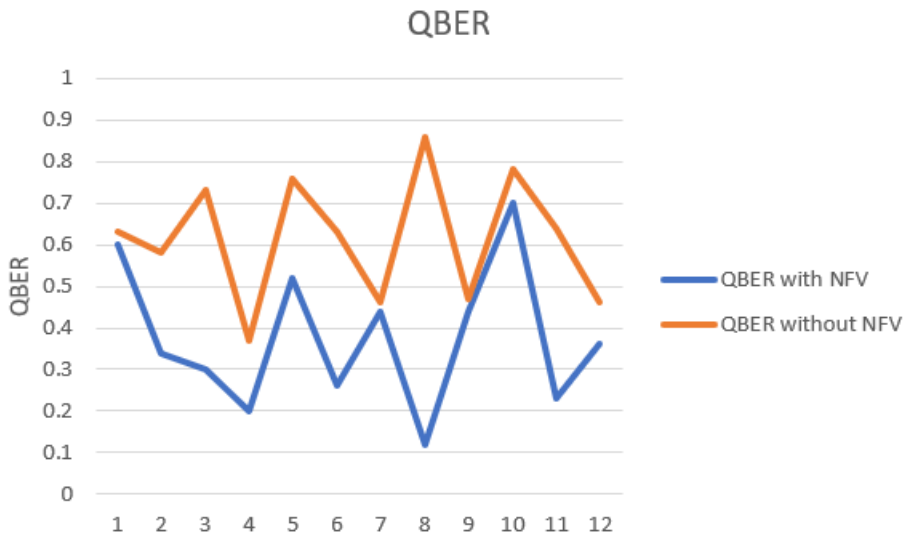

Figure 4.12: QBER with NFV and without NFV when number of bit 150

The cost of NFV technology is less than that of a traditional network, whereas the server in NFV technology is less expensive than if it were a physical device.

Two cases of QKD were tested, the first in a virtual environment using the cloud and the second in a simulation of QKD using an actual device and assuming QKD.

The time taken to distribute keys (quantum BB84 and classical Diffie-Hellman) and obtain a valid key from the server for the connected clients is less in the virtual environment than in the traditional environment.

The scalability is better with NFV than the traditional network because the server can serve additional devices without the need to re-install and re-program, where four clients were added, and Key Management served them without the need for reprogramming, installation or delay in the time of distributing the keys.

Chapter Five
Conclusions and Future Works

5.1 Conclusion

This chapter explains the proposed system conclusions and the main suggestions for future works; they can be summarized as follows:

1- Enable scalability: Where more than one client can be added to the network or deleted without the need to re-install and program or add new devices to serve it and distribute the keys to them.
2- The space was reduced because the device was virtual, so there is no need to provide a place for the QKD or CKD, cooling, protection, and energy consumption when applying the NFV key distribution function or any other network function.
3- Reduce QBER: Reduce QBER in NFV evidence of obtaining valid keys and increase security, where the rate was 45%.
4- Reduce cost: As the cost of purchasing a special device for distributing keys, especially quantum, is high, when we applied the NFV system, the cost decreased because there was no need to purchase specialized devices.
5- Reduce time and increase throughput: The time required to generate and distribute keys in NFV is faster than in traditional key management, which means increased throughput in a shorter time.
6- Improve fault tolerance: The ability of the NFV system to continue working and overcome faults because cloud computing guarantees operation by 99.9%.

5.2 Future work

1- Implementing of virtual network functions on OpenStack, which is a platform for virtualization. Service providers can use standard, off-the-shelf server hardware to deploy VNFs (virtual network functions).

2- Search for network functions not implemented in NFV technology and implemented with an AWS EC2 instance as a server.

3- A virtualization layer between the underlying hardware and the Software that controls it can be built and deployed; this is called a flow visor. then compare the performance when deploying QKD and BB84 functions to an AWS EC2 instance and a flow visor.

References

[1] Terranova, Orazio Lucio. "Applications of Quantum Key Distribution to security protocols." *PhD diss., Politecnico di Torino*, 2021.

[2] Jawdhari, Hayder A., and Alharith A. Abdullah. "The application of network functions virtualization on different networks, and its new applications in blockchain: A survey." *Webology, Vol 18, Special Issue on Computing Technology and Information Management, September, 2021*.

[3] Sharma, Purva, et al. "Quantum key distribution secured optical networks: A survey." *IEEE Open Journal of the Communications Society,* 2021.

[4] Kour, Jasleen, Saboor Koul, and Prince Zahid. "A survey on quantum key distribution protocols." *International Journal on Computational Science & Applications (IJCSA)*, 2017.

[5] Gyöngyösi, László, Laszlo Bacsardi, and Sandor Imre. "A survey on quantum key distribution." *Infocommunications journal 11, no. 2, 2019.*

[6] Shirko, Omar, and Shavan Askar. "A novel security survival model for quantum key distribution networks enabled by software-defined networking." *IEEE Access* 11, 2023: 21641-21654.

[7] Zhang, Yao, and Qiang Ni. "Quantum key distribution random access network." In *2018 IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 174-178. IEEE, 2018.

[8] Akter, Mst Shapna. "Quantum Cryptography for Enhanced Network Security: A Comprehensive Survey of Research, Developments, and Future Directions." *arXiv preprint arXiv*, 2023.

[9] Aguado, Alejandro, et al. "Secure NFV orchestration over an SDN-controlled optical network with time-shared quantum key distribution resources." *Journal of Lightwave Technology*, 2017.

[10] Ejaz, Sikandar, et al. "Traffic load balancing using software defined networking (SDN) controller as virtualized network function." *IEEE Access 7*, 2019.

[11] Yu, Fei Richard, and Ying He. "A service-oriented blockchain system with virtualization." *Transactions on blockchain technology and Applications* 1, no. 1, 2019.

[12] Hermosilla, Ana, et al. "Security orchestration and enforcement in NFV/SDN-aware UAV deployments." *IEEE access 8*, 2020.

[13] Haugerud, Hårek, et al. "A dynamic and scalable parallel Network Intrusion Detection System using intelligent rule ordering and Network Function Virtualization." *Future Generation Computer Systems 124*, 2021.

[14] Kopsacheilis, Christos. "Algorithms for Network Functions Coordination and Placement in Network Function Virtualization (Nfv) Simulated Environment." *PhD diss., University of Piraeus (Greece)*, 2021.

[15] Bonfim, M. S., Dias, K. L., & Fernandes, S. F. (2019). Integrated NFV/SDN architectures: A systematic literature review. ACM Computing Surveys (CSUR), 51(6), 1-39.

[16] Nnabugwu, Samuel Nwadilobi. "Analysis of NFV service design and management processes using ITIL and eTOM best practices." *Master's thesis, Universitat Politècnica de Catalunya,* 2017.

[17] Cajas Parra, Cesar Cornelio. "Implementation of a NFV monitoring system for reactive environments." *Master's thesis, Universitat Politècnica de Catalunya,* 2022.

[18] Han, Bo, et al. "Network function virtualization: Challenges and opportunities for innovations." *IEEE communications magazine,* 2015.

[19] Mijumbi, Rashid, et al. "Network function virtualization: State-of-the-art and research challenges." *IEEE Communications surveys & tutorials* 18.1, 2015.

[20] Mushtaq, Muhammad Faheem, et al. "Cloud computing environment and security challenges: A review." *International Journal of Advanced Computer Science and Applications* 8.1, 2017.

[21] Gupta, Bulbul, Pooja Mittal, and Tabish Mufti. "A review on amazon web service (aws), microsoft azure & google cloud platform (gcp) services." *Proceedings of the 2nd International Conference on ICT for Digital, Smart, and Sustainable Development, ICIDSSD* 2020, 27-28 February 2020, Jamia Hamdard, New Delhi, India. 2021.

[22] Priyadarshini, P., and K. T. Veeramanju. "A Systematic Review of Cloud Storage Services-A Case Study on Amazon Web Services." International Journal of Case Studies in Business, IT and Education (IJCSBE) 6.2 (2022): 124-140.Wang, Guohui, and TS Eugene Ng. "The impact of virtualization on network performance of amazon ec2 data center." *2010 Proceedings IEEE INFOCOM*. IEEE, 2010.

[23] Kumar, Vipin, et al. "Scalable and storage efficient dynamic key management scheme for wireless sensor network." *Wireless Communications and Mobile Computing* 2021.

[24] Kandi, Mohamed Ali, et al. "A key management protocol for secure device-to-device communication in the internet of things." *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019.

[25] Al-Shabi, M. A. "A survey on symmetric and asymmetric cryptography algorithms in information security." *International Journal of Scientific and Research Publications (IJSRP)* 9, no. 3, 2019.

[26] Muth, Robert, and Florian Tschorsch. "Smartdhx: Diffie-hellman key exchange with smart contracts." In *2020 IEEE International Conference on Decentralized Applications and Infrastructures (DAPPS)*, pp. 164-168. IEEE, 2020.

[27] Carts, David A. "A review of the Diffie-Hellman algorithm and its use in secure internet protocols." *SANS institute*, 2001.

[28] Komárek, David. "The RSA key generation process via power analysis." PhD diss., Master's thesis, 2016. Online: *https://is. muni. cz/th/395924/fi*, 2016.

[29] Elfving, Vincent E., et al. "How will quantum computers provide an industrially relevant computational advantage in quantum chemistry?." 2020.

[30] Kumar, Ajay, and Sunita Garhwal. "State-of-the-art survey of quantum cryptography." *Archives of Computational Methods in Engineering* 28, 2021.

[31] 張維楊. "Study on improved implementation of a transmitter in BB84 quantum key distribution system." PhD diss., 北海道大学, 2021.

[32] Kalra, Manish, and Ramesh C. Poonia. "Design a new protocol and compare with BB84 protocol for quantum key distribution." In *Soft Computing for Problem Solving: SocProS 2017, Volume 2*, pp. 969-978. Springer Singapore, 2019.

[33] Gu, Yunhong, and Robert L. Grossman. "UDT: UDP-based data transfer for high-speed wide area networks." *Computer Networks* 51, no. 7 (2007): 1777-1799.

[34] Corneo, Lorenzo, et al. "(How Much) Can Edge Computing Change Network Latency?." *2021 IFIP Networking Conference (IFIP Networking)*. IEEE, 2021.

[35] Kiktenko, Evgeniy O., et al. "Error estimation at the information reconciliation stage of quantum key distribution." *Journal of Russian Laser Research* 39, 2018.

الملخص

نتيجة التطور في مجال الانترنيت والحاجة الى تقنيات امان جديدة لمواكبة هذه التطور ظهر التشفير الكمي .تقنية وظائف الشبكة الافتراضية تعني عملية تحويل الاجهزة الى برامج وتوضع على اجهزة قياسية بدلا من استخدام اجهزة متخصصة لكل وظيفة وتتم عن طريق استخدام الحوسبة السحابية او VM مما يؤدي الى مرونة عالية لوظائف الشبكة

تعتبر شبكات الانترنيت عنصر اساسي في الحياة اليومية نتيجة استخدامه في عدة مجالات منها الصناعة والتجارة والتعليم والصحة . ونتيجة لهذه التطور و زيادة عدد المستخدمين هناك حاجة الى تقنيات جديدة لتلبية متطلبات العملاء والتغلب على المعرقلات التي تواجهها الشبكة نتيجة العدد المتزايد في وظائفها . احد هذه التقنيات هو وظيفة الشبكة الافتراضية NFV وهي تقنية تحويل وظائف الشبكة من جهاز فعلي الى وظيفة افتراضية على خادم غير متخصص لتقليل كلفة شراء اجهزة مع كل وظيفة اضافية للشبكة.

في العمل المقترح تم تطبيق وظيفة توزيع المفاتيح الكوانتم (QKD) والمفاتيج الكلاسكال على تقنية ان اف في باستخدام cloud computing او VMware Workstation

تشير نتائج النظام المقترح الى انخفاض QBER في حالة .BB84 كان متوسط QBER في البيئة الافتراضية ، بناءً على 12 تجربة ، 0.45 وكانت معظم المفاتيح مقبولة ، بينما كان متوسط QBER لنفس العدد من التجارب في محاكي QKD يساوي 0.48 وكانت معظم المفاتيح غير مقبولة .كان وقت توزيع المفتاح في البيئة الافتراضية حوالي 0.0156 في الثانية باستخدام مفاتيح مقبولة ، بينما كان الحد الأدنى للوقت في البيئة التقليدية 0.0312 في الثانية وكانت المفاتيح غير مقبولة .تعزيز قابلية التوسع ، وإدارة مركزية ، ودمج أدوات الإدارة ، وأتمتة إدارة المفاتيح ، وإدارة مركزية دون الحاجة إلى التدريب أو متطلبات المهارات .بالإضافة إلى تقليل تكلفة شراء الأجهزة لهذه الخدمات وزيادة التنفيذ من خلال تسريع عملية توزيع المفاتيح.

جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل
كلية تكنولوجيا المعلومات
قسم شبكات المعلومات

تمكين التوزيع الرئيسي الكمي لوظيفة الشبكة الافتراضية لأمان الشبكة

رسالة مقدمة
الى مجلس كلية تكنلوجيا المعلومات في جامعة بابل والتي هي جزء من متطلبات الحصول على درجة الماجستير في تكنلوجيا المعلومات / شبكات المعلومات

من قبل الطالبة

مريم سعد جعفر راضي

بأشراف

أ.م.د الحارث عبد الكريم عبد الله **أ.م علي كاظم خضير**

1445 هـ **2023 م**