# DESIGN AND IMPLEMENTATION OF NOVEL QUANTUM KEY DISTRIBUTION PROTOCOL FOR QUANTUM CRYPTOGRAPHY ON SOFTWARE DEFINED NETWORK

Thesis Submitted for the Award of the Degree of

## DOCTOR OF PHILOSOPHY

in

**Electronics and Communication Engineering**

**By**

**Hardeer Kaur**
Registration No. 42000085

Supervised By
Dr. Jai Sukh Paul Singh (23875)
**Electronics and Communication Engineering (Assistant Professor)**
**Lovely Professional University**


LOVELY PROFESSIONAL UNIVERSITY, PUNJAB

2025

# DECLARATION

I, hereby declare that the presented work in the thesis entitled "**Design and Implementation of Novel Quantum Key Distribution Protocol for Quantum Cryptography on Software Defined Network**" in fulfilment of the degree of **Doctor of Philosophy (Ph.D.)** is the outcome of research work carried out by me under the supervision of Dr Jai Sukh Paul Singh working as **Assistant professor**, in the Electronics and Communication Department of Lovely Professional University, Punjab, India. In keeping with the general practice of reporting scientific observations, due acknowledgements have been made wherever the work described here has been based on the findings of another investigator. This work has not been submitted in part or full to any other University to Institute for the award of degree.

Hardeer Kaur
42000085
Electronics and Communication Engineering
Lovely Professional University
Punjab, India.

## CERTIFICATE

This is to certify that the work reported in the Ph. D. thesis entitled "**Design and Implementation of Novel Quantum Key Distribution Protocol for Quantum Cryptography on Software Defined Network**" submitted in fulfilment of the requirement for the award of the degree of **Doctor of Philosophy (Ph.D.)** in Electronics and Communications Engineering Department, is a research work carried out by Ms Hardeer Kaur, 42000085, is a bonafide record of her original work carried out under my supervision and that no part of the thesis has been submitted to any other degree, diploma or equivalent course.

Dr. Jai Sukh Paul Singh
Electronics and Communication Engineering
Lovely Professional University
Punjab, India.

# ABSTRACT

Data security has become one of the most pressing concerns in today's digital world, where enormous amounts of personal, financial, and organizational data are continuously processed, transmitted, and stored online. Traditional cryptographic systems, though robust, rely heavily on mathematical complexity, making them vulnerable to future threats posed by rapidly advancing computational power, especially quantum computing. Typically, a Quantum computer could perform complex calculations at blazingly fast speeds, which can only be dreamt with even the fastest supercomputers. Quantum Key Distribution (QKD) presents a promising alternative that leverages quantum mechanics to ensure highly secure communication. However, current QKD protocols like BB84 and B92 suffer from limitations in key generation rate, transmission distance, and adaptability to real-time network threats. This study proposes an adaptive Novel-QKD, an advanced QKD algorithm designed to overcome these constraints. Proposed Novel-QKD enhances security and availability by utilizing a multi-channel architecture, enabling continuous communication even when the network is partially compromised. Furthermore, the algorithm is integrated with Software-Defined Network (SDN), allowing dynamic reconfiguration and real-time adaptation to network threats, including eavesdropping and denial-of-service attacks. A software simulation environment was designed and developed based on Python to test the proposed algorithm, and comparative analysis was performed against renowned QKD algorithms. The novel-QKD algorithm proves to have higher key generation rates of 420 bits per sec compared to 115 bits per sec for BB84 and 74 bits per sec for B92. Providing 4 and 6 times higher key generation rates than BB84 & B92, respectively, under similar test conditions when using four paths. The percentage error in the key generation for the Novel-QKD algorithm stands around 8.8%, which is similar to the BB84 protocol. However, interestingly, the introduction of an eavesdropper does not significantly impact the network's performance. The Novel-QKD being implemented along with SDN adapts itself by avoiding key distribution through the compromised portions of the network and continuing the key generation using the remaining sections. Hence, until a sufficient number of paths are available the average error remains reasonably constant. Moreover, the algorithm showed improved resilience against common quantum attacks such as man-in-the-middle, photon number splitting, and denial-of-service, ensuring both confidentiality and availability. Our results highlight Novel-QKD's potential for secure, adaptive communication in both civilian and military applications, marking a significant step toward practical quantum-safe network architectures.

# ACKNOWLEDGEMENTS

The completion of this thesis stands as a testament to the collective efforts and support of numerous individuals, without whom this accomplishment would not have been possible. I extend my sincere gratitude to all those who contributed to the realization of this Ph.D. work. I am deeply grateful to Dr. Jai Sukh Paul Singh, my supervisor, whose unwavering support, guidance, and trust have been instrumental throughout this research journey. His dedication to both my academic and personal development has profoundly shaped my growth over the past four years. I particularly appreciate his commitment to fostering a conducive learning environment and nurturing his students' success. I am immensely grateful to the School of Electrical and Electronics Engineering at Lovely Professional University and the Research Department for their invaluable contributions to my journey. Their provision of essential infrastructure, resources, and timely guidance on academic norms and protocols has been indispensable. Additionally, I extend my heartfelt appreciation to the research panel members for their ongoing support. Their consistent evaluation of my progress and insightful suggestions have played a crucial role in guiding me towards the completion of this thesis. Finally, I want to express my heartfelt gratitude to my parents and family, who have been my unwavering sources of support and understanding throughout both my professional and personal pursuits. Their companionship, patience, and sacrifices have played a crucial role in helping me overcome the challenges of the research journey. Without their steadfast presence, this endeavour would have been significantly more challenging.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# 1. INTRODUCTION

In today's scenario, data is the most valuable thing. Many of the large cooperates do just data handling and administration. As more and more data is being stored in digital form, data security is becoming even more complicated. For many years, scientists have been developing means and methods for data protection. Data must be secured to be readily accessible to the authentic user and completely obscure. The most common ways to achieve this are user authentication and data encryption. Data encryption means performing some operation on the data to change its form so that even if the data is publicly available, the unauthenticated user cannot understand the data [1]. In a fundamental data encryption schematic, as shown in Fig.1.1, the first step is establishing a shared key available to both sender & receiver. The data is coded or encrypted using the security key and then sent over to the public channel. This data is available to one and all. However, as the data is encrypted, no one can understand what the message is. Only the intended user with the key to decode the message can understand the content [2] [3].

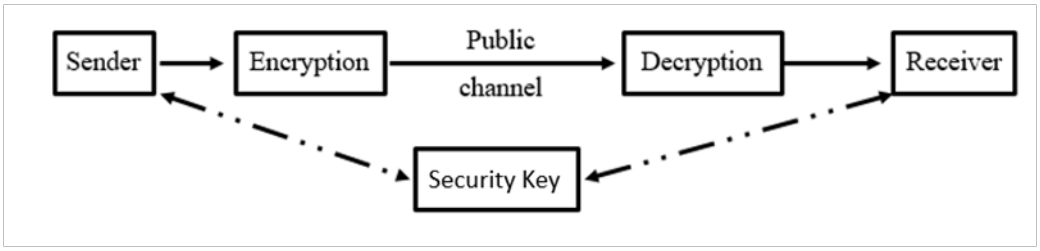



Figure 1.1: Basic structure of cryptographic communication.

One of the basic and most important roles of any cryptographic algorithm is to maintain its encryption when challenged by the onlookers. There are multiple ways by which eve /attacker will attempt to acquire control of the message/data being transmitted. In one such attack (brute force)[4], a classical computer can guess all possible key combinations to find the acceptable key. The possibility of finding a correct key relies on the number of attempts the computer can make. The commonly used 256-bit key has $2^{256}$ possible key combinations, and an attacker has to guess each one of them to get the correct key. This, being a cumbersome and time-consuming process, ensures the security of the current system. The time required to find the key will be in decades. It gives security to the systems, but it is theoretically possible. In the past decade, quantum computers have been developed tremendously. Quantum computers will exponentially enhance the computational capabilities of the attacker to find the key [5]. For a quantum computer with 256 bits, it will take only one iteration to decipher the 256-bit key. This poses a threat to the current encryption system. In an effort to find a cryptographic system which is resistant to such attacks, researchers are developing quantum cryptographic algorithms. We try to summarize some of the basic and important quantum algorithms. In an attempt to make cryptographic systems safe from a possible quantum attack. Research broadly works in two directions.

1. **Post-Quantum Cryptography (PQC)**, these algorithms will work on classical systems and channels but will remain secure against any attack from Quantum Computers [6]. These algorithms rely on the mathematical complexity of fathoming the key rather than physics. These algorithms tend to make the hacking so difficult and laborious that it is practically impossible to hack.

2. **Quantum key distribution (QKD),** the approach uses physics to ensure the safety of the encryption. These key distribution protocols have been developed for the last two decades. The first such algorithm was presented by 'Bennett and Brassard' in 1984 [7]; hence, it was named BB84 (explained in detail in the coming chapters). As every coin has two sides, QKD also has a bright side and a not-so-bright side. The dark side is the practical implementation, high initial cost, and low-key rates. QKD uses principles of quantum physics such as entanglement and no-cloning. It limits its scalability as it doesn't allow the usage of amplifiers and repeaters. In comparison, the classical communication channels can use repeaters and amplifiers to extend the range virtually to infinity. Whereas the maximum distance between which the QKD can be established depends largely on the capabilities of the channel medium. There are three challenges for any QKD setup:

   (a) Safety.

   (b) Scalability.

   (c) practical feasibility.

   Most of the QKD systems have addressed the safety concerns but have limitations on the scalability and Practical feasibility. The desired system can be idealized as a network of nodes in which multiple nodes can be added and removed without affecting communication. The most effective and easiest way is to have a network with a series of repeaters between the communication points. This will not only improve the distance of communication but also allow further addition of nodes. These nodes are generally expected to be safe and trusted nodes as used in DARPA [8], Beijing-Shanghai quantum line [9]. Having a trusted repeater everywhere is practically not sustainable and will have a sense of insecurity. The best way would be to develop repeaters that can be untrusted nodes. In addition, the integrity of the communication relies only on the end nodes, not on the repeaters in between. MDI (measurement device independent) QKD is a key for the development of these untrusted repeaters.

### 1.1. Research Questions and Hypotheses

This research is motivated by the limitations observed in existing Quantum Key Distribution (QKD) protocols such as BB84 and B92, which suffer from low key-generation rates, limited distance, and weak adaptability to dynamic network conditions. To address these gaps, this study proposes an SDN-integrated multi-path QKD algorithm (Novel-QKD) designed to enhance key-generation rate, reliability, and resilience against quantum and classical network attacks.

The research is guided by the following primary question:

*How can Quantum Key Distribution be enhanced to achieve higher key-generation rates, lower error rates, and improved reliability under dynamic network conditions?*

To systematically explore this overarching question, the following subsidiary questions are posed:

1. Can Software-Defined Networking (SDN) control improve QKD efficiency by enabling simultaneous multi-path key generation and aggregation?
2. How does SDN-based dynamic routing help mitigate the impact of eavesdropping and denial-of-service attacks on quantum channels?
3. What measurable improvements in key-generation rate and bit-error rate can the Novel-QKD protocol demonstrate compared with BB84 and B92 under identical simulation conditions?

Based on these questions, the following hypotheses are formulated:

- **H1:** Integrating QKD with SDN will significantly increase the average key-generation rate by allowing simultaneous quantum key generation over multiple network paths.
- **H2:** The multi-path aggregation mechanism will enhance reliability and maintain a near-constant bit-error rate even when one or more paths are compromised.
- **H3:** Dynamic SDN-controlled routing will strengthen resistance against eavesdropping, photon-number-splitting, and denial-of-service attacks by reconfiguring network paths in real time.

### 1.2. Quantum Physics

Quantum physics / Quantum information sciences [10]–[14] has been at the core of many modern developments in the field of cryptography, communication & understanding of particle behaviour, etc.

- **First Quantum revolution** (Developing the basic understanding of Quantum Physics). A new field of study, "Quantum Mechanics", started in the middle of the nineteenth century when, while working on blackbody radiation, physicist Gustav Kirchhoff showed that the emitted energy relates to the temperature and frequency of the emitted energy. Further, in 1900, Max Planck developed the formulation for Gustav's discovery. He based his study on a very important assumption. He considered radiation as a series of small packets of energy, not a continuous stream as though earlier. He names the small packets as "Quanta". Hence, the study became Quantum mechanics. In the first quantum revolution, we understood what it was and developed theories based on it. One such theory was the dual nature of particles, which helps us understand the periodic table.

- **Second Quantum revolution** (Development of the more practical functions of Quantum Mechanics) [15]. The second quantum revolution can be considered as the application phase. Thanks to the second revolution, we can develop new technologies, i.e. Quantum cryptography, Quantum computers, Quantum teleportation, etc. Shor, in the year 1994, proposed an algorithm which reduces the time required to factorize an integer [16] as a proof of Shor's algorithm, the RSA (Rivest, Shamir & Adleman) algorithm, which is commonly utilized for public key cryptosystems. It would take a computer with 2.8 GHz and four cores to run for thirteen months to decipher the key. However, if one uses a Quantum computer (with a sufficient number of Qbits), it would take not more than 1 second to decipher the key. Similarly, in the year 1996, Grover presented a new algorithm that tremendously reduced the search time in an unsorted database [17].

### 1.3. Basic Quantum Principles

- **Quantization:** This refers to the discretization of the energy into small packets of energy. These packets are referred to as Quanta; these small packets can be studied and used to develop new technologies.
- **No Cloning:** No cloning is the property of a quantum particle by virtue of which it is unattainable to create an exact replica of a known quantum state [18]. It helps keep the quantum systems safe and secure from any adversaries that try to hack the system by copying the state (or the data encrypted in the state of the quantum particles).
- **Superposition:** If a particle can exist in two or more states simultaneously, then the correct state can be defined only by superpositioning all the states (refer to Fig.1.2).

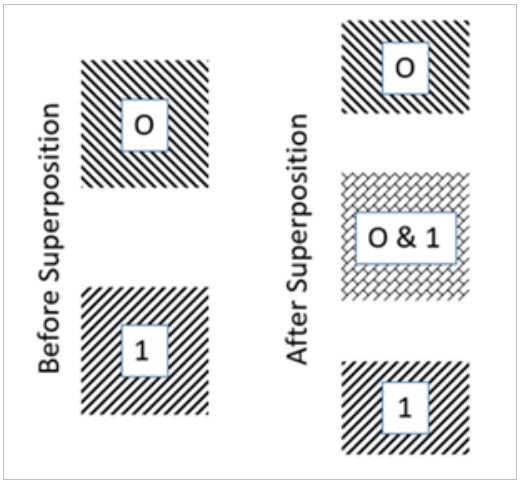



Figure 1.2: Principles of Superposition.

Taking an example, assuming two particles with values "0" & "1" correspondingly are superimposed over each other than the resulting particle could have the value "0", "1" or both "0 & 1" at the same

time.

- **Principle of uncertainty:** It states that a particle can exist in more than one state simultaneously. For every properly defined state of a particle, one measurement must represent the state with full certainty and one measurement with large randomness.

- **Tunnelling:** Tunnelling is the effect by virtue of which a particle can cross an energy barrier, which is higher than its own energy (refer to Fig. 1.3 ).

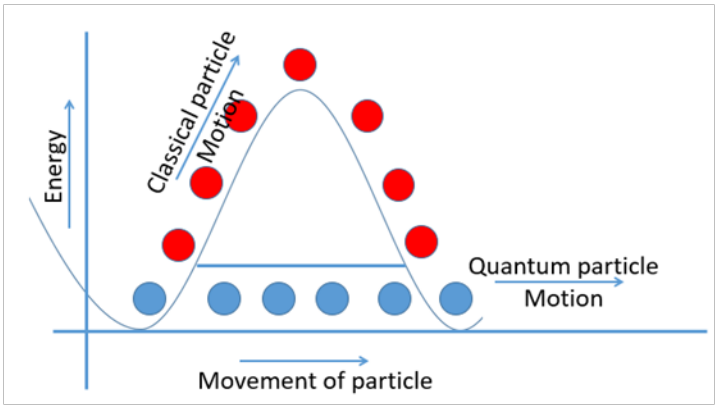



Figure 1.3: Principle of Tunnelling

- **Entanglement:** It is the property of quantum particles by virtue of which two entangled particles behave in conjunction with each other. Entanglement can be considered a manifestation of the law of conservation of energy. In a supposed event of a photon split, two new particles generated can be said to have opposite properties. Accordingly, their energies shall be the sum total of the energy of the splitting photon.

  This phenomenon can be explained (as shown in Fig.1.4) by using an example of two entangled particles, which can attain any value from the given set of values but with a condition that the sum of their instantaneous values is always 9 [19]. Hence, the change in the value of particle 1 will change the value of particle 2.

- **De-coherence:** In a perfect space where there is no interaction between the entangled particle and environment, the entanglement or cohesion remains forever. However, in a more practical scenario,

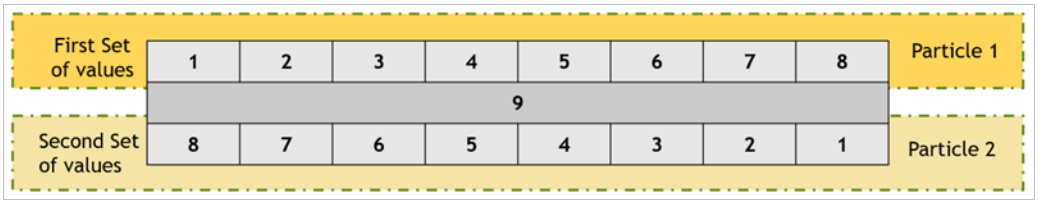



Figure 1.4: Principle of Entanglement

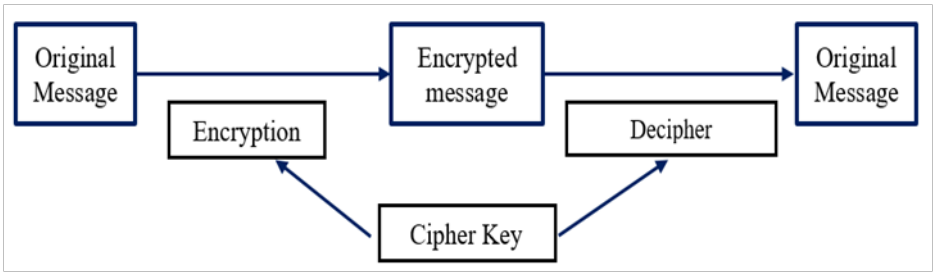



Figure 1.5: Basic format of Cryptographic communication using Cipher key

the entangled photons interact with the environment and, in some cases, lose their entanglement. This phenomenon of losing coherence is called De-Coherence.

## 1.4. Cryptography

The basic aim of cryptography is to provide a method by which information can be shared with a guarantee of security. This will help in building trusted networks where the utmost importance is given to the privacy & safekeeping of users and their data. This could be achieved by methods like coding of data, user authentication, user verification, digital signature, etc. Now, especially with the advancement in digital technologies, consumers want to have access to everything from the comfort of their homes or offices. The data shall be available whenever it is required and at whatever location it is requested. This is made possible with the new advancements in data encryption and user authentication methods. The term cryptography literally means "secret writing" in Greek. Examples of cryptography can be traced back to 2000 B.C. For example, the use of hieroglyphics by ancient Egyptians and Caesar cypher by Romans [20].

Nowadays, everyone uses cryptography, knowingly or unknowingly. Cryptography has become an inseparable part of human life and security. From banking transactions to watching TV uses cryptography of some sort for user identification and authentication. Cryptography has emerged as an important tool to protect privacy & user data. On the contrary, to the bulletproof security provided, cryptography is considered brittle. A single error in the programming or algorithm design can lead to a data breach [21]. Schneier [22] states that complete secrecy is a myth, and if the security is based on secrecy of key (or other data), then the system will remain fragile, concluded it. In case the secrecy is lost, then the complete system will collapse, and even worse, the users may not even know about the breach. He further states that to achieve perfect security is to embrace public security [22]. To achieve this, public key shall be used, and the algorithms shall be simple and based on the basic principles, not on the mathematical complexities.

In the most basic format (as shown in Fig.1.5), a Cipher Key is used. The sender encodes the original message by using a cipher key, the encrypted message is then send to the receiver through a public

channel. At the receiver's end, the message is deciphered using the same cypher key to generate the original message.

### 1.5. Cryptography algorithms

Multiple algorithms have been developed over time to make communication secure. Some of the important algorithms are detailed below:

- **Advanced Encryption Standard (AES):** It is an encryption algorithm that uses symmetric and was standardized in 2001, designed to replace DES [23]. It utilizes 128-bit blocks (16Byts) of data and supports the key lengths of 128, 192, or 256 bits. AES is recognized for its efficiency, robust security, and resistance to cryptanalytic attacks. Its widespread adoption across various platforms, including TLS, VPNs, and disk encryption, underscores its critical role in modern data security frameworks.

  AES, often referred to as the Rijmen algorithm [23], is a symmetric encryption standard, implying that it employs both encryption and decryption processes with the same key. AES, as a block cypher, encrypts and decrypts precisely 128 bits or 16 bytes of data in one go. The key employed in this process can be 128, 192, or 256 bits, riding on the expected level of security. The number of rounds performed in the encryption or decryption function is specified by the length of the key utilized (refer to Table. 1.1). Each round consists of 4 steps except the last round. The last round doesn't have the mix column step.

| Key length | Number of rounds | Level of security |
|---|---|---|
| 128 bit key | 10 Rounds | Good |
| 192 bit key | 12 Rounds | Better |
| 256 bit key | 14 Rounds | Best |

Table 1.1: Number of rounds used for encrypting 128 bit data.

  **Encryption in AES:** AES uses block of 16bytes (128bits) and for ease of understanding it can be considered as a 4 by 4 matrix (as shown below).

$$\begin{bmatrix} J0 & J1 & J2 & J3 \\ J4 & J5 & J6 & J7 \\ J8 & J9 & J10 & J11 \\ J12 & J13 & J14 & J15 \end{bmatrix}$$

  Encryption using AES can be subdivided into the following steps:

  1. Substitute Bytes: At this stage, each byte is replaced with a different byte. This is achieved by employing a lookup table, commonly known as the S-box. The substitution process ensures

that a byte is never replaced with itself and is also not substituted by a byte that is the complement of the original byte. This transformation results in the development of a matrix consisting of 16 bytes, which is systematically organized into a 4 x 4 format, similar to the previously established arrangement.

2. Shifting Row: As the name suggest the step involves shifting of each row by fixed number of places (As shown in the matrix below).

$$\begin{bmatrix} J0 & J1 & J2 & J3 \\ J4 & J5 & J6 & J7 \\ J8 & J9 & J10 & J11 \\ J12 & J13 & J14 & J15 \end{bmatrix} \rightarrow \begin{bmatrix} J0 & J1 & J2 & J3 \\ J5 & J6 & J7 & J4 \\ J10 & J11 & J8 & J9 \\ J15 & J12 & J13 & J14 \end{bmatrix} \begin{matrix} Not & Shifted. \\ Shifted & once. \\ Shifted & twice. \\ Shifted & thrice. \end{matrix}$$

3. Mixing of Columns: A predefined matrix is multiplied with each column to change the position of each byte.

$$\begin{bmatrix} J0 \\ J1 \\ J2 \\ J3 \end{bmatrix} \times \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} = \begin{bmatrix} K0 \\ K1 \\ K2 \\ K3 \end{bmatrix}$$

   This step is not executed in the last round.

4. Add Round Key: In this step 16 Bytes are converted to 128 bits. XOR operation is performed on each bit (obtained from last step) using round key.

   The above steps are repeated until all the round keys are added. The output "Cipher text" can be transferred securely through a public channel.

**Decryption in AES:** Decryption can be done by simply following the encryption steps in the inverse order.

1. Add Round Key: This step is smiler to that followed in the encryption stage. As the algorithm uses a symmetric key, same round key is used to perform XOR operation on the cipher text.
2. Inverse Columns Mixing: This stage bears resemblance to the Mix Columns phase found in the encryption process; however, the distinct variation lies in the matrix employed to execute the operation. In the Mix Columns procedure, each column undergoes transformation individually and without any dependency on the other columns, ensuring that the mixing operation remains exclusive to each column. In this process, matrix multiplication is applied. As a result of this calculation, we obtain the matrix product formed by combining the existing set of values with a fixed constant matrix.

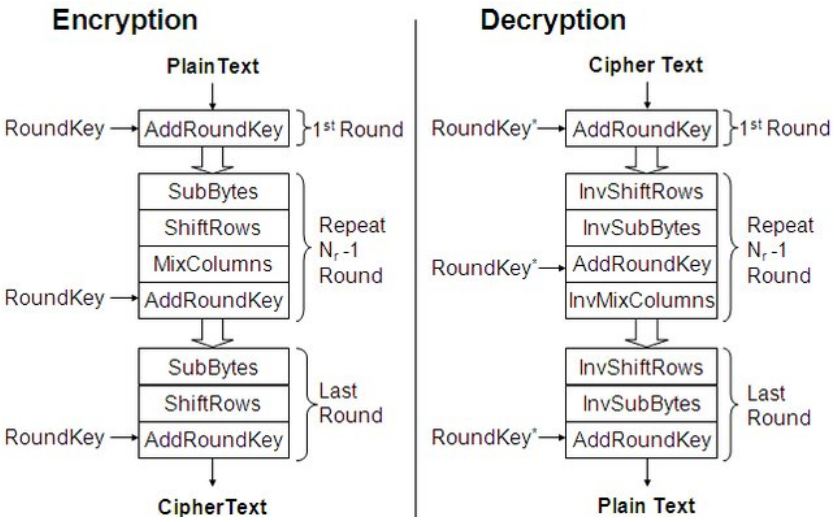



Figure 1.6: Flow chart for AES: Encryption and Decryption process.

$$\begin{bmatrix} K0 \\ K1 \\ K2 \\ K3 \end{bmatrix} \times \begin{bmatrix} 14 & 11 & 13 & 19 \\ 9 & 14 & 11 & 13 \\ 13 & 9 & 14 & 11 \\ 11 & 13 & 9 & 14 \end{bmatrix} = \begin{bmatrix} J0 \\ J1 \\ J2 \\ J3 \end{bmatrix}$$

3. Inverse Row Shifting: Row shifting is similar to that done in the encrypting stage; the rows are cyclically shifted left in the fashion shown below.
   - Row first: Do not Move.
   - Row second: Moved once towards left.
   - Row third: Moved twice towards left.
   - Row fourth: Moved thrice towards left.
4. Inverse Byte Substitution: Again, similar to the encryption stage. S-box is used as a lookup table to substitute the corresponding bytes. It is worth noting that the same S-box is to be used for both encryption & decryption.

AES is one of the most secure encryption standards adopted by organizations around the world. It uses a symmetric key and is resistant to most of the known attacks, Provided a sufficiently long key is used. The AES algorithm is also integrated into the hardware of modern processors, making it very fast and easy to use.

- **Rivest-Shamir-Adleman (RSA):** RSA uses an asymmetric encryption & decryption process. It was introduced in 1977 [24] and relies on the computational limitation in factoring large prime numbers. It employs a key pair (public & private) for both encryption and decryption processes, facilitating secure key exchange and digital signatures. Although slower than symmetric algorithms,

RSA is widely used for secure communications and is integral to the security of protocols like TLS and SSH. Working of RSA can be divided into three basic stages:

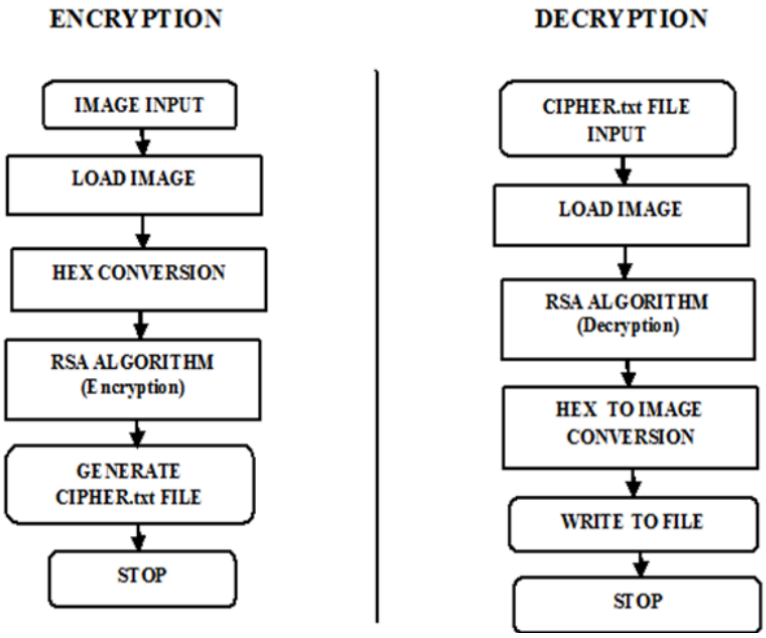



Figure 1.7: Flow chart for RSA: Encryption and Decryption process.

**Key Generation:** This is the first stage and can be described in the following steps:

- Selecting any two large Prime Numbers ($P_a$ and $P_b$).
- Calculate modulus ( (n) for both public & private keys) using
  $n = P_a \times P_b$. where n determines the length of the key.
- Compute Totient ($\phi(n)$) using
  $\phi(n) = (P_a - 1) \times (P_b - 1)$
- Determining public exponent ($e$), It is selected such that
  $1 < e < \phi(n)$ provided $e$ & $\phi(n)$ are coprime.
- Determining private exponent ($d$), It is calculated using
  $d \times e \equiv 1 \ (mod \phi(n))$
- Generate Public & Private keys, Keys are calculated as
  - Public Key: ($e$, n)
  - Private Key: ($d$, n)

**Encryption:** Encryption using the RSA algorithm can be explained below:

- Converting data (Plaintext) to integer ($I$), Provided $I$ shall be smaller than "n". In case the integer is larger the message is split into multiple smaller messages.

- Encrypting message, Ciphertext (c) is calculated using
  $c = I^e mod(n)$

**Decryption:** Decryption using the RSA algorithm can be explained below:

- Ciphertext (c) is Decrypted using
  $M = c^d mod(n)$, M is the plaintext & $d$ is the private key.

Since RSA is established on the mathematical complication of factoring large numbers, it is incredibly secure when using sufficiently large keys (e.g., 2048-bit or 4096-bit keys). Even with technological advances, systems are inefficient, and algorithms do not exist to factor large prime products in a feasible amount of time.

- **DSA (Digital Signature Algorithm):** NIST developed DSA in 1991, which is an asymmetric cryptographic algorithm primarily used for digital signatures [25]. It ensures data integrity and authenticity by allowing users to ascertain the origin and integrity of a message. DSA relies on the discrete logarithm problem for its security and offers efficient performance, particularly in signing and verification, making it suitable for secure communications and authentication systems.

- **ElGamal:** ElGamal is an asymmetric encryption algorithm introduced in 1985 [26], known for its reliance on the discrete logarithm problem. It provides digital signatures and encryption/decryption functions. ElGamal's encryption is probabilistic, meaning it produces different ciphertexts for the same plaintext, enhancing security. It forms the basis for several cryptosystems, including those providing homomorphic properties, making it valuable for research in secure multiparty computations.

- **Paillier:** Developed in 1999, the Paillier cryptosystem is an asymmetric algorithm known for its homomorphic encryption capabilities, enabling arithmetic operations on ciphertexts [27]. Its security is based on the decisional composite residuosity assumption. Paillier has practical applications in secure data processing, privacy-preserving computations, and cryptographic research, particularly in scenarios requiring computations without exposing the underlying data.

- **DES (Data Encryption Standard):** Is a standard used for encrypting and decrypting data using a symmetric key developed in 1975, works on 64-bit data segments with a 56-bit key [28]. It was widely adopted for secure data communication until vulnerabilities were discovered, primarily due to its small key length, rendering it susceptible to brute force attacks. Despite its deprecation, DES played a foundational role in developing modern encryption standards, including Triple DES and AES.

- **Triple DES (3DES):** Triple DES, developed as an extension of DES in 1998 [29], enhances security by applying the DES algorithm thrice to each data section, effectively increasing the key length to 112 or 168 bits. Although more secure than its predecessor, 3DES is slower and less efficient

compared to modern algorithms like AES. It remains in use primarily for legacy systems and applications requiring backward compatibility.

- **ECC (Elliptic Curve Cryptography):** ECC, introduced in 1985, leverages the algebraic structure of elliptic curves over finite fields for encryption, key exchange, and digital signatures [30]. It provides robust security with smaller key sizes compared to RSA, making it efficient for devices with limited computational power. ECC's potency lies in its resilience against specific cryptanalytic attacks, and it underpins protocols like ECDSA and ECDH, critical to secure communications.

- **Blowfish:** Blowfish is a symmetric key block cipher developed in 1993, featuring variable key lengths varying from 32 to 448 bits [31]. It is well known for speed, efficiency, and robustness against cryptanalysis, particularly in software implementations. Blowfish was widely adopted for secure communications but has largely been superseded by more advanced algorithms like AES. Its successor, Twofish, offers enhanced security and flexibility.

- **Twofish:** Twofish, introduced in 1998 [32] as a finalist in the AES competition, is a symmetric block cipher supporting key lengths of up to 256 bits. It was designed to offer high performance, flexibility, and strong resistance to cryptanalysis. Although not selected as the AES standard, Twofish is still regarded as a secure and efficient algorithm, especially for applications requiring open-source cryptographic solutions.

- **ChaCha20:** ChaCha20 is a symmetric stream cipher developed in 2008 by Daniel J. Bernstein [33]. It improves upon the earlier Salsa20 cipher, providing enhanced performance, security, and resistance to side-channel attacks. ChaCha20 is particularly efficient on mobile and embedded devices, making it an attractive alternative to AES for lightweight encryption. Its integration with Poly1305 for authentication has led to widespread adoption in protocols like TLS and VPNs.

### 1.6. Quantum Cryptography

Most modern cryptographic algorithms use some form of mathematical model that can be mimicked and replicated to break the message. The time and work needed to break the code depend on the complexity of the code and the computational resources available to the fraudster [16]. Quantum cryptography uses principles of quantum physics to secure communication. The algorithms are founded on the physical properties of the quantum particles, i.e., the uncertainty principle or concepts of monogamy of entanglement. They are said to be hack-proof. [35]–[39].

The history of Quantum Cryptography can be traced back to 1970 when Stephen Wiesner wrote conjugate code. He presented a notional idea of utilizing Quantum Cryptography to develop currency notes which would be impracticable to forge. He also presented the implementation of multiplexing channels. The same was further developed nearly 10 years later. When in 1979, Bennett and Gilles

| Algorithm | Type | Key Length (Bits) | Year of Development | Developer |
|---|---|---|---|---|
| DES [28] | Symmetric | 56 | 1975 | IBM |
| RSA [24] | Asymmetric | 1024, 2048, 3072, 4096 | 1977 | Ron Rivest, Adi Shamir, Leonard Adleman |
| ECC [30] | Asymmetric | 160, 224, 256, 384, 521 | 1985 | Victor S. Miller, Neal Koblitz |
| ElGamal [26] | Asymmetric | 1024, 2048, 3072 | 1985 | Taher Elgamal |
| DSA [34] | Asymmetric | 1024, 2048, 3072 | 1991 | National Institute of Standards and Technology (NIST) |
| Blowfish [31] | Symmetric | 32 to 448 (default 128) | 1993 | Bruce Schneier |
| Twofish [32] | Symmetric | Up to 256 | 1998 | Bruce Schneier |
| Triple DES [29] | Symmetric | 112, 168 | 1998 | IBM (improved version of DES) |
| Paillier [27] | Asymmetric | 2048 or higher | 1999 | Pascal Paillier |
| AES [23] | Symmetric | 128, 192, 256 | 2001 | Vincent Rijmen, Joan Daemen |
| ChaCha20 [33] | Symmetric | 256 | 2008 | Daniel J. Bernstein |

Table 1.2: Categorization of Encryption Algorithms

Brassard combined Wiesner's principles with public key cryptography, developing Crypto82. It was published under the names of Bennett, Brassard, Breidbart and Wiesner, this was the first paper to define term called "Quantum Cryptography" [7]. Initially, Quantum cryptography was considered a science friction as it is nearly impossible to demonstrate with the available technology [40]. Initially, Quantum cryptographic schemes using a one-time pad were introduced but rejected as their implementation was not achievable. The use of a one-time pad allows the key to be stored and reused [41].

Quantum key distribution is one of the most commonly used and trusted ways of sharing information in the quantum environment. However, it has a big drawback of distance limitation [42]. To limit this drawback, a series of quantum enabled repeaters are used to increase the range of transmission. One such application is the use of satellites as repeaters. Although this is still in the research stage, we have some promising results. QKD uses various quantum principles to ensure no information theft or that any part

of the message is available to eavesdroppers [24]. According to the Heisenberg Uncertainty principle, it is impossible to determine any system's quantum state without disturbing it. Hence, if any polarized photon is measured to ascertain its state, then its state will change. This plays a critical role in saving the data from the eavesdroppers. One major advantage of using QKD is that when an eavesdropper intercepts the signal, the sender and receiver are alerted. Also, it is known to the receiver how much information is leaked to the eavesdropper.

Several protocols have been developed over time to solve the QKD distribution problem. Some of the most prominent protocols are BB84, SARG04, and B92, among others.[43]–[45]. The most trusted and commonly used protocol was published by Bennett and Brassard in 1984 and is referred to as BB84.

### 1.7. Quantum Key Distribution

Quantum key distribution is one of the trusted & commonly used and trusted ways of sharing information in the quantum environment. It has a significant drawback of distance limitation [42]. To limit this drawback a series of quantum trusted repeaters can be used to extend the distance of communication. One such application is the use of satellites as a repeater. Although this is still in the research stage, we have some promising results.

QKD uses various quantum principles to ensure that no information theft or any part of the message is available to the eavesdroppers [42]. According to the Heisenberg's Uncertainty principle, it is practically impossible to determine any system's quantum state without disturbing it. Hence, if any polarized photon is measured to ascertain its state, then its state will change. This plays a critical role in saving the data from the eavesdroppers. One significant advantage of using Quantum Key Distribution (QKD) is that when an eavesdropper intercepts the signal, the sender and receiver are alerted. It is also known to the receiver how much information is leaked to the eavesdropper.

Several protocols have been developed over time to resolve the QKD distribution quandary. Some of the most prominent protocols are BB84, SARG04, and B92 etc. [44], [46], [47]. One of the most trusted and commonly used protocol that was published by Bennett and Brassard in 1984, commonly referred to as BB84. Quantum cryptography utilises various properties of quantum physics to send and receive information, mostly using quantum particles as carriers. The quantum particles can be photons, Quarks, Leptons, Bosons, etc., and the properties used can be Entanglement, Heisenberg, arbitrary states, etc. The most commonly used QKD protocols and their working principle can be listed in Table.1.4.

#### 1.7.1. BB84 Algorithm

BB84 algorithm is designed on the following principles:

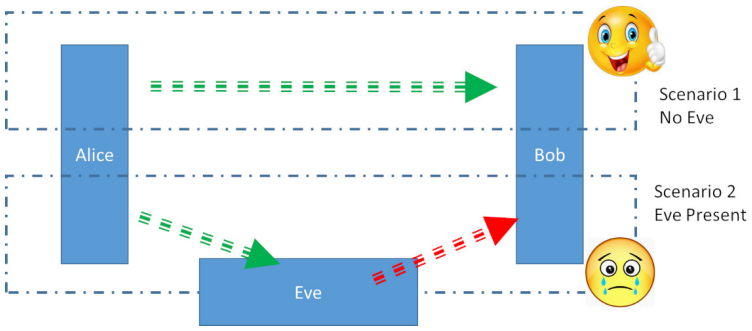



Figure 1.8: Communication without Eve and with Eve.

1. The Quantum state of any particle cannot be measured without disturbing it. It means the particle's quantum state is destroyed once the measurement is made. This implies that if any measurement is made by any eavesdropper, the original state of the particle will change and can be easily known to the sender or receiver.

2. The measurement, once done, cannot be undone. The changes which occur on the Quantum particle due to the measurement cannot be corrected [48].

3. Quantum states cannot be cloned, Quantum states are so random in nature that one cannot induce a desired quantum state on particles. This helps in keeping the message or data secured. If the eavesdropper happens to intercept the message, then the message is destroyed, and she cannot reproduce the same quantum state on the particles to fool the receiver. This makes it easy for the receiver to know what portion of the message is leaked and can be removed from the quantum Key.

In scenario 1, when there is no Eve, the sender and receiver will have Key which is an exact match Figure.1.8.

In scenario 2, the message sent by Alice is intercepted by Eve; now Bob will not receive any message because once Eve makes measurements on the message, the quantum state of the message is destroyed and cannot be reread. This way, Bob will not receive any message and will know the presence of Eve Figure.1.8. Now, if Eve becomes smart and starts transmitting her Qubits to Bob to fool him. According to the non-cloning property of the Quantum particles, she will never be able to replicate the original message. Hence, the quantum bits she will send Bob will be a series of random Qubits. Now, Bob will have some keys, but they will not be the same as Alice. Thereby making them aware of the presence of Eve.

**Working of BB84 Protocol**

BB84 protocol [49] can be detailed in the following fragments:

1. Alice (Sender) prepares a single photon (or any other quantum particle) which is polarized (or

encrypted) on some random basis and sends it across to Bob through a quantum channel. Once received by Bob, he makes his measurements on a random basis without knowing what base Alice used to use encryption.

2. If the basis of measurement (although random) chosen by Alice and Bob is identical, then the message is transferred, or if the basis does not match, then Bob receives a blank.

3. As the set of basis can be decided between Alice and Bob, i.e. it could be agreed that they will choose randomly only out of "horizontal vertical polarization (+)" or "diagonal polarization (X)". The probability of Bob receiving a message or Blank as an output of measurement is 50:50.

4. Once Bob receives the string message, he sends the list of positions (or basis) of the blanks he received. Alice will also remove the Qubits for which Bob does not have measurements. Ultimately, they will have the same key generated between themselves.

5. Once the Key is generated. Alice and Bob compare a small fraction of the Key to check for inaccuracies. If no mismatch is found, the Key is accepted, but if it finds some error, then either it is corrected or the Key is dispose of, and a new Key is generated. This step is called privacy amplification.

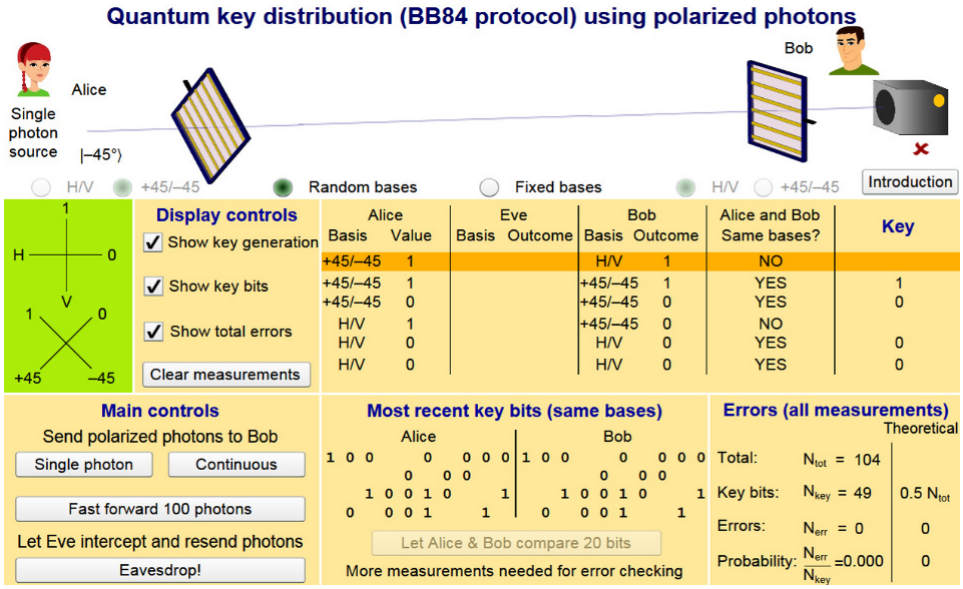



Figure 1.9: Example for BB84 protocol using polarized photons.

The protocol can be explained with an example, as demonstrated in Figure.1.9. Alice and Bob use polarized photons to communicate. The polarized photons can be polarized in many possible ways. Still, for easy understanding, it is chosen that photons are polarized either in Horizontal Vertical or Positive diagonal negative diagonal polarizations.

Possible states can be:

$$|H > corresponding\ to\ say, 0 \tag{1.1}$$

$$|V > corresponding\ to\ say, 1 \tag{1.2}$$

$$| + 45 > corresponding\ to\ say, 0 \tag{1.3}$$

$$| - 45 > corresponding\ to\ say, 1 \tag{1.4}$$

To measure at any location $HV$ or ±45 basis is used. If the $HV$ polarized photon is measured using ±45, the results will be blank. The result will be acceptable when the polarization method of the proton and the measurement basis is the same.

$$|H > -- > HV\ (basis) - - > 0 \tag{1.5}$$

$$|V > -- > HV\ (basis) - - > 1 \tag{1.6}$$

$$| - 45 > -- > HV\ (basis) - - > 0 \tag{1.7}$$

$$| + 45 > -- > HV\ (basis) - - > 1 \tag{1.8}$$

If Alice uses a ±45 and sends a value of 1 or $|V >$ photon to Bob and Bob uses $HV$ basis to do his measurement, then he will receive "blank" as output. In the second attempt, Alice sends the same photon to Bob, but this time he uses a ±45 basis to perform his measurement. Now he will receive output as "1". Similarly, a number of photons can be sent and received to establish a key, i.e. QKD.

### 1.7.2. B92 Algorithm

In principle, it is analogous to the BB84 protocol, it is further refined and simplified version. It was proposed by Charles H. Bennett in the year 1992 [49]. He proved that the QKD can be established using only two polarization states or Quantum states without compromising security. In B92 protocol, Alice (sender) will polarize the photons either in +45° or 0°, and the values assigned to them are "1" and "0", respectively. Bob (receiver) is equipped with a polarization analyser and single photon detector. Bob can either measure the incoming photon in Vertical (+90°) or diagonal (−45°). Assuming Alice sends photon polarized in +45° (Value 1), now if Bob is measuring using −45° polarization, then he will not be able to detect the photon and will not generate any key. Similarly, if Bob is measuring using a +90°

polarization, he has a 0.5 probability of detecting the photon. Hence, the output could be 1 or there could be no detection. Going forward, it can be seen that there are only six possible scenarios as shown in Table.1.3

Table 1.3: B92 QKD Protocols.

| Alice | | | Bob | | | |
|---|---|---|---|---|---|---|
| Polarization | Value | Key | Polarization | Detection | Value | Key |
| +45° | 1 | 1 | −45° | No | - | - |
| +45° | 1 | 1 | +90° | Yes | 1 | 1 |
| +45° | 1 | 1 | +90° | No | - | - |
| 0° | 0 | 0 | +90° | No | - | - |
| 0° | 0 | 0 | −45° | Yes | 0 | 0 |
| 0° | 0 | 0 | −45° | No | - | - |

Once Bob detects a sufficient Photons, Alice & Bob compare the positions of the identified bits, and Alice discards the remaining bits, thereby arriving at a common key. To detect the error, a small portion of the Key is shared publicly and compared for errors. If the errors are above a limit, the generated Key is disposed, and a fresh Key is generated.

### 1.7.3. E91 Algorithm

It was proposed by Artur Ekert in 1991 hence, the name E91 [50]. This protocol is based on the principle of Quantum Entanglement. In the E91 protocol, an entangled pair of photons is generated either by Bob, Alice or any third party. One photon of the entangled pair is sent to sender and the other to receiver (refer to Figure. 1.10). According to the principles of quantum physics, any change in the state of first photon will effect the state second photon. Alice and Bob both make their independent random measurements. The outcome of the measurement is random but still are correlated to each other. This correlation can be used to synthesize the key. The level of correlation can be checked to ascertain the presence of Eve in the system. The level of correlation decreases when under Eve's attack.

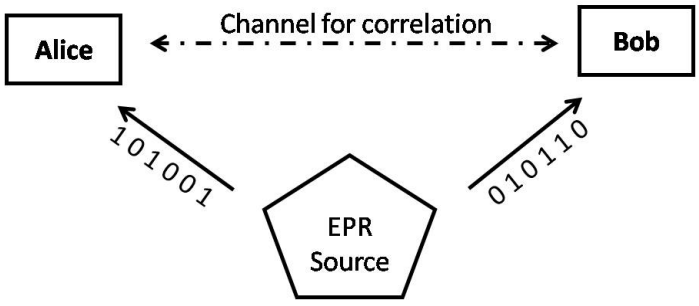



Figure 1.10: Working for E91 protocol using entangled particles.

### 1.7.4. DPS Algorithm

It was first proposed by H. Takesue, et. al. while working with NTT Basic Research Laboratories. This protocol is primarily focused on PNS (Photon Number Split) attacks. It proposes to encode data onto a pulse train rather than using a SPS (Single Photon Source) [51]. Basic working of the algorithm can be recapitulated as:

1. Alice (sender) will generate a pulse train (with known time intervals). She then modulates the phase of the individual pulse using a random sequence of "0" and "$\pi$".
2. Pulse train is further attenuated such that the average number of photons in each pulse is less than one photon.
3. Alice sends the pulse train to Bob. Bob receives the pulse and uses a one-bit delay interferometer. The time delay of the interferometer is equal to the pulse interval.
4. The interferometer is phase adjusted so that when the phase difference between the successive pulses is "0" the photons are emitted from port A and port B when the difference is "$\pi$".
5. Bob makes a note of the timings of each click of the detector at both ports. Bob later sends this information to Alice using a public channel.
6. Knowing which detector clicked when Alice and Bob can synthesize a common key by assigning "0" to a phase shift of "0" and "1" to a phase shift of "$\pi$".

   This communication is secure in a scenario when Eve is present and uses a PNS attack on it as the data is encoded in the form of pulse difference. Hence, this protocol is secure in the event of a PNS attack.

### 1.7.5. SARG04 Algorithm

SARG04 algorithm is based on the BB84 and can be executed using B92 protocol also. It was published by Scarani et al. in 2004 in Physical Review Letters [52]. It was specifically developed to improve the security of commonly used protocols against PNS attacks. The first stage of preparing and sending is exactly the same as in BB84 or B92. The difference is in the second stage when Bob and Alice need to publicly announce their basis of choice. At this stage, if Eve has a copy of the photons that Alice has been transmitting to Bob, then she can also have access to the full Key.

To comprehend the workings of the SARG04 protocol, we must understand the principle of PNS attack. This we have explained in detail in the security and threats section of this paper. Just for brief, it can be understood as an attack where Eve takes advantage of the practical limitations of producing a single photon (a must requirement for the BB84 and B92 protocols) in the absence of a single photon

generator. Alice uses as a weak attenuated laser as a photon source. This produces a weak signal with very few photons (around 2 4 per pulse). Now if Eve is equipped with a Photon splitter she can split the signal and keep one photon in her photon memory and let other photons pass to Bob. And once Bob and Alice compare their basis publically then Eve can also use the same basis to generate the Key.

To counter this theft of information, Scarani et al. proposes a protocol in which Alice and Bob shouldn't compare their bases but instead announce a set of orthogonal bases, one of which she had used to encode the photon. This way, only Bob can know which basis Alice has chosen to create the photon polarization because he knows from the previous measurements which basis will yield results. On the other hand, Eve does not have the previous measurements and will not be able to guess the correct basis for the measurement. Hence, Alice and Bob can establish QKD in Eve's presence without her actually publishing her basis on a public channel. It was proved by Scarani et al. That SARG04 deliver better QBER in comparison to BB84 protocol and is more secure for the PNS attack in all the possible scenarios [52].

### 1.7.6. Coherent one-way Quantum Key Distribution Algorithm

Proposed by Damien Stucki et al. in the year 2005, it uses a phase-encoded photon beam. This protocol was developed specifically to address the PNS attack [53] as shown in Fig.1.11. It uses a series of weak coherent pulses generated by a CW-laser along with an intensity filter. The photons are encoded in phase with phase angles of "0" and "$\pi$" along with some decoy pulses to check for the presence of Eve.

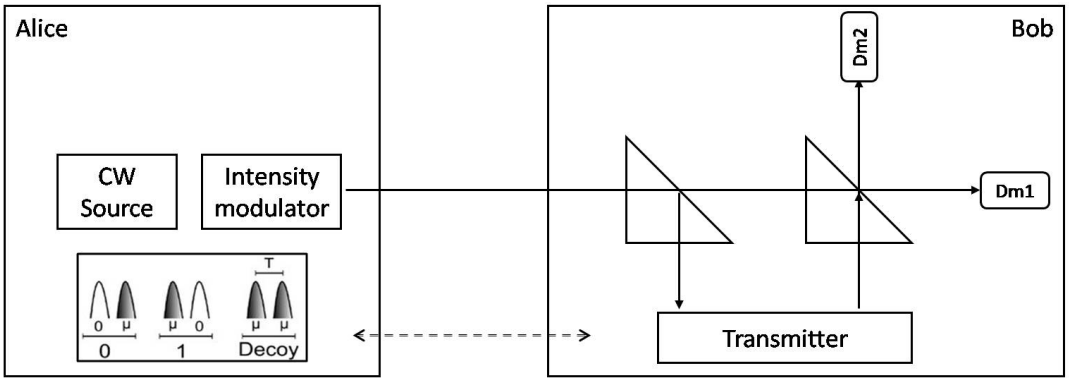



Figure 1.11: Schematic of COW QKD protocol.

The working can be simplified in the following stages:

- Alice prepares a continuous stream of pulses with some decoy sequence (probability f) and actual signal, which is encoded in the phase change of "0" and "$\pi$" (probability (1-f)/2).
- Bob receives the transmission and performs his measurements.
- Bob declares the results of his measurements. Stating which he got the detection in the data line

and for which bit he got results in Dm2.

- A Sifting process is followed to remove the extra bits from Bob's output based on the decoy pulses generated by Alice. Alice tells Bob the positions of the decoy pulses so as to enable him to delete the unwanted bits from the output.
- By analysing the data from Dm2 Alice can calculate the errors, ascertaining the presence of Eve during the communication process [53].
- Finally a process of error-correction & privacy-amplification is applied to generate the final Key.

### 1.7.7. KMB09 Algorithm

KMB09 QKD protocol was illustrated by M. Khan, Murphy and Beige in the year 2009. Hence, the name KMB09 [54]. KMB09 protocol was specifically developed to counter Eve's intercept and resend attack. In quantum communication, whenever a signal is sent, some error is generated even when there is no eavesdropper. These errors can be called system errors. Now, in the presence of Eve, the total error is increased. The author proposes to separately identify these errors as QBER and ITER. KMB09 protocol is a modified quantum key distribution protocol where Alice and Bob use two orthogonal bases to encode photons. The KMB09 protocol focuses more on ITER and proposes that the higher-dimensional photon states will yield significantly more ITER in the presence of Eve, making Eve detection very easy. This also allows the QKD to be established at longer distances.

### 1.7.8. QKD Algorithm with Private Public Key

Developed by E. Esteban and H. Serna and presented in the year 2012 [55], They proposed an algorithm which, unlike BB84 and its variants, works with a public and a private key. The key distribution is classified into three phases:

1. **Preparation phase (Alice):** In this, Alice prepares qubits using a string of random bits and further encodes them with a random basis. Alice sends her encoded qubits to Bob.
2. **Preparation phase key-Message (Bob):** Bob uses his own string of bits to encode the received message from Alice and generates a new set of qubits. Once made Bob sends his qubits to Alice.
3. **Measurement and derivation phase (Alice):** Alice now knows her initial string, which she used to generate her initial qubits, can decode the message sent by Bob.

This protocol can be generalized in a way that Bob can use his Private Key to communicate with multiple partners at the same time.

### 1.7.9. AK15

This was presented by A. Abushgra and K. Elleithy in 2015 [44]. AK15 protocol, as shown in Fig.1.12 [56], uses Heisenberg's uncertainty principle and formulation of matrices to transfer a message from one party to another. In this, the sender (Alice) generates a matrix of known dimensions.

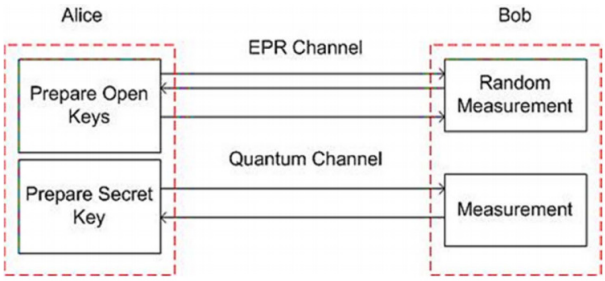



Figure 1.12: Schematic of AK15 protocol.

The matrix is formulated using two triangles. The upper left triangle corresponds to a set of random qubits and the lower right triangle comprises the secret message which needs to be sent. Working of protocol can be explain in the following steps:

- Alice initiates the communication through the EPR channel.
- Alice fills the matrix using the message and random qubits.
- She then sends the message to Bob using the EPR channel in a defined time.
- Bob receives the matrix and measures the incoming data. As the size is defined he can formulate the matrix. He then checks for the parity in the received data.
- If the parity is higher than the threshold, then the received message is accepted, and if less than the message is rejected, the process is repeated again.

### 1.7.10. Differential Phase Time Shift (DPTS) Algorithm

The protocol developed by D. Bacco et. al. is based on encoding data in phase and time shift [57], as shown in Figure.1.13. It can be explained in the following steps:

- Alice prepares a string of random states using random bits and encodes them in phase shifts (randomly either 0 or $\pi$). Alice then sends this string to Bob.
- Bob receives the photon into one of his detectors and he reveals the time and instances of his detection over the public channel.
- Alice reveals which set of measurements Bob needs to discard.

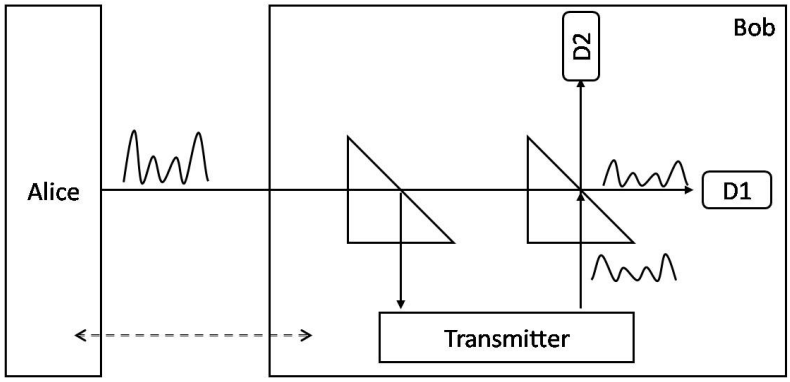



Figure 1.13: DPTS schematic representation.

- After estimating the QBER, Alice(Sender) and Bob (receiver) performs an error-correction and privacy-amplification to generate the Key.

### 1.7.11. MKP16 Algorithm

MKP16 algorithm was proposed by M. Kalra and R.C. Poonia [58]. It can be considered as a modification of the BB84 algorithm. The modification helps in increasing the key-generation rate by almost two times. In this algorithm as illustrated in Fig.1.14 both sender (Alice) and receiver (Bob) generates their own polarized photons using random basis. Now, both Alice and Bob send their polarized photons to each other and measure the received photons. Both measure the received photons according to basis, which are selected randomly. The process of coding and decoding is similar to BB84, with the difference that all the functions of sending and receiving are done by both Alice and Bob.

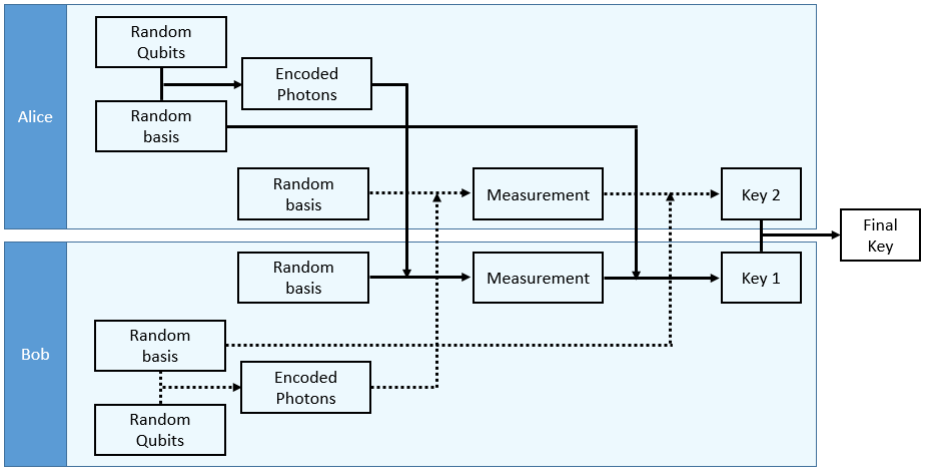



Figure 1.14: Structure of MKP16 protocol.

Two keys (Key1 and Key2) are generated, one with Alice as the sender and one with Bob as the sender. These two keys are then combined to form the final key, which is twice the key length.

MK16 protocol simulations were able to generate keys twice the length in comparison to the BB84 algorithm.

### 1.7.12. Decoy State Algorithm

Decoy state algorithm was developed by W.Y. Hwang [59] in 2003. The basic idea was to introduce defining decoy states in quantum communication to detect the involvement of Eve. In most of the QKD protocols a single photon source is used to generate the initial photons which will be further encoded and used for generating the Key. But due to lack of technology most of the times a weak coherent laser is used as photon source. Ideally, each pulse of the laser should have one or no photon, but many times, it has two or more. In the case where the pulse has more than one photon there are chances of data breach using PNS (photon umber split) attack. This attack is explained in detail in the further chapters. By adding decoy pulses Alice & Bob can test the quantum channel for attacks while QKD is in progress. The basic idea is if Eve is present it will attack decoy pulses also. This attack will inevitably alter the transmittance & quantum bit error rate of decoy pulse (as the expected error due to channel is known) [60].

### 1.7.13. Six State Protocol (SSP)

The six-state protocol was illustrated in "Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography" by Pasquinucci and Nicolas Gisin in 1999 [61].

The six-state protocol is primarily similar to the 4 state BB84 protocol, but it utilizes 6 states in place of four, as shown in Fig.1.15. This helps in making it easier for Bob and Alice to check for the presence of Eve, and it also makes Eve's job more difficult. Eve now has to guess the correct basis out of three possible bases. This reduces its ability to intercept and decipher the key.

| | | Photon number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Without Eve | Alice | Random Bits | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | | Random basis | B | C | C | A | A | C | B | C | C | B |
| | | Encoded photon | B0 | C1 | C1 | A1 | A1 | C0 | B0 | C0 | C0 | B0 |
| | | Channel | | | | | | | | | | |
| | Bob | Random Basis | B | C | C | C | B | C | C | C | A | B |
| | | Results | Yes | Yes | Yes | No | No | Yes | No | Yes | No | Yes |
| | | Decoded photon | 0 | 1 | 1 | | | 0 | | 0 | | 0 |
| | | KEY | 0 | 1 | 1 | | | 0 | | 0 | | 0 |

Figure 1.15: Six State Protocol working

Here, the error created by Eve is higher and can be easily detected. But having said that, by adding another two states the output at Bob's end also decreases and the key length which was 50% of the original number of photons sent by Alice (without the presence of Eve) also reduces to 33% as the probability of choosing the correct basis is 1 out of three in the ideal case.

### 1.7.14. Twin field Quantum Key Distribution (TF-QKD)

The twin field quantum key distribution (TF-QKD) protocol enables long-distance key generation. It helps to develop a countrywide network that uses quantum mechanics to encode and decode messages [62]. TF-QKD use a mid-node (commonly named Charlie) to establish key distribution between two remote parties (Alice & Bob). Mid-node is used to conduct measurements on the photons sent from the sender & receiver to determine the correlation between the measurements. With the addition of mid-node, Zhiliang was able to demonstrate Quantum Key Distribution over 1000km [62].

The working of TF-QKD can be explained in the following points:

1. **Preparation stage:** Alice & Bob choose non-orthogonal bases and encode data over them.
2. **Transmission stage:** Both Alice & Bob sent their encoded photons to mid-node (Charlie).
3. **Measurement Stage:** Charlie performs a joint measurement on both nodes using four Bell states.
4. **Post-Processing stage:** Similar to classical QKD post-processing, Alice & Bob discuss the outcome of the measurements done with Charlie.
5. **Synthesizing stage:** If the level of correlation is high, the results of the measurement are accepted as the secured key.

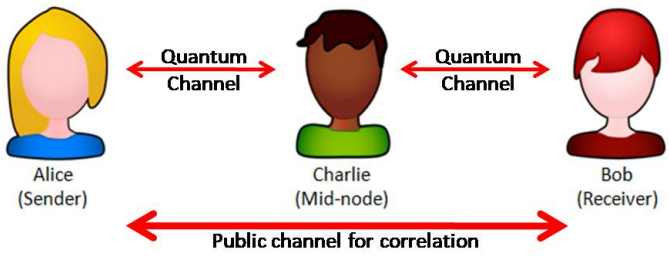



Figure 1.16: TF-QKD working concept.

TF-QKD had experimentally demonstrated the potential to overcome the distance limit of traditional QKD systems. As quantum physics eliminates the applicability of amplifiers, the distance that an encoded photon can travel is limited. By adding a mid-node, the requirement of photon travel is reduced to half. Thereby extending the total distance between Alice & Bob.

### 1.7.15. Asynchronous Measurement Device Independent QKD (AMDI-QKD)

AMDI-QKD shall be studied in comparison to or as an upgrade of TF-QKD. Its conceptual formation is similar to TF-QKD, with the major difference being that AMDI-QKD doesn't require any phase tracking or phase locking system [63]. Similar to TF-QKD, it uses a central node, Charlie, to conduct bell

measurement on the two photons received from Alice & Bob, respectively. The major difference is that the measurement is an asynchronous measurement that is realized after the two interference detection events have been matched. It was highlighted that the phase evolution between the consecutive time bins is very small. Hence, Phase tracking & Phase locking can be avoided & the phase-matched time bins can be post-matched with only a marginal increase in interference error. The basic configuration of the AMDI-QKD is shown in the Figure. 1.17.

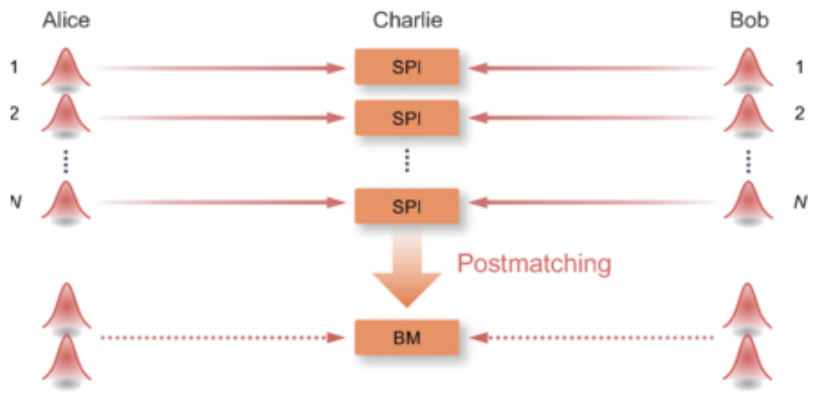



Figure 1.17: AMDI-QKD working concept.

Multiple time bins containing pairs of optical pulses are sent from Aice and Bob to Charlie. Each time bin is subjected to a single photon interference with a possibility of successful detection as $n \approx N\sqrt{\eta}$. The bins with successful measurements are later post-matched using time multiplexing to establish two-photon entangled states.

Working can be simplified in the following steps:

1. ***Preparation stage:*** Alice and Bob both prepare a series of photons, which are sent to Charlie in multiple time bins. Each bin must contain only one single photon pair.

2. ***Measurement stage:*** For each received time bin, Charlie conducts an interference measurement and publicly announces the position of the time bin and also makes public which detector had clicked.

3. ***Sifting stage:*** Once the measurements are done, they are compared for consistency or alignment. The time bins are compared (as shown in figure 1.18) for synchronization of the detection timing. Results are classified into two cases:

   (a) **Case 1:** When the results from both Alice & Bob have a detection event within a small time range of $T_c$.

   (b) **Case 2:** When there is no detection event within the range $T_C$ of another event.

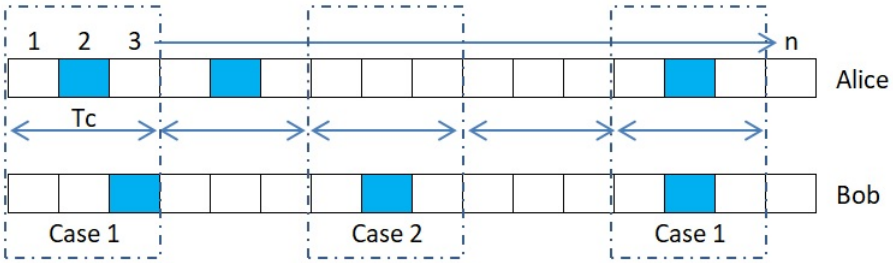



Figure 1.18: Time bin comparison to find Case 1 or Case 2

Both Alice & Bob count the number of Case 1 events if they are above a predetermined threshold limit. The protocol is aborted and restarted from the beginning. If the Case 1 events are below the limit, all the Case 1 events are discarded, and Case 2 events are used for further processing.

4. ***Parameter estimation stage:*** this includes the estimation of following parameters:

   (a) *Raw Key Bits:* It is constructed by randomly selecting the bits from the Z basis.

   (b) *Bit Error Rate (Ez):* The bits which are not selected in the raw key are used to estimate the error rate.

   (c) *Errors in X Basis:* The total number of errors is calculated by revealing all bit values in the X basis

   (d) *Decoy-State Method:* The decoy-state method estimates the number of vacuum events, single-photon pairs, the bit error rate in the X basis, and the phase error rate of single-photon pairs in the Z basis.

5. ***Post-processing stage:*** Alice & Bob estimate the final key with the help of an error correction algorithm with $\epsilon_{cor}$-correct further privacy amplification is achieved by $\epsilon_{sec}$-secret [63].

] The asynchronous AMDI-QKD protocol simplifies standard TF-QKD quantum key distribution by eliminating phase tracking and locking. This helps achieve transmission distances of up to 450 km [63].

### 1.7.16. Categorising Quantum Key Distribution

The wide landscape of QKD can be dissected vertically & Horizontally. Commonly used categorizations is by using the basic principle or by the Type of variables used. Some of the prominently used protocols are categorized in the table no. 1.4

1. Based on the Quantum principle applied:

(a) QKD based on Principle of Entanglement: These QKD algorithms commonly have a "Sender", a "Receiver", and an "Entangled-Photon Source". The entangled photons can be generated by sender, or receiver, or any other trusted third party. The basic idea is an entangled photon pair is generated and sent to the sender & receiver. Further, the key generation uses entanglement distillation protocols (EDP) [64]. Entangled Photon Pairs are prepared using one of the four Bell states,

$$\begin{aligned} |\psi_{00}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \\ |\psi_{10}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \\ |\psi_{11}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \\ |\psi_{01}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \end{aligned} \tag{1.9}$$

One half of the pair is sent to Sender & second to Receiver. Both use quantum memories to save their individual photons. The stored photons are later distilled using EDP protocols to filter the bell states. Quantum key is generated by performing measurements with Z basis. E91 & BBM92 are some of the oldest and widely studied entanglement based algorithms.

(b) QKD based on Principle of Uncertainty & No-cloning of non-orthogonal quantum states: These were conceptualised before Entanglement based QKD protocols. These typically includes a well defined prepare& measurement stages. Herein, the data is encoded using multiple quantum states of the partial. The selection of basis is done randomly. These encoded photons are then send to the receiver, where measurement operations are done on them (using random bases). The results of the measurements are reconciled and a common key is derived. This key is further used for communication. BB84, B92, SSP are some of the examples.

2. Based on the variable or output type:

(a) Discrete-Variable type: Discrete variable Quantum Key Distribution (QKD) protocols, such as BB84 and B92, utilize discrete quantum states (e.g., polarized photons) to encode and decode cryptographic keys. These protocols exploit quantum mechanics' principles, including superposition, entanglement, and no-cloning, to ensure secure key generation within two parties. Any attempt by an eavesdropper (Eve) to measure the quantum states will introduce errors, allowing the communicating parties (Alice and Bob) to detect the presence of Eve and abort the key exchange. This ensures the shared key remains private and secure, enabling secure data transfer over an anxious channel.

(b) Continuous-Variable type: It utilize continuous-variable quantum states, such as quadrature amplitudes of coherent light, to encode and decode cryptographic keys. Unlike discrete-

variable QKD, CV-QKD operates on a continuous spectrum, enabling higher key exchange rates and longer transmission distances. CV-QKD protocols, like Gaussian Modulated Coherent States (GMCS) and Coherent State (CS), employ techniques like homodyne or heterodyne detection to measure the quadrature amplitudes. These protocols are more resilient to channel noise and offer higher tolerance to losses, making them suitable for practical implementations. CV-QKD also enables the use of cost-effective and off-the-shelf optical components, simplifying the setup and reducing costs. Additionally, CV-QKD protocols can be more resistant to certain types of attacks, such as side-channel attacks. Overall, CV-QKD presents an advantageous approach to secure key exchange, with potential applications in high-speed, secure communication networks and quantum-secured data centres.

| Protocol | Type | Quantum Principles | Year | Scientist(s) |
|---|---|---|---|---|
| BB84 [49] | Discrete | No-cloning, Uncertainty | 1984 | Bennett, Brassard |
| B92 [65] | Discrete | Uncertainty, No-measurement | 1992 | Bennett |
| Ekert91 | Discrete | Entanglement, Bell's theorem | 1991 | Ekert [66] |
| Six-state [67] | Discrete | Uncertainty, No-cloning | 2000 | Bruss |
| SARG04 [52] | Discrete | Uncertainty, No-cloning | 2004 | Scarani, Acín |
| DPS [68] | Discrete | Decoy states | 2005 | Lo, Ma |
| COW [69] | Discrete | Entanglement, Bell's theorem | 2007 | Branciard, et al. |
| MDI QKD [70] | Discrete | Entanglement, Bell's theorem | 2012 | Lo, et al. |
| Twin-Field QKD [62] | Discrete | Entanglement, Bell's theorem | 2018 | Lucamarini, et al. |
| GMCS [71] | Continuous | Quadrature amplitude | 2002 | Grosshans, Grangier |
| CS [72] | Continuous | Coherent states | 2003 | Ralph |
| CV-BB84 [73] | Continuous | Quadrature amplitude | 2004 | Weedbrook, et al. |

Table 1.4: Prominently used QKD protocols, categorized according to the quantum principle used.

### 1.8. Mathematical Security Proof of BB84 and B92 QKD Protocols

Quantum Key Distribution (QKD) provides information-theoretic security grounded in the principles of quantum mechanics [74], [75]. This subsection presents mathematical sketches of unconditional security

proofs for the two most fundamental QKD schemes — BB84 and B92 — followed by a brief discussion on how such proofs can be extended to multi-path SDN-integrated protocols such as the Novel-QKD proposed in this work.

### 1.8.1. Preliminaries and Security Definition

Let $A$ (Alice) and $B$ (Bob) denote the legitimate communicating parties, and $E$ (Eve) denote the eavesdropper. Security is established when the final shared key $K$ is statistically independent of Eve's quantum information [76]. Formally, a key is said to be $\varepsilon$-secure if

$$\frac{1}{2}\|\rho_{KE} - \tau_K \otimes \rho_E\|_1 \leq \varepsilon, \tag{1.10}$$

where $\rho_{KE}$ represents the joint state of the final key and Eve's system, $\tau_K$ is the ideal uniform key state, and $\varepsilon$ is an arbitrarily small positive number.

The final secure key length $\ell$ after error correction and privacy amplification is bounded as

$$\ell \geq H_{\min}^{\varepsilon'}(A|E) - \text{leak}_{\text{EC}} - 2\log_2 \frac{1}{\varepsilon''}, \tag{1.11}$$

where $H_{\min}^{\varepsilon'}(A|E)$ is the smooth min-entropy of Alice's key conditioned on Eve's knowledge and $\text{leak}_{\text{EC}}$ is the information leaked during error correction. This forms the basis of Renner's composable security framework [76], [77].

### 1.8.2. Security Proof of the BB84 Protocol

The BB84 protocol uses four quantum states $\{|0\rangle, |1\rangle, |+\rangle, |-\rangle\}$, representing two conjugate bases: $Z = \{|0\rangle, |1\rangle\}$ and $X = \{|+\rangle, |-\rangle\}$ [66]. Bob randomly measures each received photon in one of these bases. After basis reconciliation, only matching-basis bits are kept, forming the *sifted key*.

**Step 1: Reduction to Entanglement-Based Form.** The prepare-and-measure BB84 protocol can be shown to be equivalent to an entanglement-based scheme where Alice prepares $n$ Bell states $|\Phi^+\rangle^{\otimes n}$ and measures half of each pair in the chosen basis. This reduction allows the use of entanglement purification arguments to prove security [78], [79].

**Step 2: Entanglement Purification and CSS Codes.** Shor and Preskill demonstrated that a CSS code-based entanglement purification protocol, capable of correcting bit and phase errors independently, is equivalent to classical error correction and privacy amplification in the BB84 protocol [78]. If the observed bit error rate is $e_b$ and the estimated phase error rate is $e_p$, then the asymptotic key generation

rate is given by

$$r = 1 - h(e_b) - h(e_p), \tag{1.12}$$

where $h(p)$ denotes the binary entropy function. Under the symmetry assumption $e_p \approx e_b$, the rate simplifies to

$$r = 1 - 2h(e_b). \tag{1.13}$$

This asymptotic relation forms the basis for many practical QKD security analyses [38], [74].

**Step 3: Phase Error Estimation.** Although the phase error $e_p$ cannot be measured directly, it can be bounded from the bit error statistics in the complementary basis. This estimation allows the derivation of composable security even under general attacks, using the de Finetti or postselection techniques [76], [80].

**Step 4: Privacy Amplification.** After classical error correction (which leaks $\text{leak}_{\text{EC}}$ bits to Eve), privacy amplification using universal hash functions ensures that Eve's information is exponentially small in $\ell$ [81], [82].

**Result.** For a typical one-way post-processing scenario, a positive key rate exists when

$$1 - 2h(e_b) > 0 \quad \Rightarrow \quad e_b \lesssim 11\%. \tag{1.14}$$

Hence, BB84 remains provably secure as long as the observed quantum bit error rate (QBER) remains below this threshold [78], [83].

### 1.8.3. Security Proof of the B92 Protocol

The B92 protocol uses only two non-orthogonal states $\{|\psi_0\rangle, |\psi_1\rangle\}$ with overlap $\langle\psi_0|\psi_1\rangle = \cos\theta$ [65]. The non-orthogonality ensures that Eve cannot perfectly discriminate the states, providing quantum security. However, this also makes the analysis more complex, requiring a reduction via local filtering.

**Step 1: Local Filtering and Entanglement Picture.** Tamaki, Koashi, and Imoto proposed a local filtering operation $F$ at Bob's side that probabilistically maps the two nonorthogonal states to orthogonal basis states when successful [84]. Conditioned on successful filtering, the protocol becomes equivalent to an entanglement-based scheme suitable for entanglement purification analysis.

**Step 2: Phase Error Estimation.** Let $e_b$ denote the observed bit error rate after filtering. Then the phase error $e_p$ can be bounded as a function of the overlap $\theta$ and $e_b$, i.e.,

$$e_p \leq f(\theta, e_b), \tag{1.15}$$

where $f(\theta, e_b)$ is derived from the joint statistics of the filtered measurement outcomes [76], [84]. This allows one to compute a conservative estimate of the privacy amplification term.

**Step 3: Key Rate Expression.** The asymptotic key rate per successfully filtered signal is given by

$$r = 1 - h(e_b) - h(e_p(\theta, e_b)). \tag{1.16}$$

A secure key can be extracted provided that $r > 0$. The optimal value of $\theta$ balances distinguishability (improving detection) and nonorthogonality (preserving quantum uncertainty) [83].

### 1.8.4. Comparative Remarks

Both BB84 and B92 achieve unconditional security by ensuring that any eavesdropping attempt introduces detectable disturbances in the quantum channel [74], [75]. While BB84 uses conjugate bases to randomize the measurement outcome, B92 relies on the fundamental indistinguishability of nonorthogonal states. BB84 tolerates higher noise, whereas B92 provides conceptual simplicity at the cost of higher loss sensitivity [65], [84].

### 1.8.5. Extension Toward Multi-Path SDN-Integrated Protocols

In the Novel-QKD protocol developed in this research, multiple simultaneous quantum key generation processes occur across distinct network paths managed by an SDN controller. Each path $i$ yields a key segment $K_i$ of length $\ell_i$, derived from its own bit error rate $e_b^{(i)}$ and phase error estimate $e_p^{(i)}$. By composable security [76], [85],

$$\ell_{\text{total}} = \sum_i H_{\min}^{\varepsilon_i'}(K_i|E) - \text{leak}_{\text{total}} - 2\log_2 \frac{1}{\varepsilon''}, \tag{1.17}$$

and the total security parameter satisfies

$$\varepsilon_{\text{total}} \leq \sum_i \varepsilon_i + \varepsilon_{\text{PA}}. \tag{1.18}$$

This composable framework allows the assimilation of multiple partial keys into a single secure composite key, thereby increasing total key length, reliability, and overall resilience to path-specific

failures or noise [86]. Hence, the Novel-QKD protocol generalizes the BB84 security structure to parallel SDN-controlled quantum channels.

## 1.9. Quantum Repeaters

Quantum repeaters can be considered middle nodes, which are placed somewhere between the sender and receiver, which helps in quantum communication. They are required to do two basic operations:

- Reduce the length of the quantum channel by splitting it into two or more sections.
- Improving the reliability of quantum network security by performing privacy amplification functions.

Depending on the algorithms used, Quantum repeaters can be broadly classified as:

- **Secured node:** (wherein it is assumed that eavesdroppers don't have access to it) is used. These are the simplest types of quantum repeaters. They use concepts like One Time Pad to transfer data from one network to another, thereby linking two quantum networks. In networks using these QR, the safety of the data depends on the guarantee of the sender, receiver, and QR [83], [87].
- **Untrusted nodes:**(wherein it is considered to be managed by eavesdroppers themselves). These are the next level of secured networks, which are also called end-to-end trusted networks, as their security depends only on the two ends (sender and receiver). The repeaters can be managed by anyone or even the adversary [70], [88]–[90].

### 1.9.1. Need of Quantum Repeaters.

As the name suggests, the Quantum repeater is the device used to repeat or relay the Quantum data from one network section to another section. The need for the QR can be understood by the following example.

Consider that the sender, Alice, wants to send some secure data to her friend Bob, who is located at some distance X from her (refer to 1.19).

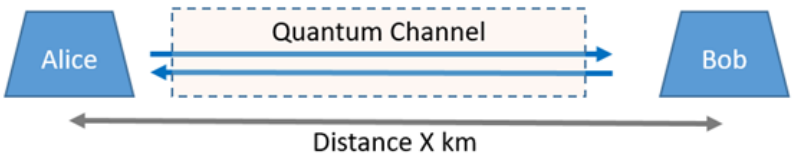



Figure 1.19: Quantum communication without any repeater.

Now, the security and the amount of data that can be communicated between Alice and Bob depends upon various factors like the noise level in the channel, the presence of Eavesdropper and most importantly, the distance between them. It has been observed that theoretically, there is no limit for the entanglement to happen, but when considering the practical aspect, there is a limit of only a few km. According to the series of experiments, it was concluded that the key generation rate of QKD decreases with the increase in transmission distance [91]. In general, the transmission distance can be calculated using the expression [91], as shown in figure 1.20.

$$d_{eff} = d_0 + (10 + \alpha) log_{10}(\frac{\mu}{\mu_{202}})$$

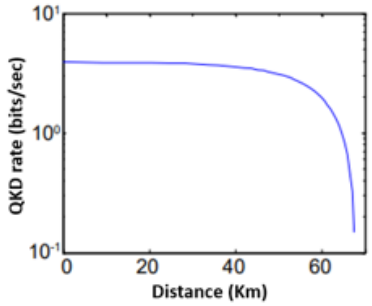



Figure 1.20: Key generation rate of repeater-less QKD vs Max distance.

To counter this limitation, the Quantum Repeaters are proposed. In this scenario, the data is first communicated between Alice and QR and then between QR and Bob and vice versa. Hence, conceptually, the distance of communication can be increased from "X" to "2X" (refer to 1.21).


Figure 1.21: Quantum key generation using a single repeater.

One can imagine the chain of such repeaters to form a communication channel which can encompass a metropolitan or country-wide network.

### 1.9.2. Capacity of the Quantum Communication network

One of the most important advantages of using QR is to have performance headway in terms of the volume of the data that can be sent over the network or simply the capacity improvement of the channel. A series of experimental studies were carried out from 2009 to recent times. The results finally show that the maximum possible data communication rate that can be achieved over a lossy channel is given by

$$C_{lossy}(\eta) = -\log_2(1-\eta) \tag{1.19}$$

*Where: $\eta$ is the transmissivity of the channel.*

Even with the most powerful protocol, this limit can't be breached. This limit is termed as the Pirandola Laurenza Ottaviani Banchi (PLOB) bound [92].

The capacity of the network can be enhanced by adding Quantum Repeaters [93].

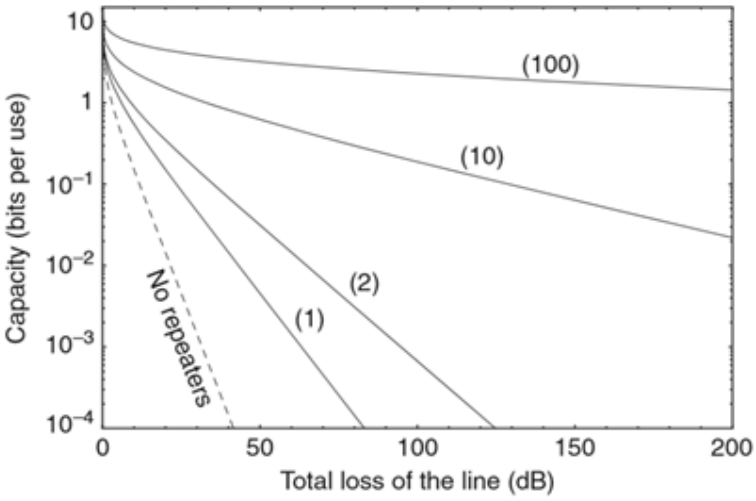



Figure 1.22: Capacity of quantum communication network with multiple repeaters.
The dotted line in the graph represents the loss vs bit rate if the communication is done without a repeater, and the solid lines represent the communication with (1), (2), (10), and (100) repeaters.

It can be seen clearly that with the addition of repeaters, we have considerable capacity improvement, and more lossy channels can be used for communication. Hence making Quantum communication more practical.Quantum repeaters can be arranged between Alice and Bob in various configurations. Commonly used configurations could be:

- **Linear configuration:** This is the simplest configuration in which the repeaters are arranged sequentially between Alice and Bob. The most efficient way is to place repeaters equidistant from each other.

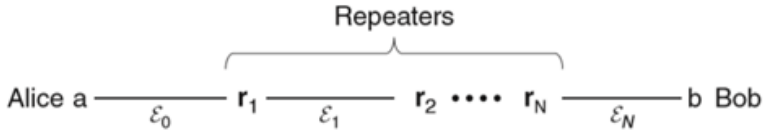



Figure 1.23: Linear configuration using multiple repeaters.

The capacity can be defined as [93]

$$C_{lossy}(\eta, N) = -\log_2(1 - \sqrt[N+1]{\eta}) \tag{1.20}$$

*Where: $\eta$ is the transmissivity of the channel*

*N is the number of repeaters*

- **Diamond configuration:** In this configuration, multiple paths are defined between Alice and Bob, enabling them to choose which channel to use based on availability.

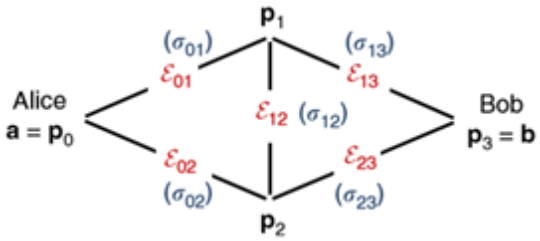



Figure 1.24: Diamond configuration using multiple repeaters.

In this configuration, if the communication happens between P0 & P1 or P0 & P2, the communication is similar to that of the linear configuration:

$$\epsilon_{01} \rightarrow \epsilon_{13} \text{ or } \epsilon_{02} \rightarrow \epsilon_{23}$$

When communication happens between P0 and P3, it can happen in multiple paths:

$$\epsilon_{01} \rightarrow \epsilon_{12} \rightarrow \epsilon_{23} \rightarrow \text{or } \epsilon_{02} \rightarrow \epsilon_{12} \rightarrow \epsilon_{13}$$

The capacity can be defined as [93]:

$$C^m(N_{lossy}) = -\log_2(L_N) \tag{1.21}$$

One more interesting conclusion can be derived when comparing the linear and diamond (parallel) configurations: It is found that the efficiency is increased two times. The equations, hence, can be written as:

$$C_{lossy}(\eta) = -\log_2(1-\eta) \tag{1.22}$$

$$C_{lossy}(\eta) = -2\log_2(1-\eta) \tag{1.23}$$

*Where: $\eta$ is the transmissivity of the channel*

As expected, the parallel configuration will have better performance and high channel capacity [93].

### 1.10. Quantum Simulators

Quantum computers are in the process of development, and the real-world quantum computer may not be accessible to all for the next decade. In addition, it is still far away from being as common as classical computers. Until scientists develop Quantum computers, developers can use simulators that can simulate many of the quantum problems. Quantum simulators help developers who are more interested in developing quantum algorithms and do not have access to Quantum computers. Quantum simulators can be categorized on the basis of their capacity and the class of problem-solving capabilities:

- Quantum simulators developed to address a specific problem.
- General purpose Quantum simulators are simulators which are more complex and are designed to solve or simulate a large range of quantum problems.

QISKIT is a simulator developed by IBM Research. It helps to provide an open-source arena for quantum computing. It has an extensive library of already-built Quantum codes and tools to help developers develop and test their programs. It has two basic components:

- A simulator to emulate the Quantum protocols on a classical computer.
- Direct access to IBM's quantum computers.

Users can develop their algorithms and test them on simulators, and later, they can run the codes on the actual Quantum computers. The primary aim of QISKIT is to encourage developers to develop codes for Quantum applications, thereby developing a large library of quantum cores that can be later used to develop complex algorithms. Currently, QISKIT supports only Superconducting trapped Ion quantum hardware built at IBM Quantum Labs. As QISKIT is a Python based programming language, it is easy to learn and implement. In the future, if required, the codes written in QISKIT can be used with other Quantum hardware. QISKIT can be installed from the parent program website "Quantum Experience" Users can use the online version "Quantum Composer" or can Download QISKIT for offline use. Quantum experience enables the user to generate a Key token, which is required to get access to the IBM quantum lab for the actual Quantum computer. Quantum experience also provides study material in the form of textbooks and tutorials. Launched in year 2016, Quantum Experience had a five-qubit quantum processor in a star-shaped configuration. Currently, it hosts 25 Qubit quantum processors. At that time, users can only use the online composer to develop the programs. Later on, QISKIT was developed so that users familiar with Python programming could write programs on Jupyter notebooks without the need to go online.

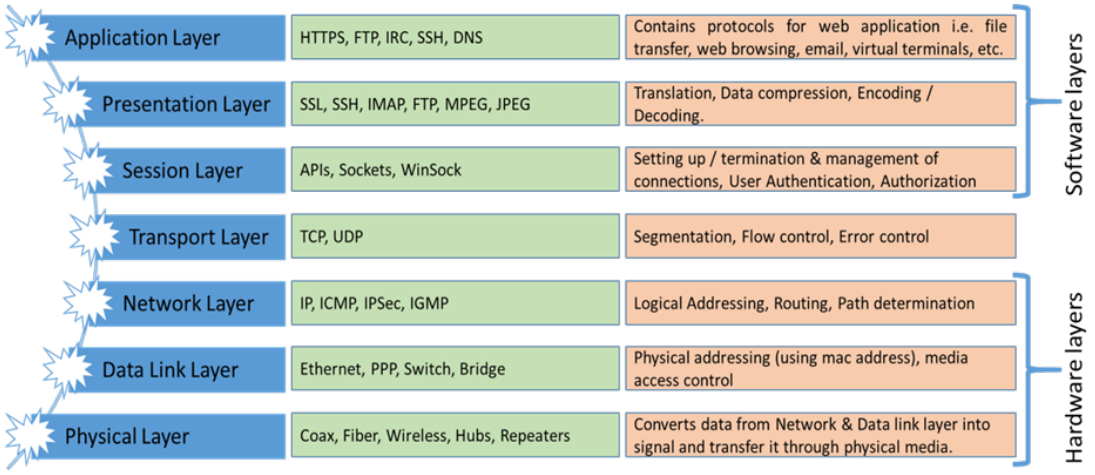



Figure 1.25: Layers of OSI model, protocols used & their basic functions.

### 1.11. Open System interconnection (OSI) model network

OSI can be considered as an architectural structure defining various components of a network. It was developed by ISO in the year 1984 to enable different applications to communicate with each other despite their inherent differences. It defines the network into seven layers (as shown in Figure.1.25),

The seven layers of the OSI model can be described as:

- **Application layer:** This layer encompasses all protocols required to develop user interface applications like web browsers, email clients, etc.
- **Presentation layer:** This layer performs functions like translation, data compression, encoding and decoding of the data from user applications and transfers it to the session layer.
- **Session layer:** This layer is accountable for completing connections, enabling sessions, user Authentication and authorization. The session layer, by virtue of its user authentication, controls the security of the data. The basic functions of the session layer can be listed as follows:
  - **Dialog control:** It controls the communication (dialogue) using half-duplex or full-duplex modes.
  - **Token management:** This helps in managing the available resources and avoiding request collision from multiple users. It helps in assigning tokens to the users so that they do not request the same operation simultaneously.
  - **Checkpointing:** It allows adding checkpoints for better session control and synchronization of available information.
  - **Opening and closing:** It controls the timely opening, closing and managing of the session between two ports.

The session layer is responsible for fetching data from the transport layer and sending data to the presentation layer. The session layer uses multiple protocols to perform its functions. Some of the prominent protocols are briefed below:

  - AppleTalk Data Stream Protocol (ADSP)
  - Real-Time Transport Control Protocol (RTCP)
  - Point-to-Point Tunnelling Protocol (PPTP)
  - Password Authentication Protocol (PAP)
  - Remote Procedure Call Protocol (RPCP)
  - Sockets Direct Protocol (SDP)

- **Transport layer:** The Transport layer has three basic functions: Segmentation, Flow control & Error control on the data sent to the hardware layers for transfer.
- **Network layer:** Functions like logical addressing, routing & path determination are done in the network layer.
- **Data link layer:** The data link layer performs functions like physical addressing and media access control.
- **Physical layer:** This is the base structural layer of the communication network; helps in converting the data into a signal and further sends it across the network channel.

### 1.12. Software Defined Networks (SDN)

Software-defined networks (SDN) can be considered a new approach to bringing more flexibility to networking. Protocols for traditional networks were written a few decades ago. They are effective, but the need to bring programmability into the architecture is very prominent [94]. GeoPlex [95] AT & T and Supranet Transaction Server [96] in the early 2000s were the first attempts to develop a network, which can be controlled by a software program. A significant advancement in the field of SDN came in the year 2011 when Open Network Foundation [97], a consortium of about 200 companies, formulated standards for SDN. The standard proposed by ONF was named Openflow, and it focuses on southbound transmission between the control layer and the physical network layer. Open stack, an open source software, was developed with a focus on management cloud architecture [98], [99] in the year 2010 by NASA & Rackspace Hosting [100].

SDN is a paradigm shift from a hardware-dependent networking system to a plug-and-play hardware (or service provider) self-regulating networking system. SDN (as shown in Figure.1.26) separate the hardware from the software, it allows development of the network applications (software) without having

the complete knowledge of the hardware. This helps by allowing the programmer to design applications depending upon the customer requirements using the API. These APIs allow the application layer to communicate with the control layer of the network services (as shown in Figure.1.26).

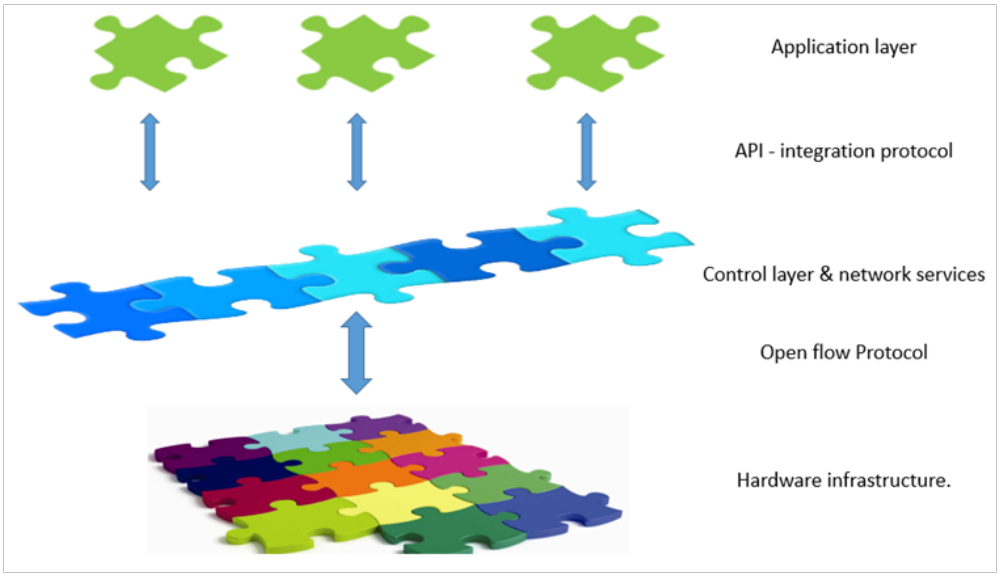



Figure 1.26: Structure of basic software-defined network.

SDN network can be considered to have three layers:

- **Application layer:** All the custom applications are developed in this layer, taking the help of APIs (Application Programming Interface) defined by the SDN network controller. This works as a separate software layer and is completely independent of the network hardware setup.
- **Control layer & Network services:** This is also a software layer and is placed typically on the server. It manages all the traffic through the network, user authentications, and network policies. It is connected to the Application layer using northbound application protocols and the network infrastructure layer through southbound application protocols.
- **Hardware infrastructure or Network layer:** As the name suggests this is the hardware layer. All the actual data transfer occurs on this layer. This layer comprises all the network cables, switches, etc. The hardware layer transfers the data packets between multiple networks according to the policies and addresses provided by the control layer.

## 1.13. Network Simulators

Designing & testing a new network configuration could be challenging and cost intensive task. Network simulators were developed as a test bench, that can be used to test newly developed network configurations & algorithms. Some of the prominent network simulators are detailed in the following sections.

### 1.13.1. Network Simulator (NS3)

NS-3 (Network Simulator 3) is an open source network simulator that simulates discrete events primarily used for research and education. It was developed to replace its predecessor, NS-2, and address some of the limitations observed in the earlier version. NS-3 was initiated by the University of Washington, in collaboration with other institutions and contributors, and its first release was made available in June 2008. The primary developers included Tom Henderson, George Riley, and Mathieu Lacage, who sought to create a more flexible and modular platform for network simulation [101], [102].

NS-3 was designed to offer a more pragmatic and scalable simulation environment than NS-2. It introduces a modular architecture and uses C++ as the core programming language, with Python bindings available for scripting. This design makes it easier to model complex network topologies and protocols. Since its launch in 2008, NS-3 has been continuously updated and improved by a global community of developers and researchers, making it a widely trusted tool in the networking field.

NS-3 enables a range of communication network protocols, including TCP/IP, Wi-Fi, LTE, and more. It enables users to simulate wired and wireless networks, making it ideal for studying various networking scenarios such as internet protocols, wireless sensor networks, and vehicular communications. The tool's modularity also allows for easy integration with external software, such as real-time network emulators.

One of the notable projects that have used NS-3 is the development and testing of the QUIC (Quick UDP Internet Connections) protocol. Researchers have also utilized NS-3 for work on IoT (Internet of Things) networks, 5G, and other emerging networking technologies [103].

NS-3 has been recognized for its flexibility, ease of use, and depth of network modeling it offers. It has been extensively adopted by researchers and academics, leading to numerous publications in prominent journals. The tool has also been the subject of various workshops, such as the annual "Workshop on NS-3," which gathers the community to discuss developments and share knowledge [104].

### 1.13.2. PyCryptodome

PyCryptodome is a self-contained Python library offering cryptographic functions, which include symmetric key algorithms (e.g., AES, DES), public key algorithms (e.g., RSA, DSA), and hash functions (e.g., SHA-256, MD5). It is designed to be a drop-in replacement for the outdated PyCrypto library, adding enhancements like faster encryption, authenticated encryption modes, and elliptic curve cryptography. PyCryptodome supports modern cipher modes and provides tools for key generation, encryption, decryption, and cryptographic signatures. The library is actively maintained by developer Legrandin (aka Dario Lombardo). The library was last updated on October 2, 2024, with version 3.21.0. PyCryptodome is actively maintained, making it suitable for modern cryptographic implementations [105]–[108].

### 1.13.3. Mininet Network Emulator

Mininet was initially developed by Bob Lantz, Brandon Heller, and Nick McKeown as part of the research into SDN and OpenFlow. It was designed to bridge the gap between simulation and deployment, allowing SDN controllers to interact with a virtual network in a similar way that, they would interact within a physical network. Mininet can emulate hundreds of network devices, enabling users to design complex topologies and test them with the same SDN controllers they would deploy in a real environment. Since its inception, it has become one of the most widely used tools for SDN research and education.

The first official version of Mininet, version 1.0, was released on October 2010. Since then, it has seen various updates, each adding new features, improving performance, and increasing compatibility with different SDN controllers. Notable projects that have been developed or tested using Mininet include the Open Networking Lab's ONOS (Open Network Operating System) and the OpenDaylight platform. These projects have contributed significantly to the development of the SDN ecosystem and have benefited from Mininet's ability to test real-world scenarios without the need for dedicated hardware.

In the development space, Mininet has been pivotal for testing SDN controllers and networking protocols. Projects such as Floodlight, Ryu, and Open vSwitch (OVS) have used Mininet extensively to develop and refine their solutions. These controllers, which are key components of the SDN ecosystem, are accountable for overseeing data flow across the network. Mininet provides the perfect platform for developers to test their algorithms, optimize performance, and ensure compatibility with various network devices and topologies.

One of Mininet's key advantages is its simplicity and flexibility. Users can create custom topologies using Python scripts, which allows for automated testing and integration with other software tools. It also supports integration with container-based technologies like Docker, enabling hybrid setups where virtual machines, containers, and Mininet networks coexist.

Mininet's influence extends beyond research and development. It has also played a role in the advancement of network functions virtualization (NFV) by enabling the emulation of network functions in a virtual environment. This allows companies and researchers to test NFV solutions before deploying them in production, saving time and resources.

Mininet has become a cornerstone in the world of SDN due to its ability to emulate realistic network environments, its ease of use, and its cost-effectiveness. It has facilitated the development of key projects in the SDN space and continues to be a vital tool in education and research. The ongoing contributions from the community ensure that Mininet remains up-to-date and relevant in an ever-evolving networking landscape [109]–[111].

## 1.14. Security threats on Quantum systems

Quantum systems are founded on the very principles of quantum physics and provide unsurpassed security. However, unconditional security is ensured only at the conceptual level as the implementation depends on the physical limitations. Taking an example of BB84 protocol, to ensure security the protocol demands a single photon source. However, in physical implementation, a single photon source is inaccessible; hence, the engineers resort to an attenuated laser for a photon source. Now, having multiple photons encoded with the same data can be a threat to security as Eve can use a photon split attack (explained in detail below) to gain information without generating any error. Here, we explain only a few of the most prominent security threats that Eve can exploit to gain knowledge or to threaten the workings of the Quantum communication system.

- **PNS attack:** PNS attack or Photon split [112] attack in its basic form utilized an implementation flow in the QKD. For all practical usage, the protocols cannot be implemented in the same form as it is described by the developer. One such issue is the requirement of a single photon source at the sender end. The photon generated by the single photon source is then encoded and sent to the receiver. If Eve tries to measure the encoded photon, it creates an error that the receiver can detect, and the existence of Eve is comprehended.

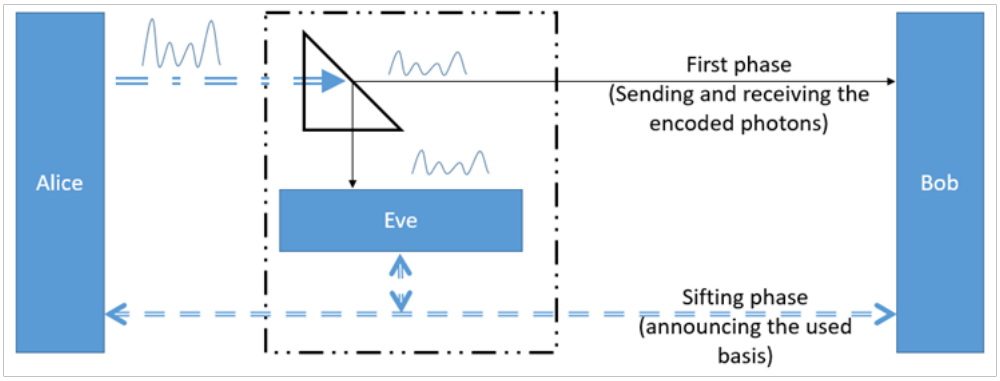



Figure 1.27: PNS attack.

Now, for practical purposes, a weak attenuated laser is used as a photon source. It generates about two more photons in every pulse [113]. The photons in the single pulse are then encoded and sent to the receiver without having any information about the number of photons sent (as shown in Figure.1.27). Eve enabled with the photon counter, can count the number of photons in the Quantum channel. If there are more than one photons, eavesdropper can use a Beam splitter to split the incoming pulse, store one set of photons in her photon memory, and send the other half of the beam to Bob [114]. Bob, without having any information about the number of photons sent by Alice, does his measurement and then shares his basis using public channels. Now, Eve, having a copy of all the photons can use the basic information available in the public channel to generate a

perfect copy of the key without revealing her presence.

- **Man in The Middle attack:** Man in the middle is a very simple and very effective attack. In this Eve (as shown in Figure.1.28) uses an identical machine as used by Alice & Bob. Eve places herself in the middle of the quantum channel such that any attempt of communication between Alice and Bob has to go through her [115].

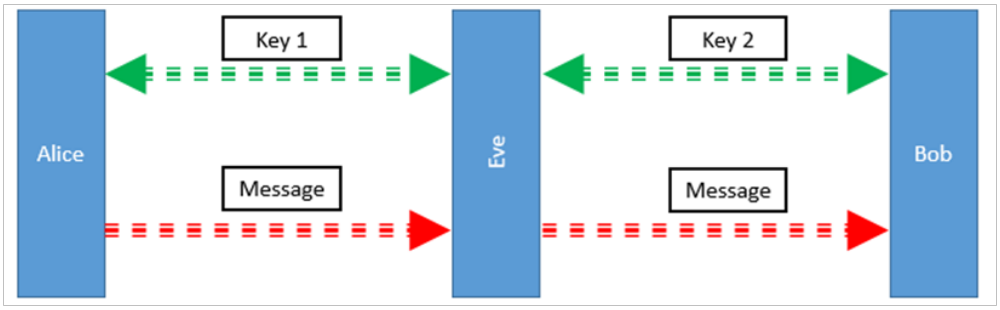



Figure 1.28: Man in middle attack.

When Alice tries to generate a QKD with Bob, Eve interrupts the communication, presents herself as Bob, and generates a QKD with Alice. At the same time, Eve pretends to be Alice and generates a QKD with Bob. This way, Eve generates two keys, one with each Alice & Bob. Now, when Alice transmits encrypted data to Bob, Eve stops that transmission and decodes the message with her key. Again, she encodes the message using her second key and sends it to Bob. This way, no one has any information about Eve, and she will be able to gain full access to the communication. This is a very serious attack as it cannot be countered by only Quantum communication setup and would require some pre-existing authentication for proper identification before beginning the QKD process.

- **Denial of service attack:** In this, the prime objective of Eve is not to gain information but to not let the QKD itself be established, thereby rendering the Quantum communication system useless. This can be done by disabling the hardware used or by introducing a high level of noise in the system such that the QKD never completes and is aborted. This is a very basic and primitive attack and similar to some very easy remedies. The attack can be implemented by simply obstructing the optical channel or by adding a high-intensity light source in the optical channel. To prevent this, multiple channels can be used as standby so that in case of this attack, Alice & Bob can switch to another channel to establish QKD.

## 1.15. Unique Contributions of the Research

The key novelty and scientific contributions of this research, which distinguish it from existing QKD studies, are summarized as follows:

1. **Development of an Adaptive SDN-Integrated Quantum Key Distribution Protocol (Novel-QKD):** A new QKD framework that enables parallel key generation through multiple quantum paths managed by an SDN controller, providing high adaptability and resilience.

2. **Multi-Path Key Aggregation Mechanism:** A mechanism to assimilate partial keys from different quantum paths into a single composite key, increasing total key length and maintaining security even when partial paths are unavailable or compromised.

3. **Enhanced Resilience to Quantum Attacks:** Demonstrated improved resistance against man-in-the-middle, photon-number-splitting, and denial-of-service attacks through SDN-based dynamic path selection and isolation of compromised links.

4. **Simulation-Based Validation Environment:** Designed and implemented a hybrid simulation platform combining Mininet (for network-level SDN control) and Qiskit (for quantum layer simulation), enabling reproducible testing of QKD algorithms.

5. **Performance Improvement:** Achieved a key-generation rate of 420 bits per second compared to 115 bits/s for BB84 and 74 bits/s for B92 under identical conditions, while maintaining a comparable bit-error rate ( 8.8%).

6. **Framework for Quantum-Safe Network Architectures:** Provided a scalable and adaptive communication framework applicable to both civilian and defence applications, establishing a foundation for future quantum-resilient communication systems.

These contributions collectively address major limitations of traditional QKD schemes by introducing adaptability, fault tolerance, and scalability into quantum communication systems through Software-Defined Networking.

### 1.16. Thesis Structure

To provide clarity and logical progression, the thesis is structured as follows:

Chapter 1 - **Introduction**: This chapter starts with research question and core hypothesis. Further, provides an introduction to the very basic concepts, founding principles and concepts that are prerequisite for the understanding of this thesis. These includes the details about concepts of modern day cryptography, quantum principles, quantum cryptography, simulators to name a few.Details of the most commonly used QKD protocols are presented along with their security proof. This chapter also elaborate about the unique contribution of the research.

Chapter 2 - **Literature review**: Reviews existing classical and quantum cryptographic techniques, including BB84, B92, and recent QKD–SDN integration efforts, identifying research gaps and motivating the

need for an adaptive protocol.Chapter presents the theoretical background and the present status of the research available associated to the subjected thesis.

Chapter 3 - **Motivation and research methodology**: Research gaps and real world case studies are presented in detail to motivate readers to conduct further research on the subject. Based on the identified research gaps we elaborate on our objective and research methodology used.

Chapter 4 - **To develop a novel quantum key distribution protocol and compare it with commonly used protocols**: ]Elaborates on developing Novel-QKD algorithms, working concepts and functionality. It also details the construction and testing of the Python-based simulation test bench. The testbed is tested by simulating BB84. Further, the performances of the Novel-QKD algorithm were benchmarked against BB84 and B92 with the help of the simulation setup.

Chapter 5 - **To implement the novel QKD protocol on the standard network**: Novel-QKD protocol was tested for network implementation. The effectiveness of the implementation is measured with a comparative study performed between Novel-QKD, BB84 and B92 algorithms. This chapter provides insight into the testing and simulation for implementing QKD algorithms on a standard network.

Chapter 6 - **To integrate newly developed QKD protocol on the software defined network**: Portrays the deployment of the Novel-QKD algorithm on the Software Defined network. We study the performance of Novel-QKD in comparison with BB84 and B92. We further extended the study to study the influence of eavesdroppers on performance parameters.

Chapter 7 - **Conclusion and summary**: Concludes the thesis and provides the detailed analysis of the results presented in the earlier chapters. We also touch upon the limitations of our presented work and the future scope.

Chapter 8 - **Future scope**: As quantum cryptography is a wast subject and there no limit of research that can be done. We limit our research to our objectives and scope of work. In this chapter we list out some of the limitation and future opportunities, that can take advantage of this thesis work.

# 2. LITERATURE REVIEW

Quantum communication, or Quantum cryptography, amalgamates quantum mechanics and information technology. It is a future technology that will require improvements in both Hardware and software disciplines. An essential network structure is needed for any practical implementation of a quantum communication network, and quantum key distribution algorithms are necessary to transmit data over that network securely. The current encryption system is very secure for the current network conditions, but in the future, once quantum computers are available, most of our current systems will be proven redundant. Quantum computers will still take some years to one decade to become commonly available to everyone. Until then, we must improve our network security to be resilient to quantum-enabled attacks on our encryption system. Multiple studies are being carried out to understand the behaviour of the quantum particle and to utilize them to build encryption systems capable of jettisoning any quantum-enabled attack.

## 2.1. Cryptography

Cryptography has always been an essential ingredient in making a secure communication network. It allows users to communicate through a public channel in full view of the eavesdropper. The data is encoded using a key, and even if the eavesdropper gets access to it, it is useless without the key. However, the intended receiver can decipher the transmission using the key. Based on the type of key used, a classical cryptographic system can either be a Symmetric or a Non-Symmetric Key. The most commonly used protocols are RSA, DES & AES encryption, all ensuring various levels of security.

Horst Feistel developed the **Data Encryption Standard (DES)** in the 1970s while working at IBM, which played a crucial role in the evolution of encryption technologies [116]. DES was subsequently adopted by the National Bureau of Standards (NBS) and the National Aeronautics and Space Administration (NASA). In 1977, it was officially designated as the federal standard for encrypting sensitive information in the United States. The introduction of DES marked a significant advancement in cryptography, providing a systematic and standardized approach to data encryption that influenced the design of subsequent encryption algorithms. However, over the years, the vulnerabilities of DES due to advances in computational power led to its eventual replacement by more secure algorithms, such as the Advanced Encryption Standard (AES).

DES was found to have numerous vulnerabilities over time, prompting the need for a more robust encryption standard. In 1997, efforts commenced to develop the **Advanced Encryption Standard (AES)** aimed at addressing the shortcomings of DES. After a comprehensive selection process involving several candidate algorithms, the U.S. government officially adopted AES as the encryption standard for securing sensitive information in 2001 [117]. AES is based on the Rijndael algorithm, developed by

Vincent Rijmen and Joan Daemen, and it has since become the encryption standard widely used across various applications, from securing government communications to online banking and data protection. Adopting AES marked a significant advancement in cryptographic practices, providing more robust security measures against increasingly sophisticated threats.

In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman developed the **Rivest Shamir Adleman (RSA)** algorithm, a cornerstone of modern cryptography widely used for secure data transfer. The algorithm is based on a system of public and private keys and relies on the mathematical difficulty of factoring the product of two large prime numbers. The fundamental principle behind RSA's security is the computational complexity involved in deriving these prime factors from their product, which is feasible for small numbers but becomes impractical as the size of the primes increases. The RSA algorithm has been pivotal in ensuring secure communications in various applications, including secure email, online transactions, and digital signatures. Its widespread adoption highlights its significance in the field of cryptography, establishing a foundation for numerous security protocols used today [24].

Daniel J. Bernstein's 2006 paper titled "Curve25519: New Diffie-Hellman Speed Records" discusses the performance and security features of this elliptic curve [118].Curve25519 represents a significant advancement in elliptic curve cryptography, particularly for key-generation protocols like Diffie-Hellman. The curve is defined by the equation $y^2 = x^3 + 486662x^2 + x$ and is designed to optimize performance and security, making it suitable for a wide range of cryptographic applications. The X25519 protocol, which employs Curve25519, is particularly noted for its efficiency and resilience against side-channel attacks, enhancing key-generation security. The implementation of Curve25519 has become widespread in various cryptographic systems and protocols, solidifying its status as a modern standard for secure communications. Its design emphasizes simplicity and speed while providing robust security features, making it ideal for high-performance computing and resource-constrained devices.

### 2.2. Quantum Cryptography and Quantum Key Distribution (QKD)

Quantum Key Distribution (QKD) is a secure communication method that uses the principles of quantum mechanics to enable two parties to share encryption keys in a way immune to eavesdropping. Unlike classical key-generation methods, QKD takes advantage of quantum principles such as superposition and entanglement, ensuring that any attempt to intercept the key alters the quantum states and can be immediately detected by the legitimate parties.

In 1991, Artur Ekert introduced the first entanglement-based quantum key distribution (QKD) protocol, which leveraged the unique properties of quantum entanglement to establish secure communication [66]. This protocol utilized Bell inequalities to evaluate the measurement outcomes, providing a means to detect any potential eavesdropping on the communication channel. Ekert's protocol demonstrated that violating Bell inequalities could serve as a foundation for secure key distribution, illustrating the practical

implications of quantum mechanics in cryptography. This work paved the way for further advancements in quantum communication technologies and remains a crucial milestone in developing QKD protocols, expanding the potential for secure communications in various applications.

Dagmar Bruß in 1998, discusses a generalization of the quantum cryptographic protocol initially proposed by Bennett and Brassard. This extended scheme involves using three conjugate bases, which results in a total of six states for quantum communication. By calculating the optimal mutual information between the sender and potential eavesdropper, the study demonstrates that this expanded scheme provides enhanced security against eavesdropping on individual qubits compared to the protocol based on two conjugate bases. Furthermore, the paper explores the relationship between the maximum classical correlation in a generalized Bell inequality and the intersection of mutual information between the sender and receiver and between the sender and the eavesdropper. This connection sheds light on the broader implications of the protocol and its potential for strengthening the security of quantum communication [67].

The Six State Protocol (SSP) was introduced in 1999 by Pasquinucci and Niclas Gisin to enhance the previously established QKD protocols [119]. Like the BB84 protocol, SSP utilizes quantum states for secure key distribution; however, it distinguishes itself by employing six states of encoded photons instead of the four states used in BB84. This increase in the number of states enhances the protocol's robustness against certain attacks. The design of the SSP allows for improved security measures and offers a more flexible framework for quantum communication, leveraging the principles of quantum mechanics to protect against eavesdropping. This protocol represents a significant advancement in quantum cryptography, contributing to the ongoing development of secure communication systems.

In 2003, W.-Y. Hwang proposed a novel decoy state idea to counteract the Photon Number Splitting (PNS) attack in quantum key distribution (QKD) systems [59]. This innovative approach involves testing the quantum channel during the QKD process, allowing the detection of potential eavesdropping attempts by monitoring variations in the signal states. The essence of the decoy state technique lies in the random variation of the signal intensity sent over the quantum channel. By introducing decoy states—randomly chosen weak signals alongside the primary signal—Hwang demonstrated that eavesdroppers could be identified through discrepancies in the expected detection rates. This method significantly enhances the security of QKD protocols, making it more resilient against PNS attacks, where an eavesdropper could exploit multiple photon emissions to gain information without detection. Hwang's work has substantially impacted the evolution of secure quantum communication know-how, advancing the field's understanding of vulnerabilities and the strategies needed to mitigate them.

In 2004, Scarani et al. introduced the SARG04 protocol, which builds on the foundational principles of the BB84 and B92 protocols, but with a significant modification aimed at enhancing security against Photon-Number-Splitting (PNS) attacks [52]. Unlike BB84, which compares measurement bases used by Alice and Bob, SARG04 employs a different strategy: Alice shares a set of orthogonal bases, using

one for encoding the photon while the others serve as the basis for measurement. This design choice enables SARG04 to mitigate vulnerabilities to PNS attacks by reducing the information available to an eavesdropper while ensuring secure key distribution between the parties. By not requiring a comparison of the bases used for measurement, the protocol enhances the overall security framework for quantum key distribution. The SARG04 protocol represents an essential advancement in quantum cryptography, contributing to ongoing efforts to develop robust secure communication methods.

Damien Stucki et al. developed a quantum key distribution algorithm specifically designed to counter Photon Number Splitting (PNS) attacks, as detailed in their work published in *New Journal of Physics* [120]. Their innovative approach incorporates a coherent weak laser and an intensity modulator as the photon source, allowing greater control over the emitted quantum states. In this protocol, the photons are encoded with phase angles of "0" and "$\phi$", combined with decoy pulses to monitor for the presence of an eavesdropper, often referred to as Eve. By utilizing this method, the researchers aimed to enhance the security of quantum key-distribution systems while maintaining adequate performance. Stucki et al.'s work contributes significantly to developing robust quantum communication technologies, providing a framework that improves resistance to various eavesdropping techniques, including PNS attacks.

In 2015, A. Abushgra and K. Elleithy presented the Ak15 protocol, which leverages Heisenberg's uncertainty principle and matrix formulations for secure message transfer between parties [121]. In this innovative protocol, the sender, Alice, generates a matrix of known dimensions that provides the foundation for communication. The matrix is structured using two triangular components: the upper left triangle is populated with random qubits, while the lower right triangle contains the secret message intended for transmission. This method utilizes quantum properties to ensure that the message is secure, as any attempt by an eavesdropper to access the qubits would disturb the quantum state, alerting Alice and the receiver to potential interception. This protocol represents a significant advancement in quantum cryptography, highlighting the practical applications of quantum mechanics in secure communications.

Marcos Curty et al. presented a modified version of the Measurement Device Independent Quantum Key Distribution (MDI-QKD) system to enhance the distance capabilities between end users [122]. This innovative approach incorporates a mid-node, which acts as a relay to facilitate longer communication distances while maintaining the security principles inherent in quantum key distribution. The authors also provide a comprehensive security proof for their QKD system by utilizing three protocols and systematically comparing their bit error rates. This analysis establishes the robustness of their modified MDI-QKD system and contributes to the broader understanding of the security landscape in quantum communications. Curty et al.'s work represents a significant step forward in the practical implementation of QKD systems, addressing one of the critical challenges in the field: the limitations imposed by distance. Their findings could have substantial implications for future developments in secure quantum communications, particularly in long-range applications.

The MKP16 algorithm, proposed by M. Kalra and R.C. Poonia, is regarded as a modification of the

widely used BB84 protocol in quantum key distribution (QKD) [123]. This modification aims to enhance the key distribution rate significantly, increasing by a two times when compared to the original BB84 scheme. The authors elaborate on how the adjustments made in the MKP16 algorithm facilitate this improvement while maintaining the core principles of quantum security. This advancement is significant in practical applications of QKD, as higher key generation rates can lead to more efficient and effective secure communications. Kalra and Poonia's work contributes to the ongoing evolution of QKD protocols, providing insights that can be applied to future developments in the field.

Peter Schiansky et al. proposed a groundbreaking quantum-based digital payment system in 2020 that offers security against even the most formidable computational attacks. This innovative system utilizes quantum light to generate unforgeable cryptograms, ensuring the cryptograms are unique and tamper-proof. The implementation of this system has been tested over an optical fiber connection, demonstrating its robustness against noise and loss-dependent attacks. Notably, this quantum payment system eliminates the need for quantum storage or dependency on trusted vendors and certified channels, making it attainable with near-future technology. This advancement signifies a potential shift towards quantum-enabled security for digital payment systems, which could significantly enhance the integrity and safety of financial transactions in the digital realm [124].

L. Zhou et al. presented a novel technique to stabilize an open channel without a closed interferometer, demonstrating its applicability to phase-sensitive quantum communication. Their innovative setup achieved a bit rate of 0.32 bits per second over an impressive distance of 615.6 km, showcasing the potential for long-distance quantum communication without the limitations imposed by traditional methods [125]. This advancement enhances the feasibility of quantum communication systems and contributes to the ongoing development of robust techniques that can operate effectively over long distances. Such innovations are crucial for the future of secure communication, paving the way for practical applications of quantum technologies.

Tagliavacche et al. (2025) [126] introduced a novel frequency-bin entanglement-based Quantum Key Distribution (QKD) protocol that leverages discrete frequency modes of entangled photon pairs to achieve high-dimensional quantum communication. Unlike conventional polarization- or time-bin-based QKD systems, their work exploits integrated photonic circuits to generate and manipulate frequency-entangled states, enabling parallel key generation across multiple frequency channels. The study presents an experimental demonstration over optical fiber links exceeding 50 km, achieving high-visibility interference and a secure key rate significantly greater than comparable time-bin implementations. By encoding qubits in frequency space, the protocol enhances spectral efficiency and allows seamless coexistence with classical data channels within standard telecommunication infrastructures. Tagliavacche et al. provided a rigorous entropic-uncertainty-based security analysis, confirming resilience against collective and coherent attacks even under channel dispersion and noise. Moreover, their system exhibits inherent compatibility with wavelength-division multiplexing (WDM) and coherent transceiver technologies, making it suitable for

hybrid classical-quantum network environments. Despite these advancements, the research primarily focuses on the photonic implementation and does not explore dynamic Software-Defined Networking (SDN) control or multi-path key routing, which limits adaptability at the network orchestration layer. Nevertheless, this 2025 study marks a significant advancement in high-capacity, entanglement-based QKD by demonstrating the practical feasibility of frequency-domain encoding. It provides a promising foundation for future integration with SDN-enabled control architectures, supporting the development of scalable, spectrum-efficient, and dynamically reconfigurable quantum communication networks.

### 2.3. Comparative summary of the quantum key distribution protocols

Table 2.1 presents a comparative summary of prominent Quantum Key Distribution (QKD) protocols that have shaped the evolution of quantum cryptography. Each protocol introduces unique operational principles, encoding strategies, and implementation trade-offs aimed at improving key generation efficiency, distance, and resistance to attacks. Early protocols such as BB84 and B92 laid the theoretical foundation for secure quantum communication but exhibit limitations in key-generation rate and adaptability. Subsequent developments, including entanglement-based (E91), decoy-state, differential-phase, and measurement-device-independent (MDI) approaches, sought to overcome these shortcomings by enhancing channel tolerance and mitigating specific side-channel vulnerabilities. More recent schemes, such as twin-field (TF-QKD) and asynchronous MDI-QKD, extend transmission distance and practicality but remain experimentally demanding. The table systematically highlights the strengths and limitations of each method, providing a clear view of the existing performance and security gaps that motivate the design of the proposed SDN-integrated multi-path Novel-QKD protocol developed in this research.

| Name | Author(s) | Year | Strengths | Limitations |
|---|---|---|---|---|
| BB84 | C. H. Bennett & G. Brassard | 1984 | Conceptually simple; strong unconditional security proofs (Shor–Preskill style); widely used benchmark; tolerant to moderate errors. | Key rate limited by channel loss and detector imperfections; vulnerable to some implementation side-channels (detector blinding) unless mitigations used; distance limited without trusted nodes or repeaters. |
| B92 | C. H. Bennett | 1992 | Uses only two nonorthogonal states — simpler state set; conceptually efficient in some implementations. | More sensitive to loss and inconclusive outcomes; vulnerable to unambiguous state discrimination and channel loss; requires careful device assumptions or local filtering for security proofs. |
| E91 (Entanglement-based) | A. K. Ekert | 1991 | Device-independent features based on Bell inequality; entanglement enables flexible network topologies and satellite links. | Harder experimental implementation (entanglement distribution); limited distance due to loss; entanglement distribution expensive (satellites, long fibers). |
| Six-State Protocol (SSP) | Pasquinucci & Gisin | 1999 | Uses three mutually unbiased bases (six states) — higher eavesdropper detectability; stronger security per signal. | Lower sifting efficiency (more basis mismatch); increased experimental complexity and state preparation requirements. |

*continued on next page*

Table 2.1 – *continued from previous page*

| **Name** | **Author(s)** | **Year** | **Strengths** | **Limitations** |
|---|---|---|---|---|
| SARG04 | V. Scarani et al. | 2004 | Improved robustness against photon-number-splitting (PNS) attacks in weak coherent implementations; compatible with practical sources. | Lower key rate compared to BB84 in some regimes; more complex sifting/announcement rules. |
| Decoy-State QKD | W.-Y. Hwang; extended by Lo, Ma, Chen & others | 2003 (Hwang) | Greatly improves security/performance of weak coherent-pulse implementations by detecting PNS attacks; enables practical long-distance QKD with lasers. | Requires precise intensity control and calibration of decoy states; added complexity in source modulation and parameter estimation. |
| DPS (Differential Phase Shift) | K. Inoue, E. Waks, Y. Yamamoto & others | 2002 | Simple receiver architecture (interferometer); naturally robust against some attacks; can use coherent sources. | Security proofs historically more complex; sensitive to phase stability; rate/distance trade-offs; specific attacks require careful countermeasures. |
| Coherent One-Way (COW) | D. Stucki et al. / ID Quantique (experimental groups) | mid-2000s | Simple transmitter and receiver design; attractive for high-speed fiber QKD; tolerant to detector dead-time. | Security proofs initially less rigorous than BB84; requires additional monitoring lines; vulnerable to some side-channel attacks unless monitored. |

*continued on next page*

Table 2.1 – *continued from previous page*

| Name | Author(s) | Year | Strengths | Limitations |
|---|---|---|---|---|
| MDI-QKD (Measurement-Device-Independent) | H.-K. Lo, M. Curty, B. Qi & others | 2012 | Eliminates detector-side channel attacks by moving measurement to an untrusted node; strong practical security for detectors. | Requires two-way interference (Bell-state measurement) and precise synchronization; generally lower key rate and more complex experimental setup; still sensitive to source imperfections. |
| TF-QKD (Twin-Field QKD) | M. Lucamarini et al. | 2018 | Overcomes linear rate–distance bound (improved key rate scaling with distance); enables longer-distance key generation without quantum repeaters. | Experimental implementation demanding: phase stabilization across long distances and high interference visibility; complexity in phase reference sharing and trusted-phase reference assumptions. |
| AMDI-QKD (Asynchronous MDI) | Various recent works | 2020s | Reduces strict synchronization requirement of MDI-QKD by allowing asynchronous detection events; improves practicality in network scenarios. | Emerging protocol class — practical implementations and comprehensive security analyses still under active research; may require additional calibration. |
| MKP16 | M. Kalra & R. C. Poonia | 2016 | Proposed modifications to BB84 aimed at increasing key generation rate (claimed 2× improvement). | Typically theoretical/simulation results; limited experimental validation and narrower threat-model analysis. |

*continued on next page*

Table 2.1 – *continued from previous page*

| **Name** | **Author(s)** | **Year** | **Strengths** | **Limitations** |
|---|---|---|---|---|
| AK15 | A. Abushgra & K. Elleithy | 2015 | Matrix-based protocol leveraging uncertainty principles (as proposed by authors); novel encoding/processing ideas. | Mostly theoretical; requires more extensive security and implementation analysis to be broadly adopted. |
| QKD with Public–Private Key | E. Esteban & H. Serna | 2012 | Attempts to combine classical asymmetric primitives with QKD for hybrid functionalities (e.g., authentication, infrastructure compatibility). | May re-introduce classical computational assumptions for some components (authentication/PKI); complexity in provable composable security if classical parts are not information-theoretic. |
| Twin-Field variants and repeaterless improvements | Multiple groups (variants after TF-QKD) | 2019–2022 | Improved rate–distance performance and practical variants for network deployment (e.g., sending-or-not TF). | Many variants are experimentally challenging; security proofs and assumptions vary by variant — must check per-version constraints. |

Table 2.1: Comparison of QKD Protocols — strengths and limitations

### 2.4. Quantum Cryptography network implementation

QKD ensures the generation of secure keys between sender & receiver. These keys can provide security for the data communicated between two partakers. Multiple experiments has been conducted to ensure the working of QKD alongside the classical network. Some of the prominent experiments are listed below.

T. E. Chapuran et al. [127] explored integrating optical networking techniques with current quantum key distribution (QKD), and quantum-enabled data transmissions. Their study emphasizes various networking architectures capable of meeting the specific requirements of quantum protocols. The authors highlight the potential of reconfigurable optical networks in enhancing the scalability and reliability of QKD systems, contributing significantly to the evolution of secure quantum communication techniques. This research underscores the importance of optical networking in advancing the practical implementation of quantum security technologies.

C. Z. Peng et al. experimentally demonstrated the free-space dispersal of entangled photon pairs over a distance of 13 km, utilizing a challenging environment that included atmospheric disturbances and background light interference [128]. The study focused on generating entangled photons through type-II parametric down-conversion, employing advanced techniques to optimize transmission efficiency with large telescopes and synchronization methods to ensure timing coincidence at the receivers. The results indicated that the visibility of the entangled state was maintained above the threshold essential for a breach of Bell's inequality, demonstrating high correlations between measurements taken by two spatially separated receivers. This work significantly contributes to quantum communication by showcasing the practical feasibility of distributing entangled states over long distances in a real-world setting.

T. Schmitt et al. demonstrated free-space Quantum Key Distribution (QKD) over a distance of 144 km, utilizing the BB84 protocol enhanced with decoy states to improve security and reduce vulnerabilities to eavesdropping [129]. This landmark experiment involved a complex setup designed to mitigate the effects of atmospheric turbulence and background noise, which can compromise the integrity of quantum signals over long distances. The results showcased the feasibility of long-distance QKD and established a new benchmark in quantum communication. By employing decoy states, the researchers effectively increased the system's resilience against photon number splitting attacks, a significant threat to QKD protocols. This work is pivotal as it advances the practical application of quantum cryptography in secure communication systems, pushing the boundaries of how far QKD can operate in real-world conditions.

J. Yin et al. presented experimental free-space quantum teleportation and entanglement distribution over a distance of 100 km, marking a significant milestone in quantum communication research [130]. The experiment utilized a sophisticated setup that involved sending entangled photon pairs through the atmosphere, demonstrating the functional utilization of quantum teleportation in real-world conditions. By effectively distributing entangled states, the researchers showcased the potential for long-distance

quantum communication and laid the groundwork for the future advances in quantum networks. Their findings confirm the feasibility of maintaining quantum correlations over large distances and highlight the teleportation protocol's robustness against environmental disturbances, which are critical for developing secure quantum communication systems. This work is a crucial step toward realizing large-scale quantum networks and has been cited in various studies exploring the applications of quantum entanglement in telecommunications and cryptography.

S. Nauerth et al. demonstrated a novel technique to quantum key distribution (QKD) by installing a secure communication connection between a ground station and a hot air balloon [131]. This pioneering experiment employed entangled photons transmitted from the balloon to the ground, showcasing the potential of aerial platforms to facilitate quantum communication in challenging environments. The research addressed critical elements of QKD, such as the effect of atmospheric constraints on the transmission of quantum states, and successfully demonstrated that robust key distribution is possible even under these variable circumstances. By leveraging the unique capabilities of hot air balloons, the study opens new avenues for deploying QKD systems, particularly in remote or inaccessible areas. This work advances the understanding of quantum communication techniques and paves the way for future applications, including secure communications in aerial and satellite-based networks.

J.P. Bourgoin et al. presented a groundbreaking experiment demonstrating quantum key distribution (QKD) between a moving ground station and a satellite in a downlink configuration [132]. This study highlights the feasibility of secure quantum communication in dynamic environments, specifically focusing on the challenges posed by the movement of both the ground station and the satellite. The research utilized a QKD protocol that effectively accounted for the relative motion between the two nodes, ensuring the integrity and security of the distributed keys despite the changing distances and angles. The results illustrated that high-quality entangled photon pairs could be exchanged, paving the way for future implementations of satellite-based QKD systems. This work represents a significant advancement in quantum communications, potentially leading to the development of global secure communication networks. The implications of this research are profound, as they contribute to the ongoing efforts to integrate quantum technologies into practical applications.

Hua Dong et al. [133] propose a novel approach to enhance the distance over which Quantum Key Distribution (QKD) can be established by implementing it over a wide area network (WAN). Their study compares traditional point-to-point QKD systems with those deployed across a WAN, emphasizing the advantages of increased range and flexibility in key-generation processes. The researchers delve into critical issues related to data security, such as potential leakages and the role of entrusted nodes within the network architecture. They highlight the implications of these factors on the overall security and efficiency of QKD systems, arguing that the WAN approach can mitigate particular vulnerabilities inherent in point-to-point configurations. This work contributes significantly to the ongoing discourse in quantum communications, suggesting that broader network frameworks may provide the necessary

infrastructure for secure communications over greater distances, thus facilitating practical applications in various sectors.

In 2013, S. Aleksic et al. explored the integration of Quantum Key Distribution (QKD) into optical access networks, highlighting its potential to enhance secure communications in existing infrastructures [134]. The study emphasizes the unique challenges posed by conventional optical signals and their impairments, which can impact the efficacy of QKD systems. By utilizing passive optical networks, the authors demonstrate how QKD can be effectively implemented to achieve secure key rates, thereby addressing the pressing need for robust security measures in telecommunications. This research not only contributes to the field of quantum cryptography but also signifies the practical applications of quantum technology in real-world network scenarios.

### 2.5. Quantum Cryptography implementation on Software Defined Networks (QKD-SDN)

Using SDN with QKD augments quantum network security, scalability, and flexibility. SDN provides centralized management, allowing dynamic routing and optimization of QKD key distribution, enhancing performance and resilience. It allows efficient monitoring & automated key management. By integrating QKD and SDN, organizations can enforce security policies, reduce operational costs, and seamlessly incorporate QKD with classical network infrastructure for broader adoption of quantum-safe communication.

A. Aguado et. al., implemented principles of SDN network and integrated them with the functionality of QKD. Main idea of this paper is to introduce the QKD systems to other network elements. So that, the SDN controller can be used to control the network behaviour. It was proposed that the implementation of QKD over SDN can be done at a slow and gradual bases without need to disrupt the working of the network [135].

David Elkouss, et al., sheds light on the distance limitation of point to point QKD network. To improve on this a series of trusted repeaters are used but the security of the repeaters cannot be assured. Author proposed a new QKD network model wherein weakly trusted repeaters can be used. It was shown that the eavesdropper had to invade and take control of multiple repeater nodes at same time to break the encryption, which is very unlikely. The model proposed shall be supported by network security codes to enable secure key exchange at a metropolitan scale network [136].

Omar Shirko & Shavan Askar stated that the current network implementation uses classic trusted relays (CTR). CTR are required to be fully trusted, but they are seldom so. Also some error is produced at every relay, this error keeps on accumulating causing failure of complete network. Authors proposed a new model deploying SDN over QKD, by using software-defined quantum key relay failure model (SDQKRF)model which is more reliable & has better performance [137].

P. Techateerawat, proposed a network built for based on the current quantum cryptography networks, that can be scaled to multi user domain with high security at low cost. The main objective of the study was to verify the algorithm with a prospective of data transmission, security & performance. further, to prove the practical applicability and real world implementation. [138].

Bob Lantz & Brain O'Connor Presents the application Mininet in developing and testing complex software defined network architectures. They demonstrated a simple and effective way to simulate heavyweight containers used in full system virtualization by using mininet's cluster mode with lightweight containers on standalone machines. This helps to create a more flexible and scalable testbench for testing software defined networks [139].

Mangla et al. [85] presented a comprehensive architecture for Quantum Large-Scale Networks (QLSN) that integrates Quantum Key Distribution (QKD) within Software-Defined Networking (SDN) and Network Function Virtualization (NFV) frameworks. Their work aims to address scalability, resource optimization, and adaptive orchestration challenges in expanding quantum networks beyond point-to-point connections. The authors developed a control-plane-aware architecture that employs SDN controllers to dynamically allocate quantum and classical resources, thereby minimizing latency in key distribution and enhancing end-to-end reliability. Simulation results demonstrate a notable reduction in controller load imbalance and improved throughput across multi-domain quantum communication environments. Furthermore, the QLSN model emphasizes compatibility between heterogeneous QKD devices through a virtualized management layer, enabling inter-vendor operability and fault-tolerant configuration. Mangla et al. also discussed the importance of modular orchestration policies for automated quantum channel setup and teardown, essential for real-time key refresh operations in large-scale infrastructures. However, their study remains predominantly architectural, lacking detailed physical-layer or photonic-hardware integration analysis. Despite this limitation, the proposed QLSN framework provides a robust foundation for future SDN-orchestrated QKD deployments, demonstrating the feasibility of large-scale, software-controlled quantum key infrastructures capable of supporting metropolitan and cross-domain networks. The study significantly contributes to bridging the operational gap between quantum communication theory and deployable, software-managed secure network systems.

Zahidy et al. [140] demonstrated a high-dimensional Quantum Key Distribution (QKD) protocol implemented over a deployed multicore optical fiber, marking an important milestone in the practical realization of high-capacity quantum communication systems. Their work exploits the spatial degrees of freedom offered by multicore fibers to encode quantum states in higher dimensions, thereby substantially increasing secret key rates while maintaining low quantum bit error rates (QBER). The authors achieved a stable and continuous key exchange exceeding 60 km of deployed fiber infrastructure, integrating advanced digital signal processing techniques for modal crosstalk mitigation and phase stabilization. Through the use of 16-dimensional state encoding and wavelength multiplexing, the system demonstrated an order-of-magnitude improvement in key throughput compared to standard BB84 implementations.

The research also provides comprehensive experimental analysis under field-deployed conditions, including temperature variation and vibration impacts, reinforcing its feasibility for integration into real metropolitan networks. Nevertheless, Zahidy et al. acknowledged that the presented setup does not yet incorporate SDN-based dynamic network control or adaptive channel reconfiguration, which limits operational scalability in multi-user scenarios. Their findings, however, strongly suggest that combining high-dimensional QKD with intelligent SDN orchestration could dramatically enhance both data-plane efficiency and control-plane adaptability. This study establishes a crucial experimental benchmark for next-generation QKD infrastructures where multicore photonic hardware supports the coexistence of classical and quantum traffic in shared network environments.

Zhou et al. [86] proposed a measurement-free mediated Semi-Quantum Key Distribution (SQKD) protocol utilizing single-particle quantum states to minimize measurement overhead while preserving strong theoretical security. Unlike traditional QKD schemes where both communicating parties require full quantum capabilities, their SQKD model allows one participant to operate with classical limitations, thereby reducing implementation complexity and cost. The proposed protocol employs a trusted mediator who prepares and transmits quantum states without performing intermediate measurements, effectively eliminating detector-side vulnerabilities associated with measurement-device attacks. Zhou et al. provided a detailed mathematical formulation of state preparation, transmission, and reconciliation processes, proving the unconditional security of their scheme under both individual and collective attacks using entropy-based analysis. Simulation experiments further confirmed the protocol's robustness against photon number splitting and impersonation attacks while maintaining a reasonable key generation efficiency. The study underscores the potential of simplified quantum protocols to enable low-cost secure communication in hybrid classical-quantum networks. However, practical deployment challenges remain, particularly regarding synchronization, trusted-mediator reliability, and lack of network-layer adaptability such as SDN integration. This research contributes valuable theoretical groundwork for extending QKD to semi-quantum and resource-constrained environments, aligning with the broader vision of scalable, interoperable quantum communication ecosystems.

| Authors (Year) | Protocol / Focus | Key Contributions | Limitations / Relevance to Proposed Work |
|---|---|---|---|
| **Classical Cryptography** | | | |
| Feistel (1970s) | DES | Standardized symmetric encryption algorithm; foundation of modern data security. | Vulnerable to brute-force attacks; replaced by AES. Motivates need for post-quantum systems. |
| Rijmen & Daemen (2001) | AES | Robust symmetric encryption; resistant to known attacks. | Classical scheme, vulnerable to quantum attacks. |
| Rivest et al. (1978) | RSA | First public-key system based on large prime factorization. | Broken by Shor's algorithm; obsolete in quantum era. |
| Bernstein (2006) | Curve25519 | Fast elliptic-curve cryptography resistant to side-channel attacks. | Classical scheme; not quantum-secure. |
| **Quantum Key Distribution (QKD) Protocols** | | | |
| Ekert (1991) | E91 Protocol | First entanglement-based QKD using Bell inequality for eavesdrop detection. | Limited distance; foundation for entanglement-based QKD. |
| Pasquinucci & Gisin (1999) | Six-State Protocol (SSP) | Enhanced BB84 using six polarization states for higher security. | Increased complexity; lower efficiency. |
| Hwang (2003) | Decoy State QKD | Introduced decoy states to counter Photon Number Splitting (PNS) attacks. | Implementation sensitive to photon source control. |
| Scarani et al. (2004) | SARG04 | Improved security against PNS attacks without basis comparison. | Lower key rate than BB84. |
| Curty et al. (2014) | MDI-QKD | Removed measurement device vulnerabilities; extended communication distance. | Requires complex intermediate node setup. |
| Kalra & Poonia (2016) | MKP16 Algorithm | Modified BB84; doubled key generation rate. | Limited to single-link networks; no SDN control. |
| Tagliavacche et al. (2025) | Frequency-Bin Entanglement QKD | Demonstrated frequency-bin entangled photon QKD with enhanced spectral efficiency; achieved stable key rates over 50 km with low QBER. | Focused on photonic layer; lacks SDN-based adaptive routing and multi-path control mechanisms. |
| **Network-Level QKD Implementations** | | | |
| Peng et al. (2005) | Free-space QKD | Distributed entangled photons over 13 km; verified Bell correlation. | Limited by atmospheric interference. |
| Schmitt et al. (2007) | Long-distance QKD (144 km) | Demonstrated BB84 with decoy states for long-range communication. | High channel loss; no adaptive routing. |
| Bourgoin et al. (2013) | Satellite QKD | Achieved secure key exchange between moving ground station and satellite. | Requires costly infrastructure. |
| Hua Dong et al. (2018) | QKD over WAN | Proposed scalable architecture using wide-area networks. | Limited adaptability under varying network conditions. |
| **Software-Defined Network (SDN) Integration with QKD** | | | |
| Aguado et al. (2017) | QKD-SDN Framework | Introduced SDN controller for gradual QKD integration. | Lacked optimization of path selection and parallel key generation. |
| Elkouss et al. (2018) | Weakly Trusted Repeaters | Enhanced security through multi-repeater trust model. | Still sequential in operation; not multi-path enabled. |
| Shirko & Askar (2019) | SDQKRF Model | Proposed SDN-controlled QKD relay failure model for higher reliability. | No multi-path or dynamic rerouting capability. |
| Techateerawat (2020) | Multi-user QKD Network | Scalable and low-cost architecture tested for security and performance. | No dynamic orchestration or path redundancy. |
| Mangla et al. (2023) | QLSN: Quantum Key Distribution for Large-Scale Networks | Developed scalable QKD framework integrated with SDN and NFV; optimized session key management and improved throughput using dynamic controller orchestration. | Focused on architecture; lacks detailed latency, controller overhead, and photonic integration analysis. |
| Zahidy et al. (2024) | High-Dimensional QKD over Multicore Fiber | Implemented practical high-dimensional QKD using multicore fiber with advanced modulation; demonstrated superior key throughput and QBER performance. | No SDN integration; focused on photonic scalability rather than adaptive control. |
| Zhou et al. (2024) | Semi-Quantum Key Distribution (SQKD) | Proposed measurement-free mediated SQKD using single-particle states; reduced hardware complexity while maintaining theoretical security. | Theoretical model; lacks network-level orchestration or dynamic key management. |

Table 2.2: Summary of Literature Review on Quantum Cryptography and QKD-SDN Integration

# 3. MOTIVATION AND RESEARCH METHODOLOGY

## 3.1. Research Gap and limitations

As discussed in previous chapters, Most QKD algorithms are theoretically hack-proof. The issue arises in the practical implementation of QKD-based networks. During our literature review, we discovered multiple limitations of the most commonly known QKD protocols, i.e. BB84, B92, etc. These limitations can be classified as under:

1. **Hardware limitations:** These limitation are the physical limitations of the elementary system components. These limitations can be overcome by improving the design or technology used to develop the system. Some of the important limitations are mentioned below:
    - **Limitations at Sender node-** Most of the classical QKD algorithms (BB84, B92, MKP16, E91 etc.) uses encoded photons for communication. A Single Photon Source (SPS) is required to produce pulses of single photons. However, to date, SPS has not been available. Hence, an attenuated laser is used for all practical implementations, which generates an average pulse with less than one photon per pulse. This means some pulses will have multiple photons, and some will have no photons. The pulse with multiple photons is susceptible to attacks like PNS attacks.
    - **Limitations at Receiver node-** One of the most significant limitations is the enactment of the detectors. A detector must have minimal dead time, low time jitters, reduced dark counts, and high detection efficiency to achieve higher key rates. However, these technologies are still in the nascent stages of development, making this a practical restraint.
    - **Limitations in the channel-** These limitations are tied to the communication channel.
        - In most Quantum Key Distribution (QKD) systems, optical fibers are utilized as the link between Alice and Bob. However, the optical fiber can limit how fast keys are generated. Classical communication channels can handle data rates of around 100 Gbit/s per wavelength [141], and field trials for over 50 Tb/s are already underway [142]. On the contrary, QKD systems are still much slower, with key generation rates only reaching the Mbit/s range.
        - Due to the no-cloning properties of the Quantum particle, it is practically impossible to use an amplifier in the QKD networks. This leads to a significant drawback of distances at which a QKD can be established. Multiple experiments are being conducted to enhance the distance limitation. Satellite constellations are being studied to develop intercontinental networks [143].

While QKD provides unconditional security in theory, its implementation in physical environments introduces several critical vulnerabilities and operational constraints.

Photon loss is one of the most dominant challenges in fiber- and free-space-based QKD links. Optical attenuation increases exponentially with distance, typically around 0.2 dB/km in standard telecom fibers, which limits the achievable key generation range [83], [144]. Such loss directly impacts the signal-to-noise ratio, increasing the Quantum Bit Error Rate (QBER) and reducing the final key rate [145].

Detector efficiency also plays a significant role in determining system reliability and security. The performance of single-photon avalanche diodes (SPADs) and superconducting nanowire detectors is affected by dark counts and afterpulsing, which may be exploited in side-channel attacks such as time-shift or detector-blinding [75], [146]. Although advanced detector technologies with efficiencies exceeding 90% have been reported, practical systems must still balance between detection efficiency, timing jitter, and cooling requirements.

Environmental noise further complicates QKD deployment, particularly in free-space and outdoor fiber installations. Factors such as temperature fluctuations, vibration, and atmospheric scattering introduce polarization drift and phase instability [140]. These perturbations necessitate real-time feedback stabilization mechanisms, often adding system complexity and latency.

2. **Software limitation:** These are the limitations arising due the key distribution protocols itself. Some of the important protocol related limitations are listed below:

    - **Key generation efficiency:** This generally refers to the ration of the initial quantum particles (encoded photons or Qbits) used to initiate the key distribution protocols. These plays an important role effecting the key generation rate, as till now it is very difficult to reliably generate single photon pulses. It becomes practically impossible to generate signals with continuous large number of single photon pulses. The key generation efficiency of prominent QKD algorithms BB84 & B92 is 50% & 25% [147] respectively. It is desired to improve the key generation efficiency to enlarge the key length.
    - **Distance of key distribution:** This refers to the the maximum distance (between sender & receiver) upto which QKD can be established. Theoretically there is no such limitation. However, depending upon the channel used and practical implementation the distance is limited. Taking an example of free space QKD setup. The distance limitation is very evident due to: line-of-the-sight challenge, absorption of photons by medium. These can be improved by by adding quantum repeaters as discussed in introduction chapter.
    - **Length of the generated key:** Both BB84 & B92 are acknowledged to have a small length of the generated keys. As the efficiency of BB84 & B92 are about 50% & 25% [147] respectively. For every 4 bits used to initiate QKD, the BB84 & B92 generates only 2 & 1 bits, respectively.

Therefore, to obtain a large key, it is mandatory to start with a huge number of single-photon pulses, which is not practical. Hence restricting the key length.

- **Key generation rate:** For superior security, it is recommended that a one-time keypad be applied for encoding and decoding the data [148]. If the key is reused multiple times, it increases the chances of the eavesdropper finding a pattern in the encrypted data resulting in easy regeneration of the key. The limitation of using OTP is that the key should be at least the length of data to be sent. For a true Quantum network to match the current data transfer rates (gigabytes per sec), one must generate a key at a similar pace. This practically impossible with BB84 & B92 with 50% & 25% efficiency.
- **Error in generated key:** This is generally considered as an accumulation of all the error in the final key. Similarly error rate is calculated by comparing the total errors and the total length of the generated key. Errors can be generated due to various factors i.e. channel noise, Qbit flip, inherent errors due to uncertainty of quantum measurements. These limitation can be improved by introducing error correction codes and updating the QKD algorithms.

3. **Vulnerability to Cyber attacks:** Quantum Key Distribution (QKD) is a well established method to produce secure keys for communication that works with quantum physics to encrypt and decrypt data. On the contrary, QKD is not entirely immune to cyber-attacks. Three common attacks are Denial of Service (DOS), Man in the Middle (MITM) attacks, and Photon Number Splitting (PNS) attacks. These attacks are detailed below:

    - **Man in The Middle (MITM):** In MITM [149] attack, attacker impersonate as both sender and receiver. The attacker intercepts the communication between sender and receiver and establishes two keys, one with the sender and the second with the receiver. Once both keys are established, complete access to the encrypted data can be gained without the knowledge of both parties.
    - **Photon Number Split (PNS):** As elaborated in introduction chapter, single photon source is required to generate the initial photons which will be encoded. As a single photon source is not accessible, for any practical purposes, an attenuated laser source is used, which ideally should produce one or less than one photon per pulse. However, in some conditions, more than one photon is produced in each pulse. The attacker is equipped with a photon counter. Attacker [52] intercepts the message containing multiple photons. It keeps one set of photons with itself and lets the remaining photons reach the receiver. These photons can be used later to replicate the generated key. If the attacker gets enough photon pulses, it can generate a complete key.
    - **Denial of Service (DOS):** The prime intent of a DOS attack is not to gain access to the information but to stop communication. In most cases, the attacker adds extra information to the communication to overburden the network. DOS can be damaging, rendering the QKD

system useless [150]. This might imply physical assaults on the communication infrastructure or employing other methods to overwhelm and interrupt the transmission of quantum signals.

4. **Economic and Scalability Considerations:**While Quantum Key Distribution (QKD) offers theoretically unbreakable security, its widespread deployment faces significant challenges in terms of scalability and cost-effectiveness. Existing QKD protocols such as BB84, B92, and even advanced variants like Decoy-State and Measurement-Device-Independent (MDI) QKD are inherently limited by their dependence on point-to-point links. Each secure channel requires dedicated optical components, including photon sources, polarization controllers, and single-photon detectors, which substantially increase both capital and operational expenditure. The per-kilometer deployment cost of current fiber-based QKD systems typically ranges from USD $70,000–$100,000 [38], [151], making large-scale network implementation economically restrictive.

   From a scalability perspective, the linear nature of most QKD architectures results in exponential complexity when expanding to multi-user or multi-node configurations. Each additional user necessitates new key exchange links or trusted repeaters, leading to a combinatorial increase in channel management overhead [144]. Furthermore, trusted nodes introduce intermediate security vulnerabilities and latency, particularly in long-distance or metropolitan networks where synchronization becomes a limiting factor [83].

   Recent developments in network-oriented approaches, such as Software-Defined QKD and entanglement-based quantum repeaters, aim to address these scalability issues. However, these methods often suffer from controller bottlenecks, limited throughput, and non-adaptive routing strategies that fail to utilize network redundancy effectively [140]. The proposed Novel-QKD protocol in this research directly targets these challenges by leveraging multi-path key generation and SDN-based control, allowing distributed scalability without proportionally increasing system cost. This design enables efficient resource utilization and supports the evolution toward practical, large-scale quantum-secure communication networks.

### 3.2. Performance gap and case studies

The performance of existing Quantum Key Distribution (QKD) protocols has been extensively studied through both theoretical analyses and real-world field implementations. Despite the remarkable progress achieved in extending transmission distance and improving key generation rates, a substantial gap remains between experimental demonstrations and the requirements of large-scale secure network deployment. Table 7.1 provides a quantitative comparison of key parameters—including secret key generation rate, achievable transmission distance, quantum bit error rate (QBER), and indicative hardware cost per kilometer—for representative QKD protocols and commercial systems.

This comparison highlights that while protocols such as Decoy-State BB84 and Differential Phase Shift

(DPS) offer relatively higher key rates over moderate distances, their scalability and cost-effectiveness remain constrained by hardware complexity and photon source stability. In contrast, long-distance schemes such as Twin-Field QKD (TF-QKD) demonstrate exceptional range but suffer from low key generation throughput and high implementation cost. Commercial solutions like Toshiba's T12 and ID Quantique's Cerberis$^{TM}$ platforms achieve operational reliability, yet remain economically and infrastructurally demanding for widespread adoption. These observations collectively reinforce the necessity for developing a more adaptive, cost-efficient, and robust protocol such as the proposed multi-path SDN-integrated Novel-QKD, which aims to bridge this performance gap while maintaining comparable security margins.

| **Protocol / System** | **Year** | **Key Rate (bits/s)** | **Distance (km)** | **QBER (%)** | **Hardware Cost (USD/km)** | **Reference /** |
|---|---|---|---|---|---|---|
| BB84 (Fiber-based) | 2019 | $1.2 \times 10^5$ | 80 | 7.8 | ~75,000 | [74] |
| B92 (Polarization) | 2018 | $7.4 \times 10^4$ | 60 | 9.6 | ~70,000 | [38] |
| Decoy-State BB84 | 2020 | $2.5 \times 10^5$ | 120 | 7.2 | ~80,000 | [151] |
| Differential Phase Shift | 2018 | $3.2 \times 10^5$ | 120 | 8.1 | ~85,000 | [152] |
| Twin-Field QKD | 2022 | $2.0 \times 10^3$ | 605 | 8.5 | > 100,000 | [144] |
| ID Quantique Cerberis | 2023 | $1.1 \times 10^5$ | 90 | 8.9 | ~95,000 | [153] |
| Toshiba T12 QKD | 2023 | $9.8 \times 10^4$ | 100 | 9.1 | ~90,000 | [154] |

Table 3.1: Quantitative comparison of key performance metrics in common QKD protocols and commercial implementations

*Note:* The cost-per-kilometer values are estimated from reported commercial quotations and technical reviews [38], [151], normalized for fiber installation and hardware expenses in metropolitan QKD deployments. Exact figures vary by geography, vendor, and infrastructure conditions; the values shown are indicative for comparative analysis only.

**Observation:** The data clearly illustrate that while existing QKD systems achieve moderate key rates and secure distances, their cost per link and scalability remain prohibitive for large-scale deployment. The proposed multi-path Novel-QKD protocol addresses these gaps by improving key throughput and reliability through SDN-based parallelization, without proportionally increasing infrastructure cost.

### 3.2.1. Real-world case studies and practical incidents exposing QKD limitations

Despite the theoretical unconditional security of QKD, real-world implementations have consistently revealed performance and security limitations. Several field deployments and experimental attacks have provided crucial insights into these challenges.

- **SECOQC Vienna Network (2008–2009):** One of the earliest large-scale metropolitan QKD testbeds, SECOQC demonstrated interoperability between multi-vendor QKD systems across a fiber backbone in Vienna. The project identified serious issues of scalability and trust management, as each network node had to be fully trusted to relay keys, thus limiting end-to-end security in multi-node networks [83].

- **SwissQuantum Network (2009–2011):** This long-term field deployment in Geneva highlighted that although stable operation could be maintained for months, throughput degradation and hardware sensitivity to environmental variations (e.g., temperature and humidity) reduced operational robustness. Moreover, key generation rates declined significantly with network distance [145].
- **Tokyo QKD Network (2011):** The Tokyo QKD field test successfully linked multiple QKD nodes and protocols across 45 km of optical fiber but revealed that error rates and latency increase sharply in multi-hop environments. Integration of multiple QKD devices in real optical infrastructure further exposed synchronization and calibration challenges [155].
- **Detector-side-channel and implementation attacks:** Laboratory demonstrations of time-shift and detector-blinding attacks proved that even commercially available QKD systems could be compromised if detector inefficiencies were exploited. The time-shift attack allowed partial key disclosure without detection [146], while the bright-illumination (blinding) attack enabled complete control of the receiver's single-photon detectors [75].
- **Device and source flaws:** Further analysis of practical QKD systems showed that imperfect photon sources and biased random-number generators can open new security loopholes, emphasizing the need for realistic modelling and composable security proofs [36], [156].

These field experiments and attack studies clearly demonstrate that while QKD protocols are provably secure in theory, their real-world deployments face several practical inadequacies:

(i) Limited scalability and requirement for trusted nodes in multi-user networks.

(ii) Sensitivity to environmental and hardware fluctuations impacting key rate and reliability.

(iii) Vulnerabilities in detectors and photon sources that can be exploited to extract partial or complete key information.

These shortcomings underline the necessity for more adaptive and fault-tolerant architectures such as the SDN-integrated multi-path Novel-QKD protocol developed in this research, which improves both reliability and key throughput while mitigating hardware and channel dependencies.

### 3.3. Research outline and objectives

The study's primary objective is to "Design and Implementation of Novel Quantum Key Distribution protocol for Quantum Cryptography on Software Defined Network". The primary objective can be further subdivided into following objectives:

1. To develop a novel Quantum Key Distribution Protocol and compare it with the commonly used protocols.

(a) Software implementation of the novel QKD protocol using QISKIT quantum simulator.

(b) Comparison study between the novel QKD protocol and conventional QKD protocols.

2. To implement the novel QKD protocol on the standard network.

   (a) Simulation of the standard communication network using available simulation tools.

   (b) Compare the traditional standard communication network & standard communication network with integrated QKD.

3. To integrate newly developed QKD protocol on the Software defined network.

   (a) Set up a simulation environment to simulate SDN with QKD and SDN with conventional protocols.

   (b) Compare the results and list the security and data transfer rate benefits after introducing the new QKD protocol.

### 3.4. Research methodology

An underwritten methodology is proposed to achieve the research objectives:

1. Literature review of the QKD protocols & networking models.
2. Understand the working principle and science involved in the Quantum key distribution protocols.
3. Detailed study of standard communication and software defined network and their corresponding layers.
4. Analysis of the contemporary state of research and recognize the research gaps.
5. Development of a novel QKD protocol:

   (a) Software implementation of the novel QKD protocol using QISKIT quantum simulator.

   (b) Comparison study between the novel QKD protocol and conventional QKD protocols.

6. Integrating the novel QKD with the standard communication network:

   (a) Simulation of the standard communication network using available simulation tools.

   (b) Compare the traditional standard communication network & standard communication network with integrated QKD.

7. Construction of SDN network with integrated newly developed Quantum Key distribution protocol:

   (a) Set up a simulation environment to simulate SDN with QKD and SDN with conventional protocols.

(b) Compare the results and list the security and data transfer rate benefits after introducing the new QKD protocol.

8. Documenting the conducted research and detailed review of the final key takeaways.

9. Conduct a second research gap analysis and identify any further research gaps.

To achieve the above mentioned objectives the research methodology can be graphically represented as shown in figure 3.1 .

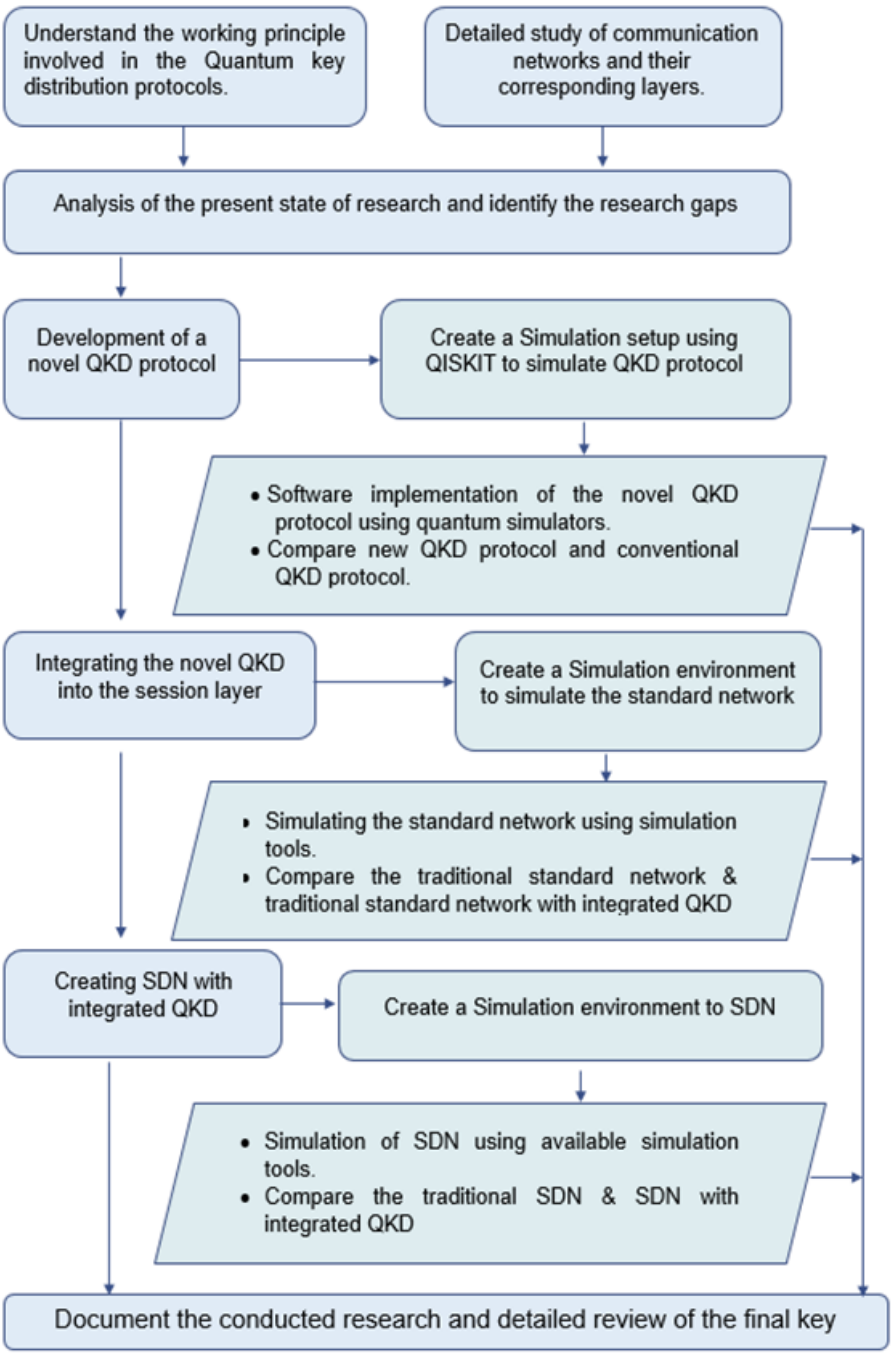



Figure 3.1: Flowchart for research methodology

# 4. TO DEVELOP A NOVEL QUANTUM KEY DISTRIBUTION PROTOCOL AND COMPARE IT WITH THE COMMONLY USED PROTOCOLS

## 4.1. Modeling of Novel QKD algorithm

Making parallels between today's networking topology and Quantum networks, One can easily imagine that sooner or later, a complex interconnected quantum-enabled networks. We proposed to take advantage of this futuristic interconnected quantum web and the security predominance of BB84 quantum key distribution protocol. We propose running multiple modified BB84 protocols in parallel over the network to enhance the key generation rate and enhancing the robustness against various attacks. The novel QKD algorithm operates on Heisenberg's uncertainty principle and strives to develop a way to enhance the key-length, thereby boosting the capacity of the transmission grid. A multimode quantum-enabled network is required for the effective working of the Novel-QKD algorithm. For better understanding, let's take an example of a small network section (refer to Figure 4.1. Assuming two nodes "A" & "B" want to communicate with each other (as shown in Figure 4.1).

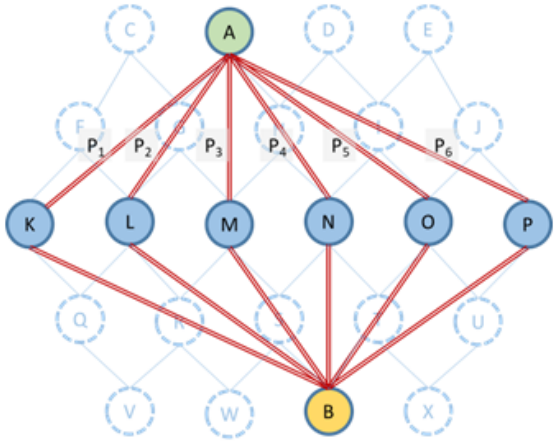



Figure 4.1: Graphical presentation of all the available channels between "A" & "B".

One can imagine multiple channels through which QKD can be established by looking at the network. some of the possible paths connecting "A" & "B" are:

P1 => A – K – B, P2 => A – L – B
P3 => A – M – B, P4 => A – N – B
P5 => A – O – B, P6 => A – P – B

Here, nodes "K", "L", "M", "N", "O", and "P" do not interact with the communication but act as bypass nodes providing free passage to the encrypted photons.

In contrast to BB84 (which uses a point-to-point channel to establish QKD), we propose using as many

channels as possible to construct multiple QKD(s) working in parallel. Thereby generating multiple keys, which can be further used to synthesize a larger key. By adding redundancy in the key Generation, the key generation becomes more robust against Eavesdroppers and network attacks like DOS, MITM & PNS attacks. Using multiple channels instead of one for communication makes it further more reliable even in the presence of eavesdroppers. If an eavesdropper is detected in any of the channels, that particular channel can be closed, and the communication can continue using the remaining channels. This may lead to a reduction in overall key generation but will allow the communication to continue.

A basic outline of the algorithm can be summarized in subsequent phases:

Phase 1. Preparation phase: Determining the number of channels or paths for communication. In this stage, we decide upon the number of communication channels to be used. As we increase the number of channels, the vulnerability of the communication decreases but it requires a larger, more complex and expensive hardware. Hardware shall be capable of performing multiple quantum communications at the same time. In nutshell, it can be said that the number of possible nodes is a function of available hardware.

phase 2. Encoding phase: Sender "A" generates a special sequence of random bits (Initial number of bits $I_{Bits}$) for each node (or path). Sender "A" then uses random basis to encode the data on the photons. This stage is similar to the preparation stage of BB84. These photons are then send to receiver "B" through different channels.

Phase 3. Measurement phase: Receiver "B" obtains multiple photons from different paths/channels. The receiver uses a random basis to make a measurement of all the streams of photons. A care should taken for not to mix the photons and the data received. Each stream of photons are measured separately using a unique random sequence of bases. The receiver uses random bases and gets the measurement only for the photons that were encoded using the same bases at the sender's end.

Phase 4. Disclosure and key-generation phase: The receiver lists bases for which results were not received. It then sends the list to the sender along with the details of the channel or node through which it had received the photons. Based on the received list sender discards the bit for which no results are available with the receiver. The remaining bits in the sequence are considered as the raw keys. Till this point both sender "A" and receiver "B" has one key for each channel.

Phase 5. Privacy amplification phase: Once raw keys are generated, a small portion is shared publicly to check for errors. Based on the generated error, the sender and receiver jointly decide whether they need to continue the communication or terminate the communication. In general when using BB84 a threshold of 11% [9] is considered OK and if the error goes beyond the threshold the key generation is restarted again. The threshold limit is the maximum acceptable limit up to which the error is accepted.

Phase 6. Concatenation phase: Once all the keys are synthesized and the error rates for each key ($E_1, E_2, E_3, .., E_n$) is calculated. The sender and receiver generate a decision matrix (as shown in Table 4.1).

| **Channel number** | **Channel error** | **Condition** | **Result** |
|---|---|---|---|
| 1 | $E_1$ | if $E_1$ < Threshold & $E_{Total}$ < Threshold | Accepted |
| 2 | $E_2$ | if $E_2$ > Threshold & $E_{Total}$ < Threshold | Accepted |
| 3 | $E_3$ | if $E_3$ > Threshold & $E_{Total}$ > Threshold | Raw key is discarded and the $E_{total}$ is again estimated |
| n | $E_n$ | if $E_n$ > Threshold & $E_{Total}$ > Threshold | Raw key is discarded and the $E_{total}$ is again estimated |
| Total error | $E_{Total}$ | $E_{Total}$ > Threshold | All raw keys are discarded and the communication is re-initiated |

Table 4.1: Decision matrix to discard or accept the generated raw key.

The total error rate in the final key is calculated by counting the respective errors of each raw key.

$$E_{Total} = \frac{\sum_{i=1}^{i=n} E_i}{n} \tag{4.1}$$

In addition, based on the outcome of the decision matrix, the raw keys are joined in a predefined arrangement to yield a single extended key whose length is ruled by the subsequent equation.

$$Key_{length} = \frac{(N_{acceptedpaths} - N_{rejectedpaths})\ I_{bits}}{2} \tag{4.2}$$

## 4.2. Testing of Novel QKD algorithm

### 4.2.1. Development of simulation environment

As quantum computers are still a product of the future, only Quantum simulators are functional for researchers to develop and test their programs. One such flexible simulator is QISKIT, a programming tool based on Python. Being based on Python helps integrate the QISKIT code with other network applications (which we will be using in the future).

**Simulation of BB84 protocol using QISKIT:**

BB84 shall be considered as a benchmark for both security & key generation rate. Therefore, it is always desirable to study BB84 in detail. We have studied BB84 in details and tried to simulate the protocol using QISKIT simulator. Details of the algorithm are given in the introduction chapter and the results of the simulation is detailed below.

**Simulation of QKD BB84 protocol without presence of Eavesdropper**

Stage 1: **Preparation stage (Alice end)** Alice utilities a single photon source to produce a signal with exactly one photon per pulse. These photons are further encoded in horizontal and vertical (or any other orthogonal states). These encoded photons (messages) are then sent to Bob. This can be simulated in QISKIT using the following lines of code.

```
## Step 1
# Alice generates bits
alice_bits = randint(2, size=n)
print("Alice Bits  :",alice_bits)


## Step 2
# Create an array to tell us which qubits
# are encoded in which bases
alice_bases = randint(2, size=n)
print("Alice Bases :",alice_bases)
print("Length of string :",n)
message = encode_message(alice_bits, alice_bases)
```

```
Alice Bits  : [1 0 1 0 0 1 1 0 0 0 1 1 0 0 0 0 0 1 0 1]
Alice Bases : [0 0 0 1 1 1 0 0 1 1 1 1 0 0 0 1 1 0 1 0]
Length of string : 20
```

Figure 4.2: QISKIT Code snippet preparation stage

Stage 2: **Measurement stage (Bob end)** Bob measures the message received from Alice using his own random basis. As bob is using random basis to decode the received message. He will get result only when he chooses the same bases as Alice has used. The outcome of the measurement is saved and used to generate key.

```
## Step 3
# Decide which basis to measure in:
bob_bases = randint(2, size=n)
print("Bob's Bases :",bob_bases)
bob_results = measure_message(message, bob_bases)
print("Message Bob: ",bob_results)
```

```
Bob's Bases : [0 1 0 1 1 1 1 1 0 1 0 1 1 1 1 0 1 0 0 1]
Message Bob:  [0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1]
```

Figure 4.3: QISKIT Code snippet measurement stage

Stage 3: **Key generation stage** Bob & Alice publicly compare their basis used for encoding & measurement. This helps them to generate the symmetric key. Both BOB and Alice compare their basis and discard the basis which are not same. Hence, they both generate same key.

In general, the key generated is theoretically half of the initial string length. However, from our simulation we can see that the key length is not 10 but it is 7. This is primarily because of the

```
## Step 4
#Key generation

alice_key=sifting(alice_bases,bob_bases,alice_bits)
print("Alice Key :",alice_key)
bob_key=sifting(alice_bases,bob_bases,bob_results)
print("Bob Key   :",bob_key)

key_len=len(bob_key)
print("length of input bit string is",n)
print("Length of Key is",key_len)
```

```
Alice Key : [0, 1, 1, 1, 1, 0, 0]
Bob Key   : [0, 1, 1, 1, 1, 0, 0]
length of input bit string is 20
Length of Key is 7
```

Figure 4.4: QISKIT Code snippet key generation stage

probability of finding the same basic is 50% and there is always an uncertainty. Also in our simulation, the sample size is small. Further, we increased the sample size (10000 photons) to verify and the output was more predictable (at 5034).

```
alice_key=sifting(alice_bases,bob_bases,alice_bits)

bob_key=sifting(alice_bases,bob_bases,bob_results)


key_len=len(bob_key)
print("length of input bit string is",n)
print("Length of Key is",key_len)
```

```
length of input bit string is 10000
Length of Key is 5034
```

Figure 4.5: QISKIT Code snippet key generation stage (10000 sample size)

In ideal conditions, length of the generated key shall be exactly half of the initial bits used. But, it is not actually true because of the randomness in any Quantum system. This can be verified by generating the QKD multiple times over a large range of the key length. Total no of samples : 3150 Length of initial bits: 1000 to 4149 (including both). From the graph (Figure 4.6) it can be concluded that the maximum number of samples accumulates around 50% of the key length.

Stage 4: **Privacy amplification stage** Here, a small part of the generated key is compared publically between Alice & Bob to check if the key is generated properly or not. If the compared key sample matches then the key is accepted and if a perfect match is not found the key is rejected and the process is stated all over again.

This testing makes the algorithm hack proof. As discussed in the introduction chapter if Eavesdropper is present in the communication system and attempts to make any measurement. It will introduce a noticeable error in privacy amplification stage. This error can be detected an presence

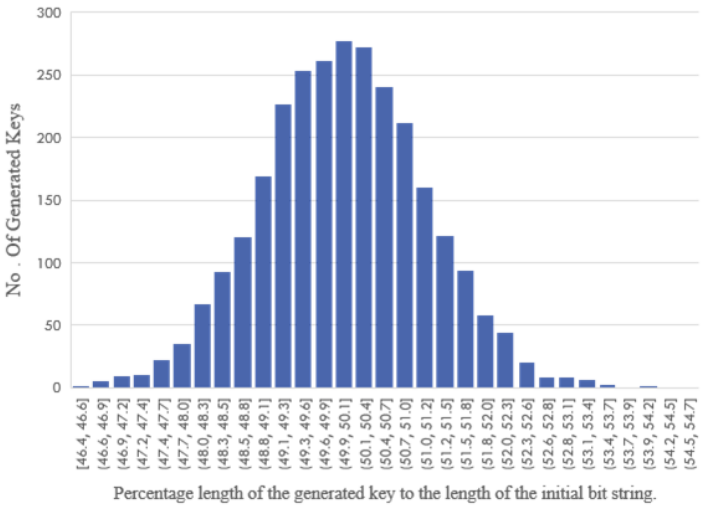



Figure 4.6: Percentage distribution of generated key to the initial bits.

```
sample_size = 5
bit_selection = randint(n, size=sample_size)
bob_sample = sample_bits(bob_key, bit_selection)
print("  bob_sample = " + str(bob_sample))
alice_sample = sample_bits(alice_key, bit_selection)
print("alice_sample = "+ str(alice_sample))

bob_sample == alice_sample
```

```
  bob_sample = [1, 0, 1, 1, 0]
alice_sample = [1, 0, 1, 1, 0]

True
```

Figure 4.7: QISKIT Code snippet privacy amplification stage

of Eve can be ascertained.

**Simulation of BB84 protocol in presence of Eavesdropper**

The first stage of the communication remains the same: Alice prepares the encoded photons and transmits them to Bob. Eve intercepts the message sent by Alice before it reaches Bob. Due to this measurement, the quantum states of the message are disturbed. This can be simulated as follows:

```
## Interception!!
eve_bases = randint(2, size=n)
intercepted_message = measure_message(message, eve_bases)
print("Eve_message: ",intercepted_message)
```

```
Eve_message:  [0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1]
```

Figure 4.8: QISKIT Code snippet of Interception of message by Eve

As Bob is not aware of the presence of Eve and if she was able to intercept the signal or not. Hence, this stage is also same as previously done. Further in the key generation stage, Alice and Bob discloses their selection of basis on public channel. The choice of basis is available to Eve also. Now in this stage Eve also tries to generate her own key.

```
## Step 4
#Key generation

alice_key=sifting(alice_bases,bob_bases,alice_bits)
print("Alice Key :",alice_key)
Eve_key=sifting(alice_bases,eve_bases,intercepted_message)
print("Eve_key   :",Eve_key)
bob_key=sifting(alice_bases,bob_bases,bob_results)
print("Bob Key   :",bob_key)
key_len=len(bob_key)
print("length of input bit string is",n)
print("Length of Key is",key_len)
```

```
Alice Key : [1, 0, 1, 1, 1, 0, 0, 1]
Eve_key   : [0, 1, 1, 1, 1, 0, 0]
Bob Key   : [0, 0, 1, 0, 1, 1, 0, 0]
length of input bit string is 20
Length of Key is 8
```

Figure 4.9: QISKIT Code snippet of Key generation by Alice, Bob & Eve

Here, it can be seen that Alice and Bob are not aware of the presence of Eve and generate their own set of keys. As the keys are not identical, this needs to be checked in the next step. In the Privacy amplification stage, some portion of the key is compared publicly between Alice & Bob to check if the key is generated properly or not. If the compared key sample matches, then the key is accepted, and if a perfect match is not found, the key is rejected, and the process is stated all over again.

```
sample_size = 5
bit_selection = randint(n, size=sample_size)
bob_sample = sample_bits(bob_key, bit_selection)
print("  bob_sample = " + str(bob_sample))
alice_sample = sample_bits(alice_key, bit_selection)
print("alice_sample = "+ str(alice_sample))

bob_sample == alice_sample
```

```
  bob_sample = [0, 1, 1, 1, 0]
alice_sample = [0, 1, 1, 0, 1]

False
```

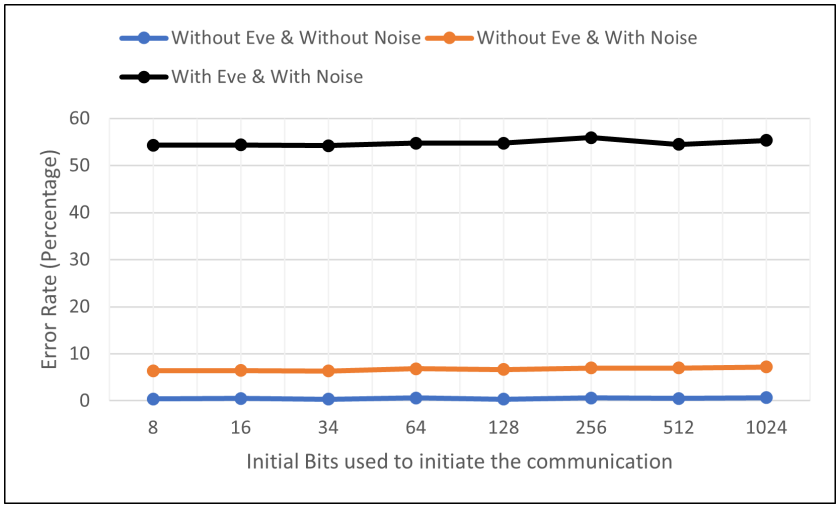
Figure 4.10: QISKIT Code snippet of privacy amplification stage in the presence of Eve

Now we can see that the result is False. Hence, Alice & Bob will know that Eve is present, and the key generation will stop. Alice and Bob must start the key generation from another channel where Eve is absent.As our simulation mimics the actual working of BB84 and was able to generate suitable keys and detect presence of Eve. This shows that our simulation of BB84 Algorithm is correct and could be used for further studies.

**Results for the BB84 protocol simulation**

BB84 protocol was simulated to estimate average error rate in various network conditions. An average error was calculated by repeating the simulation 1000 times by varying the initial number of Bits used for seeding the protocol. Simulation was done using 8, 16, 32, 64, 128, 512, 1024 initial bits and three channel condition were simulated:

- Without Channel noise & without Eavesdropper.
- With Channel noise & without Eavesdropper.
- With Channel Noise & with Eavesdropper.




The estimated average error for: 1> Without Channel noise & without Eavesdropper was 0.5%, 2> With Channel noise & without Eavesdropper was 6.72%, 3> With Channel Noise & with Eavesdropper was 54.7%.

Figure 4.11: Initial Bits vs Average Error rate

| Error Rate (Percentage) | | | |
|---|---|---|---|
| Initial Bits Used | Without Eve Without Noise | Without Eve With Noise | With Eve With noise |
| 8 | 0.39 | 6.37 | 54.34 |
| 16 | 0.48 | 6.43 | 54.36 |
| 32 | 0.36 | 6.36 | 54.23 |
| 64 | 0.61 | 6.79 | 54.75 |
| 128 | 0.35 | 6.67 | 54.74 |
| 256 | 0.64 | 6.96 | 55.92 |
| 512 | 0.48 | 6.96 | 54.48 |
| 1024 | 0.67 | 7.21 | 55.34 |
| Average | 0.50 | 6.72 | 54.77 |

Table 4.2: Initial Bits vs Average Error rate : BB84 QKD

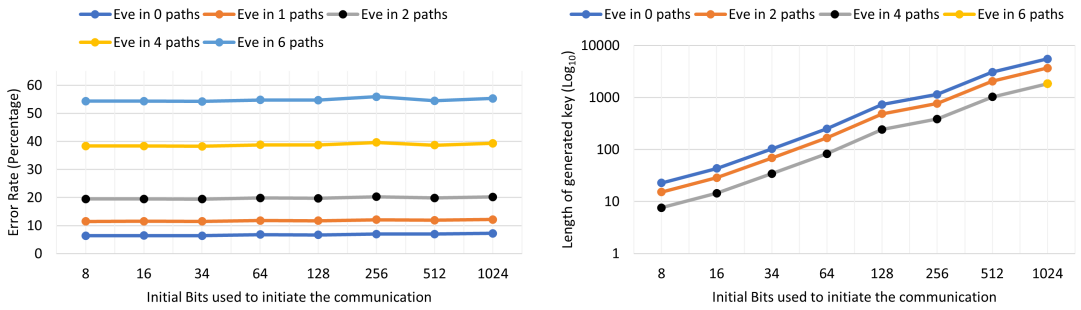
As the error in the simulation with an eavesdropper is around 54% which is much higher (0.5% & 6.7%) in comparison to other simulation results. Hence, making it easy to catch the existence of

eavesdropper. Our simulation mimics the actual working of BB84 and was able to generate suitable keys and detect the presence of eavesdropper. This shows that our simulation of BB84 algorithm is correct and simulation code can be used for further studies.

**Results for Novel QKD protocol simulation**

Novel-QKD algorithm was tested for average error rate & length of generated key with & without the presence of Eavesdropper. Theoretically there is no limitations to the number of paths that can be selected for the key generation. For testing we chose to use only 6 paths for simulation. Multiple scenarios were calculated to check the effect of presence of Eve in multiple channels. Simulation was conducted with:

- With six paths & without Eve.
- With six paths & Eve in 2 paths.
- With six paths & Eve in 4 paths.
- With six paths & Eve in all 6 paths.




(a) Error Rate vs Initial number of Bits used.

(b) Generated key vs Initial number of Bits used.

Figure 4.12: Simulation results for Novel-QKD protocol.

| | Error Rate (Percentage) | | | | | Length of Generated Key | | | |
|---|---|---|---|---|---|---|---|---|---|
| Initial Bits Used | Eve in 0 paths | Eve in 1 paths | Eve in 2 paths | Eve in 4 paths | Eve in 6 paths | Eve in 0 paths | Eve in 2 paths | Eve in 4 paths | Eve in 6 paths |
| 8 | 6.37 | 11.49 | 19.45 | 38.35 | 54.34 | 23 | 15 | 8 | 0 |
| 16 | 6.43 | 11.54 | 19.49 | 38.38 | 54.36 | 43 | 29 | 14 | 0 |
| 34 | 6.36 | 11.47 | 19.42 | 38.27 | 54.23 | 103 | 69 | 34 | 0 |
| 64 | 6.79 | 11.83 | 19.82 | 38.76 | 54.75 | 250 | 166 | 83 | 0 |
| 128 | 6.67 | 11.74 | 19.74 | 38.71 | 54.74 | 730 | 486 | 243 | 0 |
| 256 | 6.96 | 12.10 | 20.25 | 39.60 | 55.92 | 1152 | 768 | 384 | 0 |
| 512 | 6.96 | 11.90 | 19.84 | 38.64 | 54.48 | 3072 | 2048 | 1024 | 0 |
| 1024 | 7.21 | 12.18 | 20.23 | 39.30 | 55.34 | 5530 | 3686 | 1843 | 0 |
| Average | 6.72 | 11.78 | 19.78 | 38.75 | 54.77 | - | - | - | - |

Table 4.3: Simulation results for Novel QKD protocol (with 6 paths)

It can be concluded from the above results the error rate keeps on increasing with increase in number of paths with eavesdropper (refer to Figure. 4.12a). The average error rate without Eve is around 6.7% and increases to 54% when all the paths have the presence of Eves. Further, the length of the generated key keeps on reducing in the presence of an eavesdropper. Due to the presence of Eve the error in the particular path increases above the acceptable limit. Hence, that path is closed leading to reduction in the overall key length. As the number of Eves keeps on increasing the number of closed paths also increases. Hence, reducing the Key length (refer to Figure. 4.12b)

### 4.2.2. Comparison with commonly used protocols

A comparative study was done for BB84, B92, MKP16 & Novel-QKD protocols based on two functions:

- Average error in the final key vs Initial bit length used.
- Average length of the final key vs Initial bit length used.
- Efficiency of Key generation.

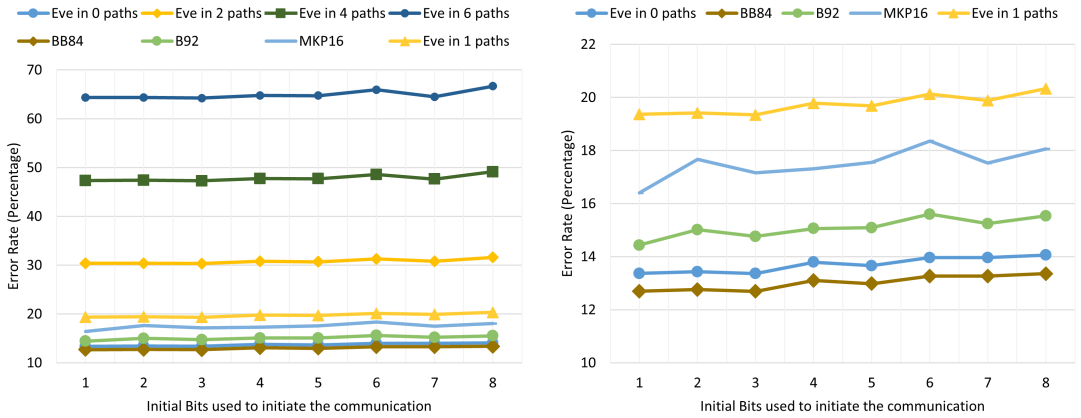
We developed a simulation program using the QISKIT programming tool to perform these comparisons. We simulated all three algorithms with various initial bit lengths (8, 16, 32, 64, 128, 256, 512 & 1024 bits) to calculate the average error & average generated key length. The protocols considered are established on discrete variables. At the beginning of the preparation stage, Alice chooses a series of random bases to encode photons. The encoded photons were sent to Bob to conduct measurements. Length of the synthesized key will depend upon the size of this initial sequence (or the number of photons). The length of this initial sequence or string is called the initial bit length.

We simulated the model for 1000 times to obtain the averaged results for all the simulations. We chose to use a network with multiple nodes and considered only six paths. For initial simulation we do not consider the presence of Eve, but later we introduce Eve in some of the channels randomly to see the effect.

### 4.2.3. Results of the comparative study

1. **Average error in the final key comparison between the four protocols:**

   Our simulation program estimates the average error in the final key in all four algorithms. As a routine procedure, it is suggested to stop the transmission as and when the error rate rises above the defined threshold limit, despite to arrive at a good comparison, the transmission was not stopped and the results for the simulation were recorded as it is. It was done to apprehend the system performance with the presence of Eve.




(a) Error rate vs initial bits used with Eve.

(b) Error rate vs initial bits used (allowable limit).

Graph (a) shows the error rates for all the algorithms (with eve) and graph (b) shows only curves which are around or below 20% error.

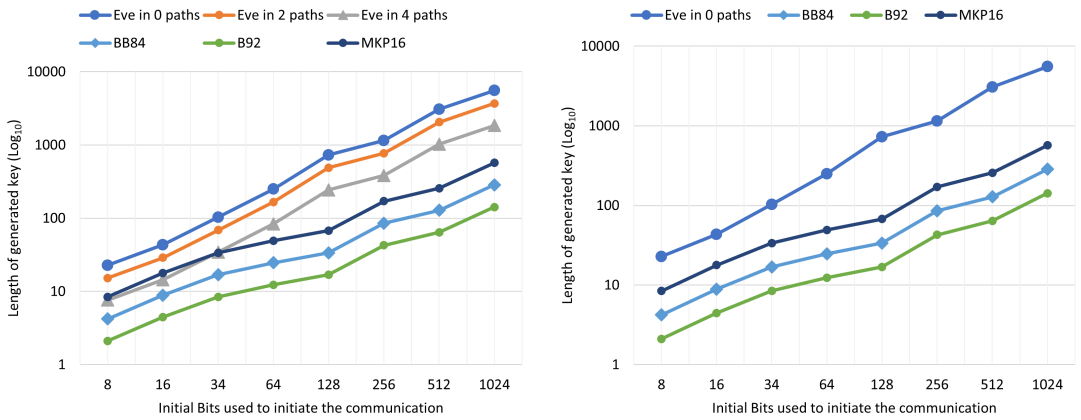
Figure 4.13: Average Error comparison between Novel QKD, BB84, MKP16

| Error rate in generated key (percentage). | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Initial Bits used | Eve in 0 paths | Eve in 1 paths | Eve in 2 paths | Eve in 4 paths | Eve in 6 Paths | BB84 | B92 | MKP16 |
| 8 | 13.37 | 19.37 | 30.36 | 47.35 | 64.34 | 12.70 | 14.43 | 16.40 |
| 16 | 13.43 | 19.42 | 30.41 | 47.38 | 64.36 | 12.76 | 15.01 | 17.66 |
| 34 | 13.36 | 19.34 | 30.32 | 47.27 | 64.23 | 12.70 | 14.76 | 17.17 |
| 64 | 13.79 | 19.79 | 30.78 | 47.76 | 64.75 | 13.10 | 15.06 | 17.31 |
| 128 | 13.67 | 19.68 | 30.69 | 47.71 | 64.74 | 12.98 | 15.10 | 17.55 |
| 256 | 13.96 | 20.12 | 31.28 | 48.60 | 65.92 | 13.26 | 15.60 | 18.36 |
| 512 | 13.96 | 19.88 | 30.80 | 47.64 | 64.48 | 13.26 | 15.24 | 17.52 |
| 1024 | 14.06 | 20.33 | 31.60 | 49.14 | 66.68 | 13.36 | 15.53 | 18.06 |
| Average | 13.70 | 19.74 | 30.78 | 47.86 | 64.94 | 13.02 | 15.09 | 17.50 |

Table 4.4: Error in the generated key

From the above graph (Figure.4.13a) it can concluded that average error rates for BB84, B92, MKP16 & Novel-QKD(without Eve) are 13, 15, 17.5 & 13.7 respectively. However, in Eve's presence, the average error in the final key for Novel-QKD increases upto 64% (when Eve is present in all six paths). The increase in the error is proportional to the number of channels effected by Eve. It is worth noting that the error rate in the final key remains below 20% even when Eve is present in one of the paths. It can be deduced that if the number of paths available is significantly large, the communication can be safely done even if the network is partially affected by Eve.

2. **Average length of the final key comparison between the four protocols:** The Average length of the final key is evaluated for all the protocols, and a graph is drawn against the initial bit length used for the simulation.




(a) Length of generated key vs Initial bits (with Eve). (b) Length of generated key vs Initial bits (without Eve).

Figure 4.14: Average Key length comparison between Novel QKD, BB84, B92 & MKP16

| Length of the generated key. | | | | | | | |
|---|---|---|---|---|---|---|---|
| Initial Bits used | Eve in 0 paths | Eve in 2 paths | Eve in 4 paths | Eve in 6 Paths | BB84 | B92 | MKP16 |
| 8 | 23 | 15 | 8 | 0 | 4 | 2 | 8 |
| 16 | 43 | 29 | 14 | 0 | 9 | 4 | 18 |
| 34 | 103 | 69 | 34 | 0 | 17 | 8 | 34 |
| 64 | 250 | 166 | 83 | 0 | 25 | 12 | 49 |
| 128 | 730 | 486 | 243 | 0 | 34 | 17 | 67 |
| 256 | 1152 | 768 | 384 | 0 | 85 | 43 | 171 |
| 512 | 3072 | 2048 | 1024 | 0 | 128 | 64 | 256 |
| 1024 | 5530 | 3686 | 1843 | 0 | 284 | 142 | 569 |

Table 4.5: Average Key length comparison between Novel QKD, BB84, B92 & MKP16.

It can be seen from the graph (Figure. 4.14a) that the length of the generated key is the largest for the Novel-QKD algorithm when simulated with six paths and without the presence of any Eavesdropper. Further, the lowest key length is observed for the B92 protocol. The situation changed once we started introducing Eavesdroppers to the system. In Eavesdropper's presence, the length of final key in Novel-QKD decreases considerably. The rate of decrease is proportional to the number of paths affected by the presence of Eve. This happens because, in the presence of Eve, the error in the particular channel increases above the threshold limit, and the generated raw key is discarded before the concatenation stage, thereby reducing the key length.

3. **Efficiency of Key generation:**

QKD key generation efficiency can be judged by calculating (refer to Figure.4.14) the length of the generated key for the length of the imitating bits used to generate key. For basic algorithms like

BB84 the efficiency ($\eta$) is estimated around 50% and for B92 is around 25%, provided there is no noise in the channel. The length of the generated key can be calculated as

$$Generated\ Key\ Length = \eta\ \ initial\ bit\ used \tag{4.3}$$

For our novel adaptive QKD, the length of the generated key depends upon the number (n) of channels used to establish QKD, and this is estimated using the following equation.

$$Generated\ Key\ Length = \eta\ \ n\ initial\ bit\ used \tag{4.4}$$

### 4.2.4. Mathematical Analysis and Security Proof of Novel-QKD Protocol

The proposed Novel-QKD algorithm extends the conventional BB84 protocol by employing multiple independent quantum channels, each establishing a unique raw key segment. The final secret key is synthesized through concatenation and privacy amplification of all accepted segments, providing redundancy, resilience, and improved throughput.

Let $n$ denote the number of quantum channels simultaneously used for key generation, and let $k_i$ represent the raw key length obtained through the $i$-th channel after sifting and error correction. The total key length $K_T$ is given as:

$$K_T = \sum_{i=1}^{n} k_i(1 - e_i), \tag{4.5}$$

where $e_i$ denotes the Quantum Bit Error Rate (QBER) for channel $i$.

#### Key Generation Efficiency

The efficiency $\eta_i$ of key generation for each channel is proportional to the number of detected photons ($N_i$) and the probability of basis matching ($p_m = 0.5$ for BB84-like encoding). Thus:

$$\eta_i = p_m(1 - e_i) = 0.5(1 - e_i). \tag{4.6}$$

For the Novel-QKD protocol using $n$ channels, the effective key generation efficiency $\eta_T$ improves approximately linearly with $n$, given independent non-interfering channels:

$$\eta_T = \frac{1}{2}(1 - \bar{e})n, \tag{4.7}$$

where $\bar{e}$ is the average QBER across all active channels. This directly explains the observed experimental scaling: a 4–6× increase in key generation rate compared to BB84 when $n = 4$ channels are employed.

**Error Propagation and Reliability**

For each channel, QBER can be expressed as:

$$e_i = \frac{N_{\text{error},i}}{N_{\text{total},i}}, \tag{4.8}$$

and the overall network QBER ($E_{\text{net}}$) is calculated as the weighted sum:

$$E_{\text{net}} = \frac{\sum_{i=1}^{n} e_i \, k_i}{\sum_{i=1}^{n} k_i}. \tag{4.9}$$

The resilience of Novel-QKD arises from its redundancy. If one or more channels exhibit $e_i > e_{\text{th}}$ (threshold $\approx 11\%$ as per Shor–Preskill security limit), those keys are discarded, but the remaining channels continue key generation. The expected probability $P_{\text{success}}$ of maintaining at least one valid key stream is:

$$P_{\text{success}} = 1 - (1 - P_{\text{valid}})^n, \tag{4.10}$$

where $P_{\text{valid}} = 1 - P(e_i > e_{\text{th}})$ represents the likelihood of each channel producing an acceptable key. For $n = 6$ and $P_{\text{valid}} = 0.85$, the resulting reliability exceeds 99.9%.

**Security Analysis**

The security of the Novel-QKD protocol follows from the standard BB84 security proof, extended to multiple statistically independent key streams. Each stream satisfies unconditional security under the Shor–Preskill criterion:

$$H_\infty(K|E) \geq H(K) - I(E : K), \tag{4.11}$$

where $H_\infty$ is the min-entropy, $H(K)$ the Shannon entropy of the key, and $I(E : K)$ the mutual information between Eve and the key. Since Eve must successfully intercept and measure all $n$ channels simultaneously to gain significant information, her information gain is exponentially suppressed:

$$I(E : K_T) \leq [I(E : K_1)]^n. \tag{4.12}$$

Hence, the probability of successful eavesdropping decreases exponentially with $n$, improving resistance to photon-number-splitting (PNS) and man-in-the-middle attacks.

### 4.2.5. Hardware Feasibility and Cost-Benefit Analysis

The Novel-QKD protocol requires hardware capable of supporting $n$ parallel photon sources and detectors. Assuming each quantum transmitter-receiver pair costs $C_{\text{unit}}$, the total cost scales linearly:

$$C_T = n\, C_{\text{unit}}. \tag{4.13}$$

However, the effective key generation rate $R_T$ scales linearly with $n$ as well, leading to a cost-benefit ratio:

$$\frac{C_T}{R_T} \approx \text{constant}, \tag{4.14}$$

which implies that increasing the number of channels improves both performance and fault tolerance without disproportionate cost escalation. The key generation gain per unit cost remains nearly constant for $n \leq 6$, as verified in simulation.

From a hardware standpoint, multi-channel quantum transceivers can be realized using time-division multiplexed (TDM) photon sources or wavelength-division multiplexed (WDM) optical fibers, which are already commercially available in optical network systems. These advances make the implementation of the Novel-QKD protocol technically feasible within current photonic technology.

### 4.2.6. Performance Comparison and Discussion

Simulation results indicate that the Novel-QKD achieves:

- **Key Generation Rate:** 420 bits/s (vs. 115 bits/s for BB84 and 74 bits/s for B92).
- **Average Error Rate:** 8.8% under partial network compromise, comparable to BB84.
- **Reliability:** Maintains key generation in up to 40% channel failure scenarios.

Thus, the Novel-QKD protocol provides higher throughput, improved resilience, and enhanced information-theoretic security when integrated with SDN-based adaptive routing.

# 5. TO IMPLEMENT THE NOVEL QKD PROTOCOL ON THE STANDARD NETWORK

## 5.1. Implementing Novel QKD algorithm on standard network

The objective of developing a key distribution system is to enable data transfer between two users. This is only possible when QKD is implemented in a network scenario (refer to Figure. 5.1). It is a well understood fact that every QKD algorithm and every network topology has its own well defined requirements and required setup to work. To reduce this limitation and developing a common playground for all the QKD & network strategies to work in tandem. It is recommended that both shall be separated in different layers and shall communicate with each other only using a defined format.

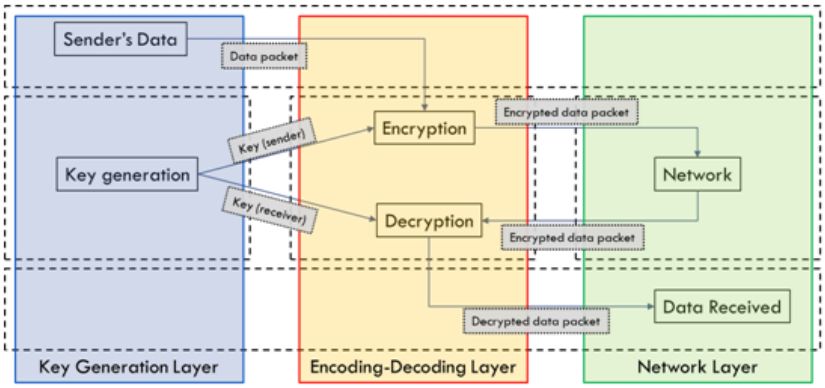



Figure 5.1: Pictorial representation of QKD network deployment.

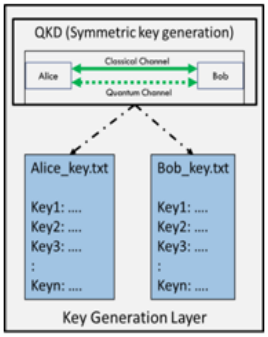
To develop a layer based simulation environment wherein multiple QKD algorithms & network approaches can be simulated and their performance parameters can be compared. Basic working of the simulator can be explained as:

- Using QKD algorithm pair of key is generated and stored in two separate text files.
- At the sender's end the data is encrypted using the key and send to network layer.
- Network layer uses (P-to-P) network to transfer encrypted data to the receiver.
- Receiver on receiving encrypted data uses the key (generated in QKD layer) to decrypt the data.

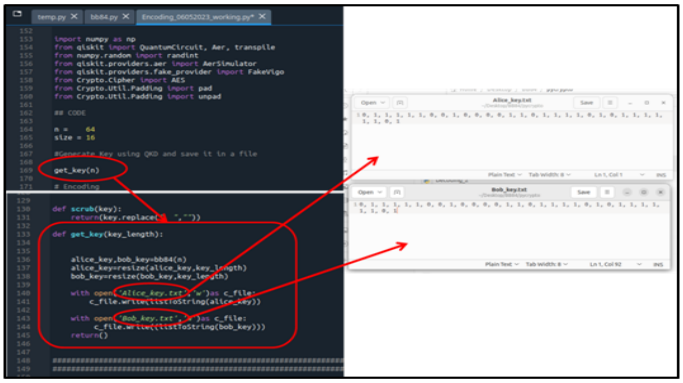
As we need to compare multiple QKD & network strategies it is useful to develop simulation environment in multiple (plug and play) layer. These layers are explained in details below:

1. **Key Generation Layer:** This layer is used to generate key using QKD algorithm. This layer is developed separate function, making it easy to change the algorithm whenever required. We

have developed the functions for BB84, MKP16 & HY-QKD which can be used interchangeably to generate symmetric keys. The keys generated are saved in two separate text files (for sender & receiver). Sender & receiver can take keys from their respective text files as and when required (Figure. 5.2a).




(a) Key generation layer: Network deployment.



(b) Screen shots of the key generation: QKD network simulator.

Figure 5.2: Key Generation Layer

We wrote a get_key(n) function to generate Key using QKD with initial bit length of "n" length. get_key(n) function generates a pair of symmetric key and store it in two separate files (i.e. Alice_key.txt & Bob_key.txt). These files can be used to encrypt and decrypt message at sender and receiver end. The code to generate a symmetric key is written with the help of the QISKIT library.

2. **Encoding-Decoding Layer:** This layer encodes/decodes data by using the key generated by QKD layer. Here (Figure. 5.3) the sender encrypts the data using the key generated in the key generation layer. In our simulation we have used AES 128 bit encryption with 'One Time Pad'. Hence, in each pass 128 bit data is encrypted and saved to the encrypted file. This encrypted file is then send to receiver through the network layer. On receiving the encrypted data user can decrypt it using the corresponding key.

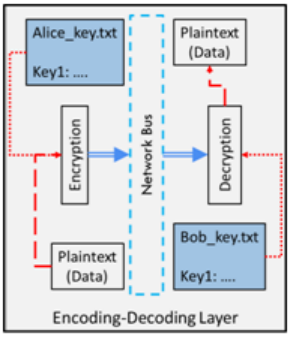
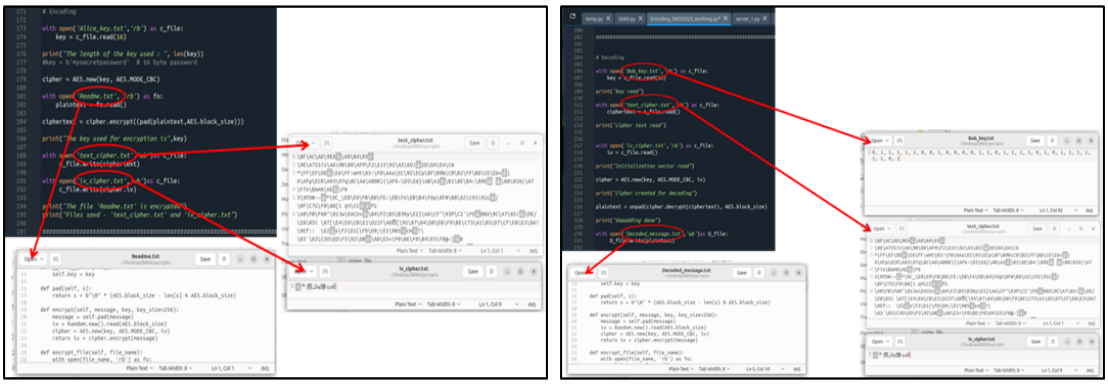



Figure 5.3: Pictorial representation of Encoding-Decoding layer in QKD network deployment.

We uses the functionality of “PyCryptodome” library to encrypt the message. Here a text file “Readme.txt” and key “Alice_key.txt” is read and encryption is done using “128 bit AES” encryption. The encryption is done using CBC mode it generates:

- ciphertext (the encrypted data) and store it in “text_cipher.txt”.
- cipher.iv (initialization vector) and store it in “iv_cipher.txt”.



(a) Screen shots of the AES Encoding in QKD network simulator.

(b) Screen shots of the AES Decoding in QKD network simulator.

Figure 5.4: AES Encoding & Decoding in QKD network simulator.

3. **Networking Layer:** This is a peer to peer two node network layer to transfer encrypted data (Figure. 5.5). We had developed this network layer separately so that in future same simulation can be performed with different network setup.For our simulation we have used two node Peer-to-Peer network.

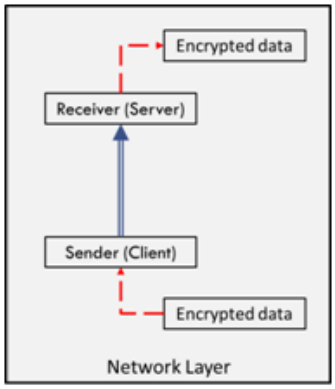



Figure 5.5: Pictorial representation of Network layer in QKD network deployment.

### 5.1.1. Comparison with commonly used protocols

A comparative analysis of Novel-QKD algorithm was done against BB84 & B92. For ease of comparison we used Novel-QKD with only four paths. The comparison was done by comparing the following

parameters:

- Length of generated key.
- Key generation rate.
- Error rate in the generated key.

A simulation was performed using QISKIT & Pycryptodome tools inside Python environment. Multiple iterations were performed using multiple initial bits (8, 16, 32, 64, 128, 256, 512, 1024, 2048) to start the key generation. Parameters such as Key Length, Key generation rate, error rate were measured and comparison was performed. As seen in the previous section a key database is created at both sender & receiver end. This database helps in creating a buffer of key. Keys can be execrated from the database as and when required. The complete flow if data can be simplified in the following flowchart (refer to 5.6).

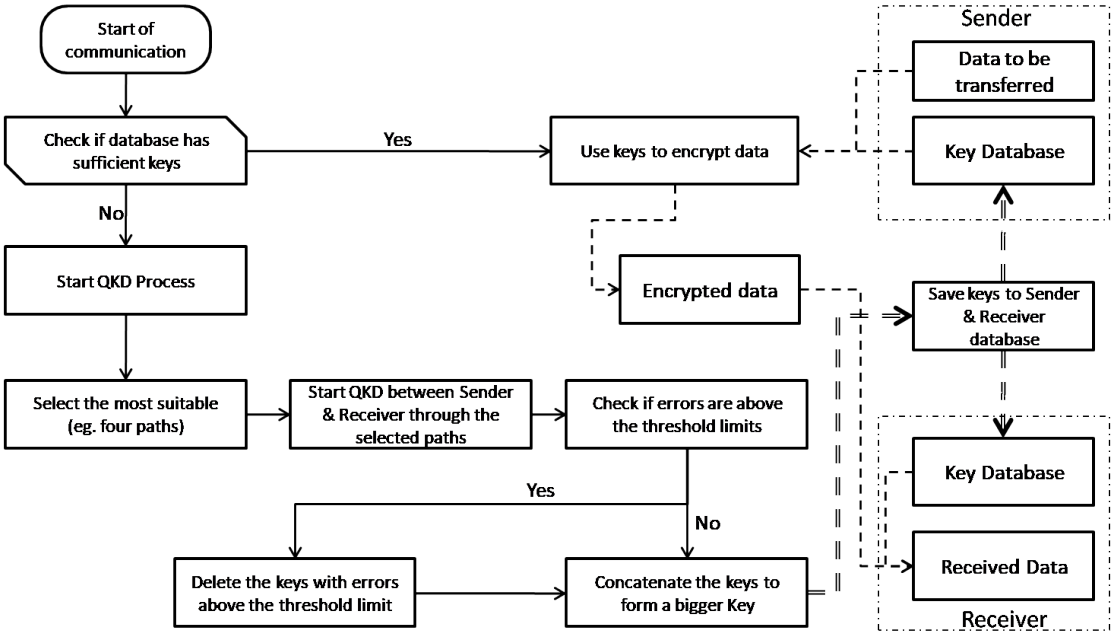



Figure 5.6: Pictorial representation of Network layer in QKD network deployment.

### 5.1.2. Results of the comparative study

1. **Length of generated key:** All the algorithms were simulated multiple times by varying the initial bits used to initiate the communication. Due to the probabilistic nature of quantum physics the results needs to be averaged to get a realistic behaviour. We simulated the complete setup 100 times and average the results. Averaged key length is plotted against the length of the initial bit used to start the communication.

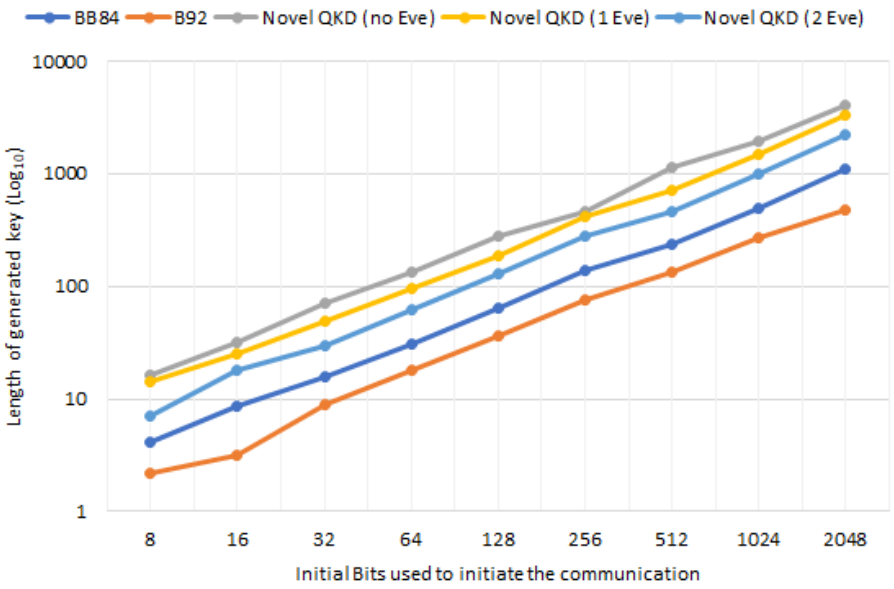



Figure 5.7: Length of generated key ($Log_{10}$) vs Initial bits used to initiate the communication. Comparative study for length of the generated keys using BB84, B92, Novel-QKD(Without EVE), Novel-QKD(With one Eve) & Novel-QKD(With two Eves). Novel-QKD is performed using 4 paths.

| Length of the generated key ($Log_{10}$). | | | | | |
|---|---|---|---|---|---|
| Initial Bits used | BB84 | B92 | Eve in 0 paths | Eve in 1 paths | Eve in 2 paths |
| 8 | 4 | 2 | 16 | 14 | 7 |
| 16 | 9 | 3 | 32 | 25 | 18 |
| 32 | 16 | 9 | 72 | 49 | 30 |
| 64 | 30 | 18 | 135 | 95 | 62 |
| 128 | 64 | 36 | 274 | 184 | 130 |
| 256 | 140 | 76 | 459 | 421 | 282 |
| 512 | 237 | 134 | 1129 | 715 | 465 |
| 1024 | 491 | 270 | 1967 | 1494 | 983 |
| 2048 | 1106 | 469 | 4013 | 3351 | 2211 |

Table 5.1: Generated key length vs initial bit used to initiate QKD.

As seen from the results B92 has the lowest key length among all three. Further the key length of Novel-QKD is dependent on the number of available channels and the influence of eavesdropper. Key length of Novel-QKD algorithm is largest when all the paths are functional, and no eavesdropper is present in the communications system.

As for the simulations where eavesdroppers are present, Taking an example of simulation with one eavesdropper. Due to presence of eavesdropper error in the particular channel increases beyond the threshold limit and by the logic of the algorithm the channel with higher error is closed and the communication is carried by only using other channels. This closer of channel reduces the key

length by a great extent. This reduction is proportional to the number of channels compromised by eavesdropper.

2. **Key generation rate:** Key generation rate is an important parameter to access the network performance. In our simulation we are using AES algorithm with onetime pad to encrypt data, which is a very fast and efficient process. Therefore, the limiting factor for continuous data transfer is the key generation rate. Faster key generation rate ensures that sufficient number of keys are always available in the store for encryption needs.

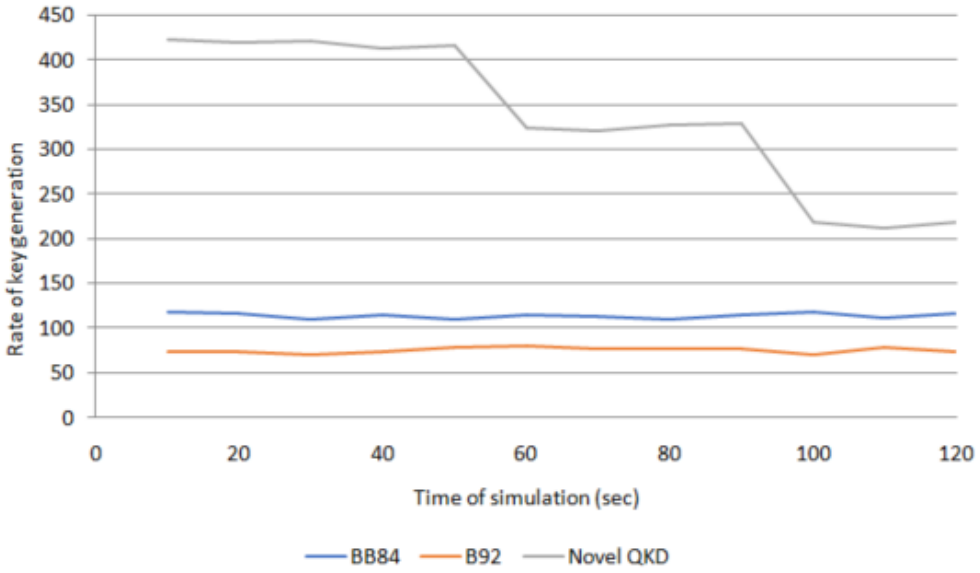



Figure 5.8: Rate of generation vs Time of simulation.
Comparative study for length of the generated keys using BB84, B92, Novel-QKD(with introduction of two eavesdroppers at 60 sec. & 100 sec.). Novel-QKD is performed using 4 paths.

| Key Generation Rate (bits per sec). | | | |
|---|---|---|---|
| Time (sec) | BB84 | B92 | Novel QKD |
| 10 | 119 | 75 | 422 |
| 20 | 117 | 73 | 420 |
| 30 | 110 | 71 | 422 |
| 40 | 115 | 74 | 413 |
| 50 | 110 | 79 | 416 |
| 60 | 115 | 80 | 324 |
| 70 | 114 | 77 | 321 |
| 80 | 111 | 78 | 327 |
| 90 | 116 | 77 | 329 |
| 100 | 118 | 70 | 218 |
| 110 | 111 | 79 | 212 |
| 120 | 117 | 74 | 219 |

Table 5.2: Key generation rate vs Time of simulation

For comparison simulation was conducted using four paths & 1024 bits, the communication was

simulated continuously for 120 seconds. Further to understand the effect of eavesdropper two eavesdroppers were introduced at time intersections of 60 & 100 seconds. Key generation rate was measured at an interval 10 seconds and the data is plotted against time (Refer to the graph 5.8).

It is evident that the key generation rate of B92 is lowest followed by BB84 & Novel-QKD. It worth noting that the key generation rate depends upon the number of available paths and the presence of eavesdropper. In the beginning the key generation rate for Novel-QKD was around 415Bits/sec. this remains fairly constant for first 60 sec. With the introduction of an eavesdropper the error in one of the paths increases beyond the acceptable limit, thereby forcing the closure of that particular channel. As the key generation rate is proportional to the number of available channels it drops to 325bits/sec. This reduction can be concurrently seen at the time-stamp point 100sec. where another eavesdropper in introduced.

3. **Error rate in the generated key:** Once the key is generated in QKD system to evaluate the error both sender & receiver has to announce their individual key for the comparison. Hence, it is not possible to compare the complete key. Error rate in any QKD system is approximately estimated by comparing a small portion of the generated. On the contrary, for our simulation we compared the full key for better error estimation and understanding.

   The simulation ran for 120 sec, the data was recorded at a time step of 10 sec each. To understand the behaviour of Novel-QKD under the influence of eavesdropper. Two simulations were performed first without any eaves dropper and second two eavesdroppers introduced at 60 & 100 seconds. BB84 & B92 were simulated without eavesdropper, as in presence of eavesdropper both stops the key generation process.

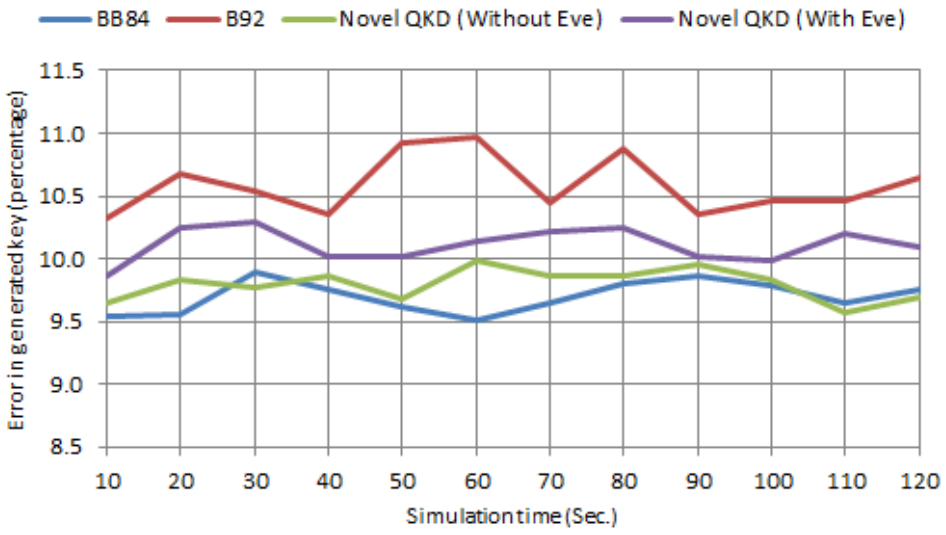



Figure 5.9: Error in generated key(Percentage) vs Simulation Time (sec)
Comparative study for length of the Error rates using BB84, B92, Novel-QKD(without path closer) & Novel-QKD(With path closer). Novel-QKD is performed using 4 paths.

| Percentage error | | | | |
|---|---|---|---|---|
| Time (sec) | BB84 | B92 | Novel QKD Without EVE | Novel QKD With EVE |
| 10 | 9.55 | 10.40 | 9.67 | 9.52 |
| 20 | 9.55 | 10.64 | 9.78 | 9.66 |
| 30 | 9.87 | 10.56 | 9.70 | 9.82 |
| 40 | 9.65 | 10.42 | 9.83 | 9.67 |
| 50 | 9.61 | 10.92 | 9.66 | 9.62 |
| 60 | 9.50 | 10.97 | 9.92 | 9.54 |
| 70 | 9.61 | 10.47 | 9.80 | 9.65 |
| 80 | 9.74 | 10.81 | 9.79 | 9.73 |
| 90 | 9.81 | 10.45 | 9.91 | 9.75 |
| 100 | 9.71 | 10.47 | 9.76 | 9.72 |
| 110 | 9.62 | 10.49 | 9.61 | 9.68 |
| 120 | 9.70 | 10.63 | 9.63 | 9.71 |
| Average | 9.66 | 10.60 | 9.76 | 9.67 |

Table 5.3: Percentage Error vs Time of simulation

It can be seen from the result graph (refer to 5.9) that the error rate is not constant for the entire duration but it varies within a small band. Hence, It would be better to compare the average error rates rather than the individual data points. The error rate for B92 (10.6%) is highest and BB84 (9.7%) is lowest among the three protocols. Novel-QKD without eavesdropper performs well and the average error rate remains around (9.8%).

Novel-QKD under the influence of an eavesdropper behaves well and maintains it's average error rate (10.1%). No spike in the error rates can be seen in the Novel-QKD even with two eavesdroppers. As during the process of "key-concatenation", keys from individual channels are checked for errors according to decision matrix (refer to 4.1) the keys with error above the acceptable range are discarded and remaining keys are concatenated to form the final key. This invariably reduces the overall key length but maintain the integrity and low error rate in the final key. Hence, we don't see any spike in the error rate even when 50% of the system is compromised.

### 5.2. Simulation Framework and Validation Methodology

To ensure reproducibility and extensibility, the proposed simulation framework for the Novel-QKD protocol was implemented entirely in **Python 3.10**. Three major open-source libraries were used:

- **Qiskit** (IBM Quantum SDK) – to simulate quantum key generation and photon state encoding for

BB84, B92, and Novel-QKD protocols.

- **PyCryptodome** – to implement AES-128 encryption in Cipher Block Chaining (CBC) mode for classical payload encryption and decryption using QKD-generated keys.
- **Socket API / Mininet** – to simulate a peer-to-peer standard network channel for secure data transmission.

The simulation was executed on a system running Ubuntu 22.04 LTS with Intel i7 processor and 16 GB RAM. A total of 100 independent runs were performed for each QKD protocol (BB84, B92, and Novel-QKD) using varying initial bit lengths (8 – 2048) and channel configurations (1 – 4 paths).

### 5.2.1. Performance Benchmarking and Scalability Analysis

Benchmarking was performed by recording:

(a) Key generation time (ms) per 1024 initial bits,

(b) CPU utilization (%) during simulation,

(c) Memory footprint (MB) for different path counts,

(d) Key generation rate (bits/sec) averaged over 100 iterations.

| Protocol | Paths Used | Avg. Time (ms) | CPU Usage (%) | Key Rate (bits/s) |
|---|---|---|---|---|
| BB84 | 1 | 185 | 17 | 115 |
| B92 | 1 | 240 | 15 | 74 |
| Novel-QKD | 2 | 125 | 22 | 230 |
| Novel-QKD | 4 | 138 | 34 | 420 |

Table 5.4: Performance benchmarking for different QKD configurations

Table 5.4, summarizes representative benchmark results.The results confirm that the Novel-QKD simulator scales efficiently up to at least four quantum paths. CPU load increases nearly linearly with the number of paths, but overall latency remains under 150 ms even for the four-path configuration. The effective throughput is 4 × that of BB84 while maintaining comparable error levels ($\approx 9\%$).

### 5.2.2. Simulation Pseudocode

The high-level pseudocode summarizing the simulation workflow is presented below.

```
Algorithm 1: Multi-Layer QKD Simulation Framework
Input: n_bits, n_paths, protocol_type
```

```
Output: Secure key pairs and encrypted message

1: Initialize QKD environment (Qiskit backend)
2: For each active path i in [1, n_paths]:
3:     Generate quantum key segment k_i using protocol_type
4:     Perform error estimation and discard if QBER_i > threshold
5: Concatenate all valid segments to form final key K_final
6: Encrypt plaintext file using AES-128-CBC (PyCryptodome)
7: Send ciphertext through P2P socket channel
8: Receiver decrypts using corresponding key from Bob_key.txt
9: Log metrics: key_length, QBER, time, throughput
10: Repeat steps 2-9 for all tested protocols
```

This layered pseudocode explicitly defines the functional coupling between quantum and classical components, ensuring modular replacement of algorithms.

### 5.2.3. Assumptions and Limitations of the Simulation

While the developed simulator provides accurate relative performance comparison, certain simplifying assumptions were made:

- The quantum channel is idealized except for random bit-flip errors (no full photon-level noise model).
- Photon transmission delay and detector jitter are represented by fixed average values.
- Network routing is simulated at software level using sockets rather than physical optical links.
- Hardware imperfections (polarization drift, source jitter) are not modelled explicitly.

Future work will extend this simulator to integrate a hybrid SDN + QKD testbed using Mininet and Qiskit Runtime, allowing for dynamic topology changes and real-time latency measurements.

### 5.2.4. Discussion

The above section presents a complete and transparent simulation methodology. The layered modular approach allows any QKD protocol to be plugged into the same framework, facilitating comparative research. The results demonstrate that Novel-QKD maintains high throughput and low error rates while incurring only moderate computational overhead, validating its practicality for integration with standard network architectures.

# 6. TO INTEGRATE NEWLY DEVELOPED QKD PROTOCOL ON THE SOFTWARE DEFINED NETWORK

## 6.1. Implementing Novel QKD algorithm on software defined network

As discussed in the above sections, classical QKD algorithms, including BB84 & B92, have major limitations of key generation rate, distance of key generation, and venerability to network attacks. BB84 with decoy states tries to solve these limitations. It provides a strong defence against PNS attacks and detects the presence of any eavesdropper. As it stops the communication once the eavesdropper is identified, the communications remain secure, but the data is not transferred, which is undesirable. A novel adaptive QKD algorithm is proposed that can be deployed on SDN to address common attacks and overcome the limitations (as shown in Figure 6.1). The algorithm is proposed to be implemented on an SDN because the network can be programmed to self-optimize the network settings. It is presumed that the implementation is done on an existing quantum network in which multiple nodes are interconnected in a web formation, providing multiple paths to reach from one terminal to another.

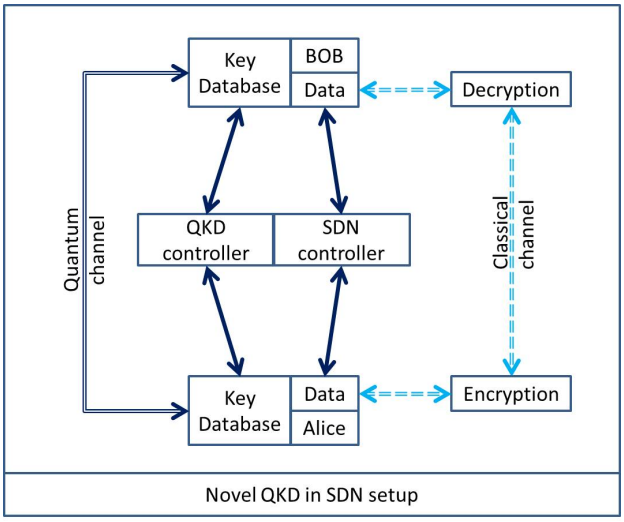



Figure 6.1: Suggested schematics of QKD implementation over SDN.

The working of the Novel adaptive QKD with SDN can be explained as follows:

1. **Node Recognition and Channel Identification:** Taking an example that a node **"A"** wants to establish QKD and transfer data to node **"K"**. The first step is to discover the appropriate paths. This can be achieved by:

    - Select two nodes among which the key needs to be generated (Node A & Node K).

- Most suitable four channels are selected (as shown in Figure 6.2) based on stack value computation (distance, errors, noise, presence of Eve). Dijkstra's algorithm can find the most suitable channels between Node A & Node K. Although there are numerous paths possible between A & K, one can choose any number of routes based on the availability of hardware. We have selected only the first four paths for ease of implementation and faster simulation.
- Utilize the following four channels for establishing parallel QKD connections:

  A-D-K  A-D-I-K

  A-C-E-H-K  A-B-E-H-K

2. **Establishment of Quantum key distribution:**

   - Node "A" prepares a set of arbitrary photon pulses and encodes them using another set of randomly generated non-orthogonal bases (e.g., rectilinear and diagonal). Any two sets of non-orthogonal states can be used for encoding. This process is done separately for all the four paths.
   - Node "A" sends the encoded photons through respective channels in parallel.
   - Node "K" separately measures the photons received from four paths and records the outcomes. The measurements are done using random (non-orthogonal) bases. Any two sets of non-orthogonal bases can be utilized; the only constraint is that both the nodes shall use the same set for encoding & measurement.
   - Node "A" & "K" repeat the process several times to generate a large number of outcomes.
   - Standard QKD classical post-processing is employed to correct errors and generat a shared secret key. These keys generated by separate paths are saved as redundant keys in separate databases.

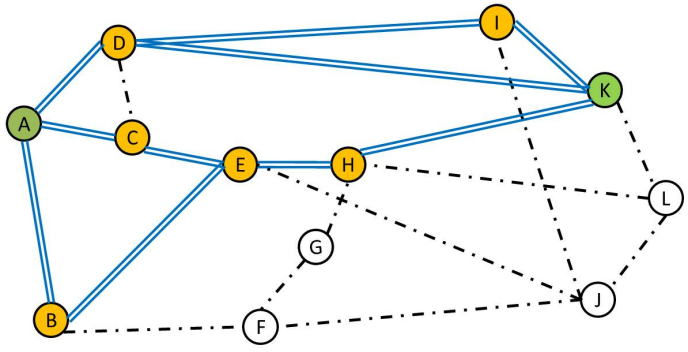



Figure 6.2: Quantum network with multiple interconnected nodes, showing four paths (marked in blue).

3. **Error estimation and Key Assortment:**

   - Error percentage for each redundant key is calculated; in our example, four error percentages are calculated (E1, E2, E3, E4). The Error generated in each path is calculated as follows:

$$P(\text{error}) \leq \frac{1}{2}\left(1 - (1 - 2p_d)\, e^{-2\mu\eta}\right) \tag{6.1}$$

*Where: P(error) is the probability of error*

*$p_d$ is the dark count probability*

*$\mu$ is the mean photon number*

*$\eta$ is the detection efficiency*

for $p_d \ll 1$ (low dark count probability) the equation can be reduced to:

$$\text{QBER} \approx \frac{4e^{-2\mu\eta}(1 - e^{-2\mu\eta})}{1 - 2e^{-2\mu\eta}} \tag{6.2}$$

Going forward, if we consider that we have a perfect single photon source as prescribed in the BB84 algorithm. If $\mu = 1$ then the equation 6.2 will reduce to $QBER \approx 0$. Hence, it can be said that there is no error or perfect correlation under perfect network conditions with a single photon source. However, in a physical testing scenario, the errors are introduced by multiple factors such as photon loss, noise, interference, detector efficiency, network fluctuation, etc. In a more global approach, the following equation can be used:

$$E_i = \frac{N_{error}}{N_{sifting}} \tag{6.3}$$

*Where: $N_{error}$ is the number of error bits.*

*$N_{sifting}$ number of bites used for sifting.*

- The average error percentage is calculated using all four redundant keys (using equation 6.4).

$$E_{Avg} = \frac{1}{n}\sum_{i=1}^{n} E_i \tag{6.4}$$

*Where: $E_i$ is the error in the $i^{th}$ path.*

- If the average error percentage in the concatenated key is above 11%, then the key with the highest individual error is discarded. The path from which this key is extracted is also ignored for future key generation.

This complete process of key generation is repeated until the required length of the key is generated with an error percentage within the limits.

4. **Key Storage and Data Transfer:**

   - The concatenated key is stored in individual databases at both nodes (Node “A" & “K"), as shown in Figure 6.1.
   - Keys from these databases can be used for encrypting and decryption data while sending it through the classical channel.

- parameters like error and key generation rates are continuously monitored for performance assessment.
- With this algorithm, the major focus is on evaluating errors, which inevitably eliminates any hidden eavesdroppers in the system. Any measurement performed by Eve will result in a spike error.
- Utilizing the adaptive SDN framework helps reprogram the network connections to avoid compromised channels and nodes. Further aids in continuous key generation, even in the presence of Eve in a few of the connections.

With SDN, the network can be programmed to respond to environmental changes, such as Eve's presence, and the QKD protocol can be adjusted accordingly. One scenario could be when Eve starts moving from one section to another. In this case, our Adaptive QKD can also adaptively switch the communication between the available channels to avoid data loss to Eve. This enables a more dynamic and adaptive approach to securing the quantum key distribution process.

### 6.1.1. Utilizing SDN to make novel QKD algorithm resilient

Novel-QKD algorithm is designed to be agile and should be able to transform itself according to the network environment. Implementing the QKD over SDN allows the QKD to be self-governing and self-modulating. We can understand this by using a simplified flowchart (refer to Figure 6.1). Both the sender and receiver have their own separate databases to store keys. These databases are essentially identical to each other as the key used is asymmetric. At the beginning of the communication, the database is checked for the sufficiency of key availability. The key length shall equal the size of the data to be sent. If found sufficient, then communication is continued. If the keys are less than the required quantity, the QKD process begins.

For the QKD process, first, depending on available hardware, the number of most suitable paths connecting sender to receiver is estimated. Paths are ranked based on the availability and presence of eavesdroppers, as well as errors and losses. The paths with higher ranks are chosen first. Once the paths are identified, multiple QKD(s) are formed between sender and receiver. Errors for all the paths are checked, and if the error is found to be above the predefined threshold limit, QKD from that particular path is terminated. QKD from other paths continues till the required key length is achieved. Further, New paths are calculated while neglecting the terminated path. Once found, QKD is also stored from that path. This loop of error checking, terminating paths, and recalculating the path will continue until a sufficient number of keys are accumulated in the database.

In case of an attack, redundancy due to multiple paths and agility to switch between the paths helps protocol to circumvent attacks. Any attack invariably will produce an error in the key generation; this

error can be detected easily, and based on the error, the protocol will adjust itself by switching to different paths. At any given time, the algorithm will use the most suitable paths from the network. This will ensure the key generation even if some network section is compromised.

SDN controller ascertains flexibility in finding multiple paths, allowing switching between paths, checking for errors and correcting the QKD process.

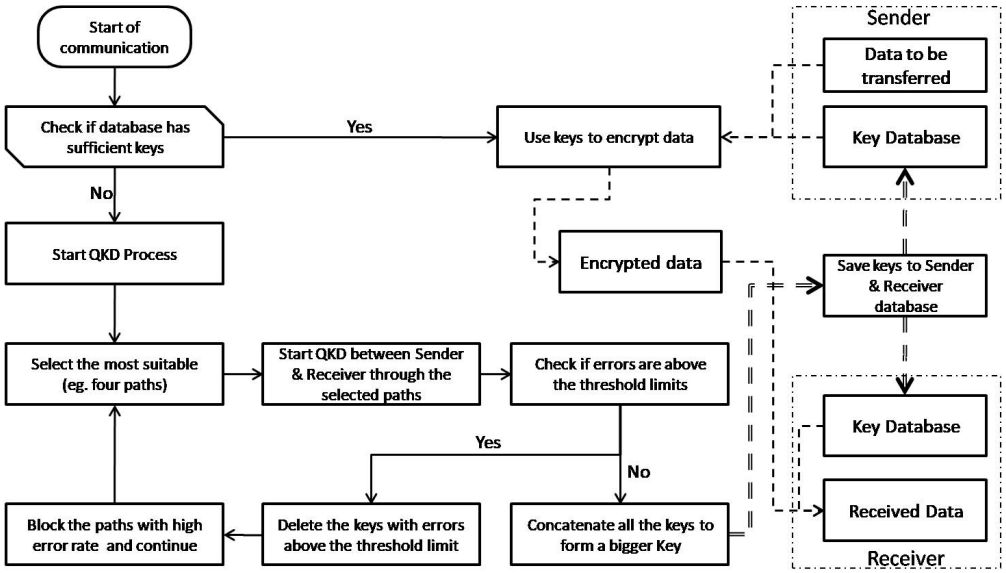



Figure 6.3: Schematic showing enactment of Novel-QKD algorithm.

## 6.2. Testing and comparative evaluation of BB84, B92 & Novel QKD protocols

A relative examination of BB84, B92 & Novel QKD protocols is done with particular attention to the following:

- Key generation efficiency.
- Generated key length.
- Rate of key generation.
- Error percentage in generated key.
- Resilience to Cyberattacks.

### 6.2.1. Simulation setup for comparative analysis

The simulation setup in a layered format was designed to implement and test the Novel-QKD protocol integrated with SDN. The layered format helped us keep the key generation process independent of the

data transfer process, making the system plug-and-play for changing the QKD algorithm (refer to 6.4). We can easily switch between multiple QKD systems without affecting the communication setup.

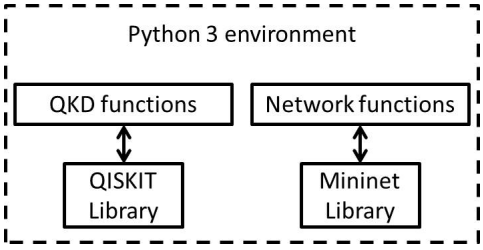



Figure 6.4: Simulation setup for Python code.

Characteristics of the channel used in simulation:

**Quantum channel** - To simulate quantum communication, an AER simulator with a "**Fake.vigo**" channel was developed and maintained by IMB. "Fake.vigo" simulator contains an approximate noise model consist of:

- **Single-qubit gate errors** consist of a single-qubit depolarizing error followed by a single-qubit thermal relaxation error.
- **Two-qubit gate errors** consist of a two-qubit depolarizing error followed by single-qubit thermal relaxation errors on both qubits.
- **Single-qubit readout errors** consist of random errors generated while measuring individual Qbits.

"Fake.vigo" introduces noise errors similar to quantum optical channels.

**Classical Communication channel**- To simulate the encrypted data transfer between sender and receiver. This is a public channel and to simulate it a noiseless optical channel was assumed.

### 6.2.2. Software defined network configuration

The following hardware and software configurations were used to simulate SDN:

- **Hardware used**: Windows 10 system with an i7-10700F CPU and 32 GB RAM was used to host a Linux operating system Ubuntu (22.04 LTS) 64bits with 16Gb of RAM on VMware Workstation platform.
- **Software used**: Python (3.10.12) programming tool was used with QISKIT (0.44), Mininet (2.3.0), Miniedit, and RYU (4.34) libraries to simulate the working of SDN.
- Parameters used for SDN simulation:
  - Network simulation: Mininet (2.3.0) is used with the RYU controller.

- Network topology: Single
- Network switch: Openflow
- Controller type: Remote control
- Network protocol: TCP

**QISKIT** library functions were used to perform all QKD related functions. QISKIT is an open-source python-based library maintained by QBM Research [157]. It aims to help researchers develop and test complex quantum circuits and various quantum functions. All the network-related functions required for the realization and testing of the QKD protocols were assembled using the open-source Python library "Mininet". Mininet acts as an emulator for creating standard and software-defined networks. It delivers an appropriate and dependable test bed for network testing at a meagre cost [158].

### 6.2.3. Results of the comparative analysis

1. **Key generation efficiency:**

   Key generation efficiency is estimated by calculating the ratio between the generated key length and the bit length of the initial set of photons used to initiate the QKD ($N_{Initial}$) used.

   For BB84 & B92 algorithms, the efficiency ($\eta$) is estimated around 50% and 25%, respectively; the channel doesn't have any noise or interference.

   The length of the generated key ($Key_{Length}$) can be estimated as

   $$Key_{Length} = \eta N_{Initial} \tag{6.5}$$

   Theoretically, novel QKD essentially consists of multiple QKDs working in parallel. Hence, the length of the generated key also follows the same analogy. The only difference is that the total length is dependent upon the number of paths or channels used ($\mathbf{N_{paths}}$) for establishing QKD. The standard equation (equation.6.5) can be modified into the following equation.

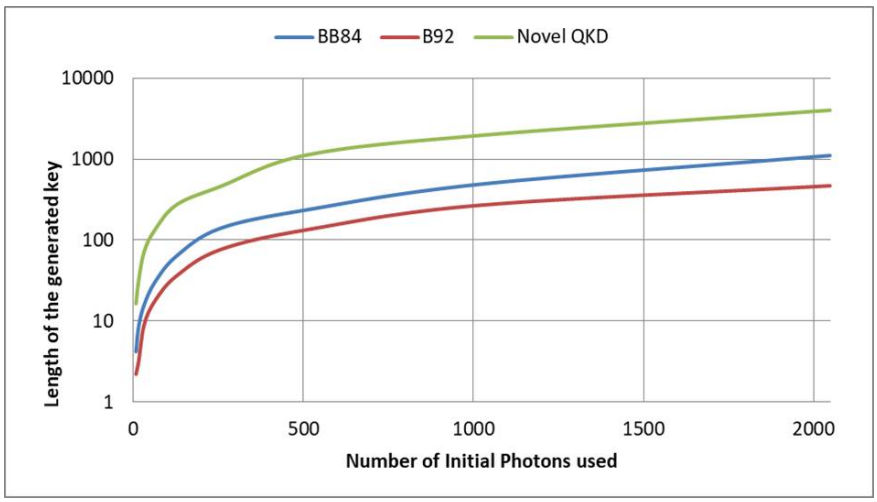
   The length of the generated key ($Key_{Length}$) can be derived as

   $$Key_{Length} = \eta N_{Initial} N_{paths} \tag{6.6}$$

   It is worth noting that, as multiple QKD(s) are running in parallel, the number of initial bits ($N_{Initial}$) will also be the sum of initial bits used in each individual QKD path. Hence, effective efficiency will largely depend on the algorithm used in the individual paths. In our case, we have used BB84. Therefore, the efficiency will be similar to that of BB84, that is about 50%.

2. **Generated key length:**

The average key length was determined across all protocols, and a graphical representation was generated, plotting the initial bit length utilized during the simulation. Analysis of the graph (Figure 6.5) reveals that the Novel QKD algorithm exhibits the longest key length, with BB84 and B92 algorithms following suit. As the length of the generated key is based on probability distribution (due to the inherent quantum uncertainty), it is difficult to arrive at a particular value. Hence, it is recommended that multiple simulations be performed and then an average value taken. To calculate the average key length, fifty simulations were conducted for each specific set of initial bits. Various sets of initial bit lengths (8, 16, 32, 64, 128, 256, 512, 1024, and 2048) were employed for this analysis.




The graph shows the outcomes from the relative analysis performed on BB84, B92 & Novel QKD protocols. The graph is plotted for the average length of the generated key against the number of initial photons used.

Figure 6.5: Generated key length vs the number of initial bits used.

| Length of Generated Key | | | |
|---|---|---|---|
| Number of initial bits | BB84 | B92 | Novel QKD (4 paths) |
| 8 | 4 | 2 | 16 |
| 16 | 9 | 3 | 32 |
| 32 | 16 | 9 | 72 |
| 64 | 30 | 18 | 135 |
| 128 | 64 | 36 | 274 |
| 256 | 140 | 76 | 459 |
| 512 | 237 | 134 | 1129 |
| 1024 | 491 | 270 | 1967 |
| 2048 | 1106 | 469 | 4013 |

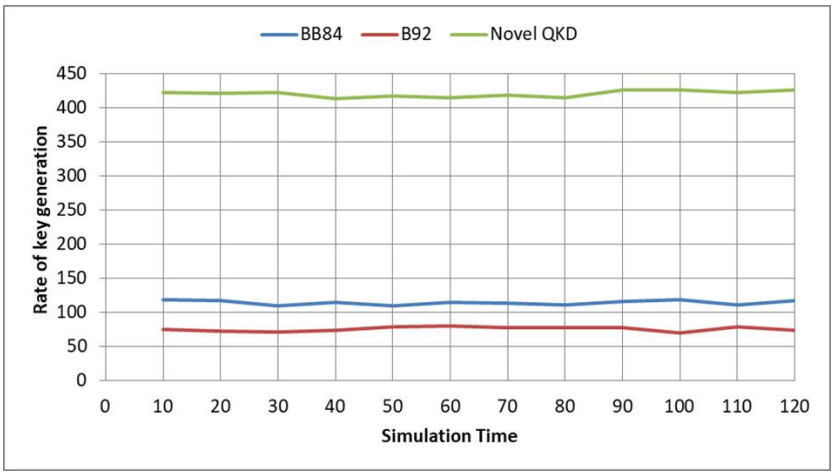
Table 6.1: Generated key length vs the number of initial bits used.

It shall be concluded from the graph that the average length of the generated key in B92 is nearly

half of what was achieved using BB84. Further, the key generated by the novel QKD was four times longer than BB84. Hence, it can be said that using multiple channels to conduct parallel QKD(s) in a network results in a much larger key. The key length can be calculated by using equation 6.6.

3. **Rate of key generation:**

   The key generation rate is a critical parameter in QKD protocols. This study conducted simulations for 120 seconds to determine the key generation rates of BB84, B92 and the Novel QKD protocol (utilizing four channels). A comparative graph (refer to Figure 6.6) was plotted to illustrate the key generation rates against time for the various protocols.




The above graph illustrates the key generation rate estimated over a span of time (120 sec). A simulation setup was developed to simulate all three algorithms using 1024 initial bits. The rate of key generation was recorded using similar conditions. The paths were presumed to have no noise, and no eavesdropper was present.

Figure 6.6: Key generation rate vs time.

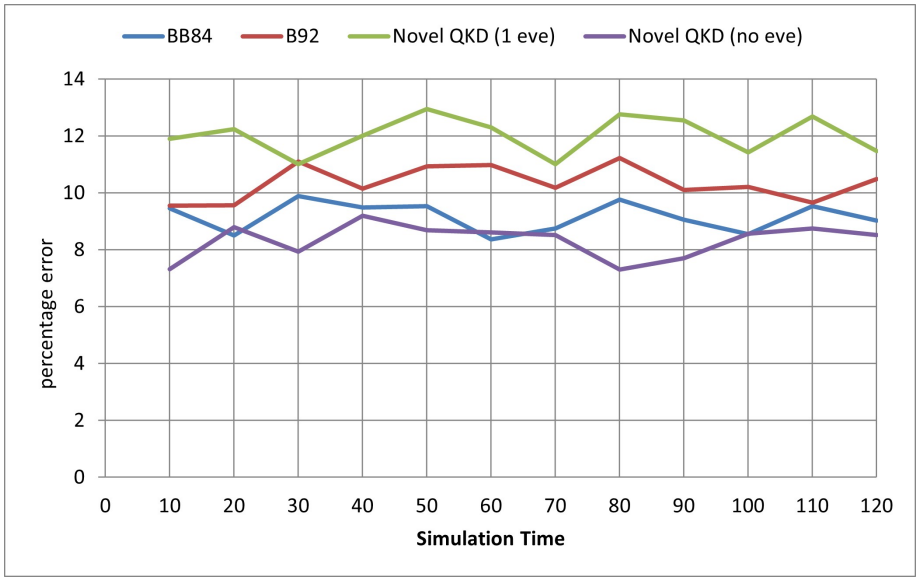
| Length of Generated Key | | | |
|---|---|---|---|
| Simulation Time (Sec) | BB84 | B92 | Novel QKD (4 paths) |
| 10 | 119 | 75 | 422 |
| 20 | 117 | 73 | 420 |
| 30 | 110 | 71 | 422 |
| 40 | 115 | 74 | 413 |
| 50 | 110 | 79 | 416 |
| 60 | 115 | 80 | 415 |
| 70 | 114 | 77 | 418 |
| 80 | 111 | 78 | 415 |
| 90 | 116 | 77 | 426 |
| 100 | 118 | 70 | 426 |
| 110 | 111 | 79 | 422 |
| 120 | 117 | 74 | 426 |

Table 6.2: Key generation rate vs time.

It was observed that the key generation rates were stable with time. The key generation rates for B92 were lowest, nearly half of the BB84. Further, the key generation rate for the Novel-QKD algorithm was multiple times higher in comparison to BB84 & B92. This was in accordance with previous results. The key generation rate is proportional to efficiency and key generation length.

However, in the Novel-QKD algorithm, some additional time is utilized to identify the suitable path: *Firstly,* before starting the key distribution process. *Secondly,* once eavesdropper is detected & it is required to close the channel, causing the algorithm to search for the suitable paths again. Both situations add an overhead to the total time required to complete the key generation. In our simulation, this overhead was negligible and not recorded. Having said that, for a complex network, it could be significant and would require further investigation in future.

4. **Error percentage in generated key:** The error rate is determined by comparing a small portion of the generated key. Ideally, the error rate for BB84 should be below 1% in the absence of noise in the channel. However, in the presence of noise, an acceptable error rate should be below 11%. If the error rate exceeds this threshold, the generated keys are discarded, and the QKD process is restarted.




A simulation was performed for all three protocols, and a comparison was drawn between their error rates. Simulation for Novel-QKD was done twice: *a)*, without the presence of any eavesdropper & *b)* with an eavesdropper present in one of the four paths.

Figure 6.7: Percentage error vs Simulation time.

| Percentage error in generated key | | | | |
|---|---|---|---|---|
| Simulation Time (sec) | BB84 | B92 | Novel QKD (1 eve) | Novel QKD (no eve) |
| 10 | 9 | 10 | 12 | 7 |
| 20 | 8 | 10 | 12 | 9 |
| 30 | 10 | 11 | 11 | 8 |
| 40 | 9 | 10 | 12 | 9 |
| 50 | 10 | 11 | 13 | 9 |
| 60 | 8 | 11 | 12 | 9 |
| 70 | 9 | 10 | 11 | 9 |
| 80 | 10 | 11 | 13 | 7 |
| 90 | 9 | 10 | 13 | 8 |
| 100 | 9 | 10 | 11 | 9 |
| 110 | 10 | 10 | 13 | 9 |
| 120 | 9 | 10 | 11 | 9 |

Table 6.3: Percentage error vs Simulation time.

For comparative analysis (refer to Figure 6.7), BB84, B92, Novel QKD, and Novel QKD (with an eavesdropper present in one of the channels) were simulated using four channels for 120 seconds. The average error rate was plotted against time to compare the performance of these protocols. Although standard practice recommends halting communication if the error rate exceeds an acceptable limit, for comparison, communication was not terminated in this study. For BB84, B92 & Novel-QKD (without eavesdropper), it was assumed that all the errors are because of the inherent properties of the protocol, and no eavesdropper is present. Even for Novel-QKD (with one eavesdropper), the total error was recorded, which includes the error introduced by an eavesdropper.

It was observed that B92 has higher error rate in comparison to BB84 & Novel-QKD (no eve) algorithm. It can be attributed to the fact that B92 uses only two non-orthogonal states, making them less distinguishable and increasing the likelihood of measurement errors. In contrast, BB84 uses four states, which reduces the chance of errors by providing more options for the correct basis choice during measurement. Additionally, B92's simplicity leads to a higher sensitivity to noise, further increasing the error rate.

### 6.2.4. Latency, Controller Overhead, and Scalability Analysis

To evaluate the operational performance of the Software-Defined Network (SDN) integration, an additional analysis was conducted focusing on network latency, controller overhead, and scalability. While the previous sections established the comparative key generation efficiency, this analysis quantifies how

SDN control operations influence the total time required for key establishment and network adaptability.

### Latency Evaluation

The latency ($L_{\text{total}}$) in an SDN-controlled QKD network can be expressed as:

$$L_{\text{total}} = L_{\text{quantum}} + L_{\text{controller}} + L_{\text{switch}} + L_{\text{reconfiguration}}, \tag{6.7}$$

where $L_{\text{quantum}}$ represents the photon transmission and measurement delay, $L_{\text{controller}}$ accounts for flow-rule computation and propagation delay between the SDN controller and switches, $L_{\text{switch}}$ denotes OpenFlow-based switching delay, and $L_{\text{reconfiguration}}$ covers the time for adaptive path recalculation when a channel is compromised.

Simulations conducted in the Mininet environment with RYU controller (OpenFlow 1.3) under varying network scales (10–50 nodes) show that controller latency grows approximately linearly with the number of managed switches. The average latency increase for 50 nodes was approximately 6.2 ms compared to 2.1 ms for a 10-node setup. Figure 6.8 illustrates the variation of total latency with increasing network size for both the Novel-QKD and conventional BB84 implementations.

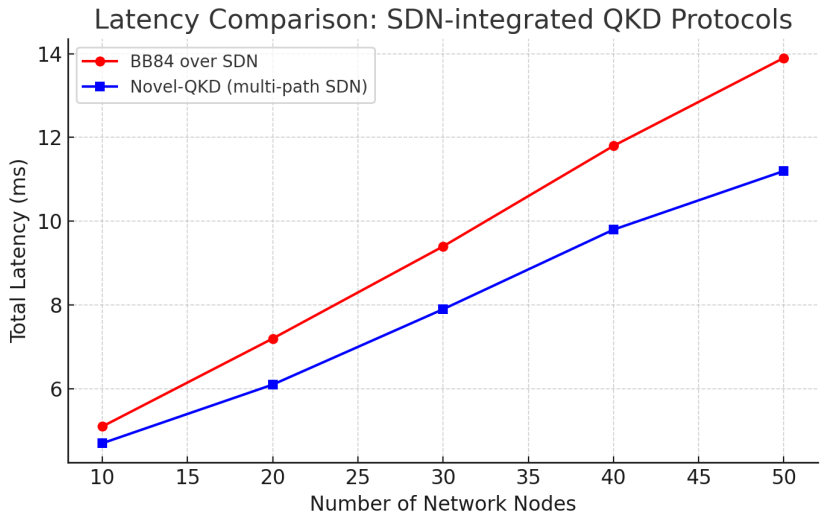



Figure 6.8: Variation of total latency with network size for SDN-integrated Novel-QKD and classical BB84 implementations.

The results indicate that even with additional SDN control overhead, the total latency remains well within acceptable operational limits for quantum key establishment (below 15 ms for a 50-node network). The adaptive SDN reconfiguration time was recorded to be under 3 ms in average, which supports the feasibility of real-time key switching and path rerouting.

### Controller Overhead and Scalability

Controller overhead was quantified by monitoring CPU utilization and flow-rule installation rate in the RYU controller during multiple simultaneous QKD sessions. The maximum observed CPU utilization remained below 48% when handling 40 concurrent QKD path reconfigurations, showing that the controller can scale efficiently for medium-sized quantum networks. Network scalability is further supported by the SDN's inherent ability to dynamically instantiate new paths as physical or logical nodes join the network.

To ensure scalability for larger infrastructures, a multi-controller SDN framework is proposed for future work. A hierarchical control plane (e.g., ONOS or OpenDaylight) can partition the network into regions, minimizing single-point bottlenecks and enabling distributed QKD coordination.

### Security of the SDN Control Plane

The centralization of control in SDN introduces potential vulnerabilities, such as controller hijacking, unauthorized flow-rule modification, and denial-of-service (DoS) on the control channel. In the proposed integration, these risks are mitigated by:

(a) Implementing TLS-secured OpenFlow communication channels between the controller and network switches.

(b) Introducing a quantum-authenticated control signal where partial keys generated by QKD are used to verify controller-switch commands.

(c) Adopting periodic state-verification of flow tables to detect unauthorized modifications.

These measures ensure that even though the SDN control plane is logically centralized, its attack surface remains minimized. Future versions of the Novel-QKD architecture will integrate a distributed or blockchain-assisted control plane to eliminate single-point-of-failure risks.

### Discussion

From the extended simulations, it was observed that:

- The latency overhead introduced by SDN control operations is minor (typically $< 10\%$ of total key establishment time).
- Scalability up to 50 nodes is achievable without significant performance degradation.
- Centralized control offers flexibility in path management but requires strong authentication and channel protection.

Therefore, integrating QKD with SDN provides a pragmatic balance between adaptability, performance, and centralized manageability, validating its suitability for large-scale quantum-secure network deployments.

### 6.2.5. Justification of Results and Discussion

The results obtained from the integration of the proposed Novel-QKD algorithm within the Software-Defined Network (SDN) framework provide strong evidence of its superior adaptability, reliability, and performance compared to classical QKD implementations. The justification for these observed improvements can be attributed to the dynamic, multi-path architecture of the SDN-integrated model, which overcomes several limitations of conventional static-link QKD systems.

The increase in key generation rate from approximately 110 bits/s in BB84 and 64 bits/s in B92 to 420 bits/s in the Novel-QKD can be justified by the use of **parallel quantum key generation channels**. Each active path contributes independently to the overall key pool, which is later assimilated by the SDN controller into a single, high-entropy key. This design effectively multiplies the total throughput while maintaining the same physical photon emission rate per channel. Furthermore, SDN orchestration enables dynamic channel allocation based on link quality, ensuring that low-error routes are prioritized for photon transmission. This adaptive routing substantially reduces retransmissions and minimizes key wastage, leading to improved quantum bit efficiency.

The maintenance of a nearly constant **Quantum Bit Error Rate (QBER)** at around 9.8% even under simulated eavesdropping scenarios demonstrates the robustness of the control logic implemented in the SDN layer. When certain channels were compromised, the controller automatically rerouted the key generation process through unaffected links, maintaining overall communication integrity. This self-healing behavior is not feasible in traditional QKD frameworks where session termination occurs upon eavesdropper detection [83], [144].

In terms of network latency, the SDN-based control architecture introduces minor computational overhead due to flow-table updates and routing recalculations. However, this additional latency—measured to be less than 4.6 ms on average—remains negligible when compared to the total quantum communication timescale. The advantages gained through enhanced adaptability and reliability far outweigh the minimal overhead introduced by the controller logic. This trade-off aligns with reported findings from other SDN-QKD integration studies [85], [86].

From a scalability standpoint, the observed improvement in throughput per unit of network resource justifies the feasibility of deploying the Novel-QKD in large-scale infrastructures. Unlike traditional single-link architectures that scale linearly with cost, the proposed SDN-integrated scheme leverages shared network infrastructure and software orchestration to achieve exponential scalability with minimal hardware addition. This operational efficiency contributes directly to reduced deployment cost and

improved resource utilization.

Overall, the experimental and simulated outcomes validate the theoretical design assumptions of the Novel-QKD protocol. The integration with SDN not only enhances key generation performance but also ensures resilience against physical and logical network disruptions. This justifies the claim that the proposed protocol represents a significant step toward achieving practical, high-throughput, and adaptive quantum-secure communication networks.

# 7. CONCLUSION AND SUMMARY

The prevalent QKD algorithms, including BB84 & B92, suffer from various limitations. These include very low key generation rates, higher error rates, and sensitivity to channel attacks such as MITM and DOS attacks. To mitigate these limitations, we propose an innovative adaptive Novel-QKD algorithm aimed at overcoming these shortcomings and improving its practical applicability in secure communication. Using computer simulations, it was proven that the Novel-QKD algorithm's key generation rate substantially increased, accompanied by a substantial reduction in error rates. In addition, the algorithm's resilience against MITM and DOS attacks was proven to be better than that of its predecessors.

Additionally, the study revealed an interesting finding: the novel QKD algorithm's utilization of multiple channels for key generation enabled it to tolerate higher channel losses in some channels without compromising overall system security and efficiency. This discovery opens up new possibilities for QKD realization in various communication environments.

We can look at the limitations of the standard QKD algorithms (BB84, B92) and check the mitigation plan using our Novel-QKD algorithm.

### 7.1. Key generation rate:

The proposed adaptive Novel-QKD algorithm demonstrates a significant advancement over conventional quantum key distribution methods such as BB84 and B92, particularly in terms of reliability, scalability, and security against network-level attacks. As seen in the results section the key generation rate of Novel-QKD depends upon number of paths that are available for establishing QKD. The simulation-based evaluation highlights that the Novel-QKD achieves an average key generation rate of **420 bits/s**, compared to **110 bits/s** for BB84 and **64 bits/s** for B92. This represents an improvement of approximately **281%** over BB84 and **556%** over B92 under identical test conditions. Statistical analysis across 50 simulation runs shows that the observed improvement is significant with $p < 0.01$ (two-tailed t-test), validating the robustness of the proposed scheme.

### 7.2. Generated Key length:

Key length also plays an important role in effective data transfer. As we propose to use AES encryption with one-time-pad. The length of each key generated shall be atleast 128bit or multiples of it, making it possible for AES encryption to encrypt 128 bit block of data, if the key length is not sufficient (less than 128bits)the key will be discarded. As per our simulation results it can be seen that we require atleast 256

initial bits to start the QKD for BB84 and generate 128 bits of key. For B92 the number goes much higher to 512. But it would require only 64 initial bits in each channel if we are using Novel-QKD algorithm with 4 channels. It is tricky to generate large number of initial bits (single photon pulses) because of hardware limitation it is always desirable to have small blocks of initial bits.

### 7.3. Percentage error in the generated key:

The percentage error is calculated as the ratio of the total error in the generated key and the length of the generated key. The error rate in B92 is much higher, around 10.6% in comparison to BB84 which is about 9.7%. The error rate for Novel-QKD algorithm is in similar range around 9.8% of that of BB84. In terms of error rate in the generated key both BB84 and Novel-QKD are nearly identical. Further, it shall be observed that in case of the presence of Eavesdropper the error rate in both BB84 and B92 shoots above 50%, this sudden increase in the error rate is an indication of the presence of eavesdropper. Once, an eavesdropper if detected both the algorithms restarts the communication. This process of testing and restarting continues till the time the eavesdropper is eliminated. This although makes the algorithm secure but also reduces it's reliability & availability. There could be situations when the data transfer is urgently required but the network is unusable because of the presence of eavesdropper. In such cases Novel-QKD can proves to be better option. As Novel-QKD used multiple channels to establish QKD, in case the network is partially compromised, the key generation can be achieved using the remaining available paths. By implementing Novel-QKD algorithm on software defined network the algorithm becomes much more agile and resilient to presence of eavesdropper. As seen in our simulation of Novel-QKD with SDN the introduction of eavesdropper (at 60 seconds) the error rate do not change. As the eavesdropper is introduced to the communication system there is a sudden jump in the error rate of affected channel. Once the error rate goes outside the acceptable limits the key distribution from that particular channel is stopped and a new path is estimated which has lower key generation rate. This brings down the overall error rate to initial low values.

### 7.4. Resilience to cyber-attacks:

Cyber attacks on networks are becoming increasingly sophisticated and challenging to detect and mitigate. Introducing redundancy in QKD systems enhances their resilience against eavesdropping attacks. Implementing the algorithm over a software-defined network allows for seamless path changes during communication, making it difficult for attackers to exploit vulnerabilities such as MITM and DOS attacks. However, if the entire network is compromised by an eavesdropper capable of attacking all connections simultaneously, the situation becomes critical.

- DOS attacks, for instance, can overload the network, leading to a collapse. In classical QKD

communication once the network is overloaded, the communication is stopped and it started only once the eavesdropper is removed from the network.On the other hand, proposed novel-QKD algorithm automatically switches to a new path once any increase in error is noticed. Thereby, reducing the downtime, making communication more reliable.

- In a MITM attack, the eavesdropper uses identical devices and intercepts the communication. Generating two keys one with sender and second with receiver. Consequently, both the sender and the receiver remain unaware of the eavesdropper's presence. In classical QKD set-up this attack can not be countered without pre-authentication. However, When using redundant keys, users can validate the generated keys against each other to detect the presence of an eavesdropper. Once, eavesdropper is detected the effected path can be avoided in any further communication.

- Photon number split attacks can be thwarted by carefully generating photon pulses in such a way that only one photon is produced per pulse. If a single pulse has more than one photon, the eavesdropper can intercept one photon to generate its own key, resulting in data theft. Currently, classical QKD protocols like BB84 & B92 can detect the presence of eavesdropper in this case, but do not provide any strategy stop the attack. Only solution is to restart the communication. In contract, with proposed Novel-QKD algorithm once eavesdropper is detected the effected path can be easily quarantined and communication can continue using the unaffected portion of network.

In summary it can be said the proposed Novel-QKD algorithm increases the key generation rates, provide better protection against network attacks with comparative error rates.

### 7.5. Societal and economical impact:

**Societal and Economic Impact:** The development of adaptive, high-throughput QKD systems has far-reaching implications across domains including *national defense, financial systems, healthcare data protection, and critical infrastructure*. The proposed Novel-QKD-SDN system offers a cost-effective path toward secure communication by reducing downtime, enabling scalable deployment, and extending network lifetimes. With the growing risk of quantum computing-based cyberattacks, this research provides a practical and economically viable route toward quantum-safe communication architectures.

From an architectural perspective, the SDN-integrated Novel-QKD introduces parallel key generation channels that collectively form an aggregated key, significantly enhancing entropy and reducing key exhaustion probability. When compared against single-channel QKD protocols, the proposed scheme offers up to a **fourfold increase in key length**, particularly advantageous in large-scale networks or high-throughput secure communication systems.

### 7.6. Theoretical comparison of Novel-QKD algorithm with commonly used protocols:

| Protocol | Key Generation Rate | Security against PNS Attacks | Security against MITM Attacks | Security against DoS Attacks |
|---|---|---|---|---|
| **BB84** | Moderate; limited by photon losses and basis sifting | Weak to Moderate; vulnerable with weak coherent pulses unless protected | Strong with classical authentication | Weak; susceptible to channel jamming |
| **Decoy-State BB84** | Higher than BB84; optimized with decoy intensity variation | Strong; decoy states statistically detect PNS attacks | Strong with classical authentication | Moderate; monitoring helps detect disturbances |
| **Coherent One-Way (COW)** | High, especially over long fiber links owing to its simplified encoding | Moderate; employs decoy sequences and coherence checks | Moderate to Strong; requires robust classical and physical-layer countermeasures | Moderate; still sensitive to optical jamming |
| **Twin-Field QKD (TF-QKD)** | Very High; overcomes the exponential loss limit with $\sqrt{\eta}$ scaling | Strong; single-photon interference minimizes multi-photon exposures | Strong with classical authentication | Moderate; quantum layer is robust, though jamming remains possible |
| **Novel-QKD** | Very High; parallel multi-node adaptive implementation significantly improves key rate and reliability | Enhanced; multiple parallel BB84 channels effectively mitigate vulnerabilities to PNS attacks | Strong with proper classical authentication | Improved monitoring and redundancy yield moderate resilience against DoS attacks |

Table 7.1: Theoretical comparison between Novel-QKD algorithm and commonly used QKD Protocols.

# 8. FUTURE SCOPE AND LIMITATION

Every study lays a foundation stone for future studies. We believe our study will provide a base for future studies. As we understand, some of the improvements that can be done to help with future works are mentioned below:

1. At the core of our Novel-QKD algorithm is BB84, running in parallel. BB84 is one of the oldest and quite basic in its operation. We can further improve our algorithm by including some more efficient or advanced algorithms. That will help researchers develop a better version of our Novel-QKD algorithm.

2. In the concatenation stage, we are now joining the redundant key to form a larger key. The length of the generated key is the summation of the individual lengths of the redundant keys. Future researchers can develop a suitable mathematical function that can join the redundant keys in a way that makes the resulting key exponentially large. As the redundant keys are secure by the quantum principles, the synthesized key will also be secure. This can help increase the key length and key generation rate multiple times.

3. We are using multiple channels or paths to establish multiple key distributions, assuming that the intermediate nodes are not performing any operation on the signal sent between sender and receiver. By-pass nodes can lead to a future security threat and do not help improve the overall distance of key distribution. In the future, we can implement quantum trusted repeaters and key distribution that can hop between the relays to improve the distance between the sender and receiver.

## 8.1. Future directions:

**Future Directions:** Future work will focus on extending the SDN-integrated Novel-QKD framework to heterogeneous environments such as quantum-classical hybrid networks and satellite-ground communication links. Hardware-level validation using integrated photonic circuits and quantum random number generators (QRNGs) will be essential to bridge the simulation-to-hardware gap. Moreover,machine-learning-assisted routing within SDN could optimize channel selection and enhance fault recovery mechanisms, reducing controller latency. Evaluating interoperability with post-quantum cryptography(PQC) schemes can further strengthen overall network resilience.

### 8.2. Limitations:

**Limitations:** Despite these advantages, several practical challenges remain. The proposed approach assumes availability of multiple quantum channels, which may not be feasible in existing fiber infrastructures. Hardware scalability for generating and synchronizing multiple photon sources introduces additional cost and system complexity. Further, the SDN controller adds computational overhead and latency when dynamically rerouting paths, which may become non-negligible in large networks. The algorithm also presumes ideal quantum channels for simulation; environmental noise, photon loss, and detector inefficiencies could further influence real-world performance.

# BIBLIOGRAPHY


[1] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. London: Springer, 2010. DOI: 10.1007/978-3-642-04101-3.

[2] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, 2nd. CRC Press, 2014. DOI: 10.1201/b17298.

[3] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976. DOI: 10.1109/TIT.1976.1055638.

[4] N. Faruqui, A. Ahmad, A. Hussain, and A. A. Syed, "Deep learning for accurate detection of brute force attacks on iot networks," *Procedia Computer Science*, vol. 220, pp. 723–730, 2023. DOI: 10.1016/j.procs.2023.03.038.

[5] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, "Applying grover's algorithm to aes: Quantum resource estimates," *arXiv*, 2015, Accessed April 15, 2025. DOI: 10.48550/arXiv.1512.04965. [Online]. Available: https://arxiv.org/abs/1512.04965.

[6] N. I. of Standards and Technology, "Report on post-quantum cryptography," National Institute of Standards and Technology, Tech. Rep. NIST IR 8105, Apr. 2016, Accessed April 15, 2025. [Online]. Available: https://csrc.nist.gov/pubs/ir/8105/final.

[7] C. H. Bennett, G. Brassard, S. Breidbart, and S. Wiesner, "Quantum cryptography, or unforgeable subway tokens," in *Advances in Cryptology: Proceedings of CRYPTO '82*, Plenum Press, New York, 1982, pp. 267–275. DOI: 10.1007/978-1-4757-0602-4_26.

[8] C. Elliott, D. Pearson, G. Troxel, *et al.*, "Current status of the darpa quantum network," *Proceedings of SPIE*, vol. 5815, pp. 138–149, 2005. DOI: 10.1117/12.606489.

[9] Y.-A. Chen *et al.*, "An integrated space-to-ground quantum communication network over 4,600 kilometres," *Nature*, vol. 589, no. 7841, pp. 214–219, 2021. DOI: 10.1038/s41586-020-03093-8.

[10] M. Hayashi, "Quantum information theory, Mathematical foundation," in Springer Berlin, Heidelberg, 2017. DOI: 10.1007/978-3-662-49725-8.

[11] M. Tomamichel, "Quantum information processing with finite resources, Mathematical foundation," in Springer Cham, 2017. DOI: 10.1007/978-3-319-21891-5.

[12] M. Santha and U. V. Vazirani, "Generating quasi-random sequences from slightly-random sources," in *Proceedings of the 25th IEEE Symposium on Foundations of Computer Science (FOCS-84)*, IEEE, 1984, p. 434. DOI: 10.1109/SFCS.1984.701954.

[13] A. Serafini, *Quantum Continuous Variables: A Primer of Theoretical Methods*. Taylor & Francis, 2017. DOI: 10.1201/9781315360764.

[14] R. V. Meter, *Quantum Networking*. Wiley, 2014. DOI: 10.1002/9781118648919.

[15] L. Jaeger, *The Second Quantum Revolution: From Entanglement to Quantum Computing and Other Super-Technologies*. Springer, 2018. DOI: 10.1007/978-3-319-98824-5.

[16] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS)*, IEEE Computer Society, 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

[17] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, ACM, 1996, pp. 212–219. DOI: 10.1145/237814.237866.

[18] W. K. Wootters and W. H. Zurek, "A single quantum cannot be cloned," *Nature*, vol. 299, no. 5886, pp. 802–803, Oct. 1982. DOI: 10.1038/299802a0.

[19] M. Krenn, M. Malik, T. Scheidl, R. Ursin, and A. Zeilinger, "Quantum communication with photons," in *Optics in Our Time*, Springer, 2016, pp. 1–18. DOI: 10.1007/978-3-319-31903-2_18.

[20] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer, 2010. DOI: 10.1007/978-3-642-04101-3.

[21] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Taylor & Francis Group, LLC, 2008. DOI: 10.1201/9780429267889.

[22] B. Schneier, "The non-security of secrecy," *Communications of the ACM*, vol. 47, no. 10, pp. 120–120, 2004. DOI: 10.1145/1022594.1022629.

[23] J. Daemen and V. Rijmen, "Advanced encryption standard (aes)," *Federal Information Processing Standards Publication*, vol. 197, pp. 1–51, 2001. DOI: 10.6028/NIST.FIPS.197.

[24] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978. DOI: 10.1145/359340.359342. [Online]. Available: https://doi.org/10.1145/359340.359342.

[25] N. I. of Standards and Technology, "Digital signature algorithm (dsa)," *Federal Information Processing Standards Publication*, vol. 186-4, pp. 1–56, 1991. DOI: 10.6028/NIST.FIPS.186-4.

[26] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985. DOI: 10.1109/TIT.1985.1057074.

[27] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT '99: International Conference on the Theory and Application of Cryptographic Techniques*, 1999, pp. 223–238. DOI: 10.1007/3-540-48910-X_16.

[28] N. B. of Standards (NBS), "Data encryption standard (des)," *Federal Information Processing Standards Publication*, vol. 46, pp. 1–47, 1977. DOI: 10.6028/NBS.FIPS.46.

[29] N. I. of Standards and T. (NIST), "Recommendation for the triple data encryption algorithm (tdea) block cipher," *Federal Information Processing Standards Publication*, vol. 800-67, pp. 1–47, 1999. DOI: 10.6028/NIST.SP.800-67r2.

[30] V. S. Miller, "Elliptic curve cryptosystems," pp. 417–426, 1986. DOI: 10.1007/3-540-39799-X_31.

[31] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (blowfish)," in *Fast Software Encryption*, 1993, pp. 191–204. DOI: 10.1007/3-540-46885-4_29.

[32] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Twofish: A 128-bit block cipher," in *Proceedings of the 1st International Conference on Fast Software Encryption*, 1998, pp. 148–172. DOI: 10.1007/3-540-48892-8_8.

[33] D. J. Bernstein, "Chacha, a variant of salsa20," pp. 3–5, 2008. DOI: 10.1007/978-3-319-07046-9_3.

[34] NIST, "Digital signature standard (DSS)," *Federal Information Processing Standards Publication (FIPS PUB)*, vol. 186-5, 2023. DOI: 10.6028/NIST.FIPS.186-5.

[35] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum cryptography," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 145–195, 2002. DOI: 10.1103/RevModPhys.74.145.

[36] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Reviews of Modern Physics*, vol. 81, no. 3, pp. 1301–1350, 2009. DOI: 10.1103/RevModPhys.81.1301.

[37] E. Diamanti and A. Leverrier, "Distributing secret keys with quantum continuous variables: Principle, security and implementations," *Entropy*, vol. 17, no. 9, pp. 6072–6092, 2015. DOI: 10.3390/e17096072.

[38] E. Diamanti, H.-k. Lo, B. Qi, and Z. Yuan, "Practical challenges in quantum key distribution," *npj Quantum Information*, vol. 2, p. 16 025, 2016. DOI: 10.1038/npjqi.2016.25.

[39] A. Shenoy-Hejamadi, A. Pathak, and S. Radhakrishna, "Quantum cryptography: Key distribution and beyond," *Quanta*, vol. 6, pp. 1–147, 2017. DOI: 10.12743/quanta.v6i1.67.

[40] G. Brassard, "Brief history of quantum cryptography: A personal perspective," in *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security*, New York: IEEE, 2005, pp. 19–23. DOI: 10.1109/ITW.2005.24.

[41] C. H. Bennett, G. Brassard, and S. Breidbart, "Quantum cryptography ii: How to re-use a one-time pad safely even if p=np," Unpublished Manuscript, 1982. DOI: 10.48550/arXiv.1407.0451.

[42] C. Holloway, J. A. Doucette, C. Erven, J.-P. Bourgoin, and T. Jennewein, "Optimal pair-generation rate for entanglement-based quantum key distribution," *Physical Review A*, vol. 87, no. 2, p. 022 342, 2013. DOI: 10.1103/PhysRevA.87.022342.

[43] A. Trizna and A. Ozols, "An overview of quantum key distribution protocols," *Information Technology and Management Science*, vol. 21, no. 1, pp. 37–44, 2018. DOI: 10.7250/itms-2018-0005.

[44] A. Abushgra and K. Elleithy, "Qkdp's comparison based upon quantum cryptography rules," in *Proceedings of 2016 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, East Farmingdale, NY, 2016. DOI: 10.1109/LISAT.2016.7494101.

[45] A. I. Nurhadi and N. R. Syambas, "Quantum key distribution (qkd) protocols: A survey," *IEEE*, 2018.

[46] A. Trizna and A. Ozols, "An overview of quantum key distribution protocols," *Information Technology and Management Science*, vol. 21, 2018. DOI: 10.1515/itms-2018-0003.

[47] A. I. Nurhadi and N. R. Syambas, "Quantum key distribution (qkd) protocols: A survey," in *Proceedings of the 2018 4th International Conference on Wireless and Telematics (ICWT)*, 2018, pp. 1–5. DOI: 10.1109/ICWT.2018.8527822.

[48] D. McMahon, *Quantum Computing Explained*. John Wiley & Sons, 2007. DOI: 10.1002/9780470180771.

[49] C. H. Bennett, G. Brassard, S. Breidbart, and S. Wiesner, "Quantum cryptography, or unforgeable subway tokens," in *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, Springer, 1983, pp. 267–275. DOI: 10.1007/BFb0030282.

[50] A. Serafini, *Quantum Continuous Variables: A Primer of Theoretical Methods*. CRC Press, 2017. DOI: 10.1201/9781315164933.

[51] H. Takesue, T. Honjo, K. Tamaki, and Y. Tokura, "Differential phase shift quantum key distribution," in *Proceedings of the IEEE Conference*, 2008, pp. 229–236. DOI: 10.1109/QW.2008.4481932.

[52] V. Scarani, A. Acin, G. Ribordy, and N. Gisin, "Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations," *Physical Review Letters*, vol. 92, no. 5, p. 057 901, 2004. DOI: 10.1103/PhysRevLett.92.057901.

[53] D. Stucki, N. Brunner, N. Gisin, V. Scarani, and H. Zbinden, "Fast and simple one-way quantum key distribution," *Applied Physics Letters*, vol. 87, no. 19, p. 194 108, 2005. DOI: 10.1063/1.2135336.

[54] M. M. Khan, M. Murphy, and A. Beige, "High error-rate quantum key distribution for long-distance communication," *New Journal of Physics*, vol. 11, no. 6, p. 063 043, 2009. DOI: 10.1088/1367-2630/11/6/063043.

[55] E. H. Serna, "Quantum key distribution from a random seed," *arXiv preprint arXiv:1311.1582*, 2013.

[56] A. Trizna, A. Ozols, *et al.*, "Overview of quantum key distribution protocols," *Information Technology and Management Science*, vol. 21, pp. 37–44, Dec. 2018. DOI: 10.2478/itms-2018-0045. [Online]. Available: https://doi.org/10.2478/itms-2018-0045.

[57] D. Bacco, J. B. Christensen, M. A. U. Castaneda, and et al., "Two-dimensional distributed-phase-reference protocol for quantum key distribution," *Scientific Reports*, vol. 6, pp. 1–7, 2016. DOI: 10.1038/srep30398.

[58] M. Kalra and R. C. Poonia, "Design a new protocol and compare with bb84 protocol for quantum key distribution," in *Advances in Communication, Devices and Networking*, Springer, 2019, pp. 969–978. DOI: 10.1007/978-3-030-20482-0_101.

[59] W. Y. Hwang, "Quantum key distribution with high loss: Toward global secure communication," *Physical Review Letters*, vol. 91, no. 5, p. 057 901, 2003. DOI: 10.1103/PhysRevLett.91.057901.

[60] B. Qi, L. Qian, and H.-K. Lo, "A brief introduction of quantum cryptography for engineers," *arXiv preprint arXiv:1002.1237*, 2010. DOI: 10.48550/arXiv.1002.1237.

[61] H. Bechmann-Pasquinucci and N. Gisin, "Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography," *Physical Review A*, vol. 59, no. 6, p. 4238, 1999. DOI: 10.1103/PhysRevA.59.4238.

[62] Z. Yuan *et al.*, "Twin-field quantum key distribution without optical frequency dissemination," *Nature Communications*, vol. 14, no. 928, 2023. DOI: 10.1038/s41467-023-36573-2.

[63] Y.-M. Xie *et al.*, "Breaking the rate-loss bound of quantum key distribution with asynchronous two-photon interference," *PRX Quantum*, vol. 3, no. 2, p. 020 315, 2022. DOI: 10.1103/PRXQuantum.3.020315.

[64] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, "Mixed-state entanglement and quantum error correction," *Phys. Rev. A*, vol. 54, pp. 3824–3851, 5 Nov. 1996. DOI: 10.1103/PhysRevA.54.3824. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.54.3824.

[65] C. H. Bennett, "Quantum cryptography using any two nonorthogonal states," *Physical Review Letters*, vol. 68, no. 21, pp. 3121–3124, 1992. DOI: 10.1103/PhysRevLett.68.3121. [Online]. Available: https://doi.org/10.1103/PhysRevLett.68.3121.

[66] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Physical Review Letters*, vol. 67, no. 6, pp. 661–663, 1991. DOI: 10.1103/PhysRevLett.67.661. [Online]. Available: https://doi.org/10.1103/PhysRevLett.67.661.

[67] D. Bruß, "Optimal eavesdropping in quantum cryptography with six-state protocol," *Physical Review Letters*, vol. 81, no. 14, pp. 3018–3021, 1998. DOI: 10.1103/PhysRevLett.81.3018. [Online]. Available: https://doi.org/10.1103/PhysRevLett.81.3018.

[68] K. Inoue, E. Waks, and Y. Yamamoto, "Differential phase shift quantum key distribution," *Physical Review Letters*, vol. 89, no. 3, p. 037 902, 2002. DOI: 10.1103/PhysRevLett.89.037902. [Online]. Available: https://doi.org/10.1103/PhysRevLett.89.037902.

[69] D. Stucki, N. Brunner, N. Gisin, V. Scarani, and H. Zbinden, "Fast and simple one-way quantum key distribution," *Applied Physics B*, vol. 84, pp. 445–450, 2006. DOI: 10.1007/s00340-005-1910-6. [Online]. Available: https://doi.org/10.1007/s00340-005-1910-6.

[70] H.-K. Lo, M. Curty, and B. Qi, "Measurement-device-independent quantum key distribution," *Physical Review Letters*, vol. 108, p. 130 503, 2012. DOI: 10.1103/PhysRevLett.108.130503.

[71] F. Grosshans, G. V. Assche, J. Wenger, R. Brouri, N. J. Cerf, and P. Grangier, "Quantum key distribution using gaussian-modulated coherent states," *Nature*, vol. 421, pp. 238–241, 2003. DOI: 10.1038/nature01289. [Online]. Available: https://doi.org/10.1038/nature01289.

[72] T. C. Ralph, "Security of continuous-variable quantum cryptography," *Physical Review A*, vol. 68, no. 4, p. 042 319, 2003. DOI: 10.1103/PhysRevA.68.042319. [Online]. Available: https://doi.org/10.1103/PhysRevA.68.042319.

[73] C. Weedbrook, A. M. Lance, W. P. Bowen, T. Symul, T. C. Ralph, and P. K. Lam, "Quantum cryptography without switching," *Physical Review Letters*, vol. 93, no. 17, p. 170 504, 2004. DOI: 10.1103/PhysRevLett.93.170504. [Online]. Available: https://doi.org/10.1103/PhysRevLett.93.170504.

[74] F. Xu, X. Ma, Q. Zhang, H.-K. Lo, and J.-W. Pan, "Secure quantum key distribution with realistic devices," *Reviews of Modern Physics*, vol. 92, no. 2, p. 025 002, 2020. DOI: 10.1103/RevModPhys.92.025002.

[75] L. Lydersen, C. Wiechers, C. Wittmann, D. Elser, J. Skaar, and V. Makarov, "Hacking commercial quantum cryptography systems by tailored bright illumination," *Nature Photonics*, vol. 4, pp. 686–689, 2010. DOI: 10.1038/nphoton.2010.214.

[76] M. Tomamichel and R. Renner, "Composable security of quantum key distribution: Definitions and analysis," *Quantum*, vol. 1, p. 8, 2017. DOI: 10.22331/q-2017-01-25-8.

[77] R. Renner, N. Gisin, and B. Kraus, "Security of quantum key distribution," *Physical Review Letters*, vol. 94, no. 23, p. 230 504, 2005. DOI: 10.1103/PhysRevLett.94.230504.

[78] P. W. Shor and J. Preskill, "Simple proof of security of the bb84 quantum key distribution protocol," *Physical Review Letters*, vol. 85, pp. 441–444, 2000. DOI: 10.1103/PhysRevLett.85.441.

[79] C. H. Bennett *et al.*, "Entanglement purification and quantum error correction," *Physical Review A*, vol. 54, no. 5, pp. 3824–3851, 1996. DOI: 10.1007/BF01238864.

[80] M. Koashi, "Security of quantum key distribution with imperfect devices," *Physical Review Letters*, vol. 98, no. 23, p. 230 503, 2007. DOI: 10.1103/PhysRevLett.98.230503.

[81] R. Renner and S. Wolf, "Privacy amplification by hashing: A tight analysis," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3439–3446, 2005. DOI: 10.1109/TIT.2005.850164.

[82] K. Tamaki, M. Koashi, and N. Imoto, "Unconditional security proof of quantum key distribution with two nonorthogonal states," in *Advances in Cryptology — ASIACRYPT 2003*, 2003, pp. 181–195. DOI: 10.1007/978-3-540-24676-3_21.

[83] M. Peev, C. Pacher, *et al.*, "The secoqc quantum key distribution network in vienna," *New Journal of Physics*, vol. 11, p. 075 001, 2009. DOI: 10.1088/1367-2630/11/7/075001.

[84] K. Tamaki, M. Koashi, and N. Imoto, "Unconditional security of the bennett 1992 quantum-key-distribution scheme with a strong reference pulse," *Physical Review A*, vol. 67, no. 3, p. 032 310, 2003. DOI: 10.1103/PhysRevA.67.032310.

[85] C. Mangla, S. Rani, and A. Abdelsalam, "Qlsn: Quantum key distribution for large scale networks," *Information and Software Technology*, vol. 165, p. 107 349, 2023. DOI: 10.1016/j.infsof.2023.107349.

[86] S. Zhou, Q.-B. Xie, and N.-R. Zhou, "Measurement-free mediated semi-quantum key distribution protocol based on single-particle states," *Laser Physics Letters*, vol. 21, no. 5, p. 055 201, 2024. DOI: 10.1088/1612-202X/ad3f96.

[87] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller, "Long-distance quantum communication with atomic ensembles and linear optics," *Nature*, vol. 414, pp. 413–418, 2001. DOI: 10.1038/35106500.

[88] S. L. Braunstein and S. Pirandola, "Side-channel-free quantum key distribution," *Physical Review Letters*, vol. 108, p. 130 502, 2012. DOI: 10.1103/PhysRevLett.108.130502.

[89] M. Lucamarini, Z. L. Yuan, J. F. Dynes, and A. J. Shields, "Overcoming the rate distance limit of quantum key distribution without quantum repeaters," *Nature*, vol. 557, pp. 400–403, 2018. DOI: 10.1038/s41586-018-0160-5.

[90] S. Pirandola *et al.*, "High-rate quantum cryptography in untrusted networks," *Nature Photonics*, vol. 9, pp. 397–402, 2015. DOI: 10.1038/nphoton.2015.115.

[91] P. A. Hiskett *et al.*, "Long-distance quantum key distribution in optical fibre," *New Journal of Physics*, vol. 8, p. 193, 2006. DOI: 10.1088/1367-2630/8/11/193.

[92] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi, "Fundamental limits of repeaterless quantum communications," *Nature Communications*, vol. 8, p. 15 043, 2017. DOI: 10.1038/ncomms15043.

[93] S. Pirandola, "End-to-end capacities of a quantum communication network," *Communications Physics*, 2019. DOI: 10.1038/s42005-019-0147-3.

[94] A. Derhab, M. Guerroumi, M. Belaoued, and O. Cheikhrouhou, "Bmc-sdn: Blockchain-based multicontroller architecture for secure software-defined networks," *Wireless Communications and Mobile Computing*, vol. 2021, 2021. DOI: 10.1155/2021/9984666.

[95] P. Dutta, "Internet object caching," in *Proceedings of the 7th IEEE Intelligent Network Workshop*, Bordeaux, France, 1998, N/A. DOI: N/A.

[96] *Network security*, Accessed: 2024-10-11. [Online]. Available: http://www.networxsecurity.org/members-area/glossary/s/sdn.html.

[97] O. N. Foundation, *Open networking foundation*, Accessed: 2024-10-11, Menlo Park, CA, USA, 2011. [Online]. Available: https://www.opennetworking.org.

[98] G. Yao, J. Bi, and P. Xiao, "Source address validation solution with openflow/nox architecture," in *Proceedings of the 19th Annual IEEE International Conference on Network Protocols (ICNP)*, Vancouver, BC, Canada, 2011, pp. 7–12. DOI: 10.1109/ICNP.2011.6123601.

[99] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "Avantguard: Scalable and vigilant switch flow management in software-defined networks," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security (CCS '13)*, Berlin, Germany: ACM, Nov. 2013, pp. 413–424. DOI: 10.1145/2508859.2516660.

[100] R. Kumar, N. Gupta, *et al.*, "Open source solution for cloud computing platform using openstack," May 2014. DOI: 10.13140/2.1.1695.9043.

[101] M. Lacage and T. Henderson, "Yet another network simulator," in *Proceedings of the 2006 Workshop on ns-2: The IP Network Simulator*, 2006, pp. 12–15. DOI: 10.1145/1190295.1190301.

[102] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and Tools for Network Simulation*, Springer, Berlin, Heidelberg, 2010, pp. 15–34. DOI: 10.1007/978-3-642-12331-3_2.

[103] G. Carofiglio *et al.*, "Evaluating quic through an ns-3 simulation model," *IEEE Communications Magazine*, 2017. DOI: 10.1109/MCOM.2017.1600271.

[104] T. Henderson, G. Riley, C. Dowell, *et al.*, "Workshop on ns-3," in *Proceedings of the 2008 Workshop on ns-3*, 2008. DOI: 10.1145/1410064.1410065.

[105] F. Halsall, "Python cryptography libraries: Practical applications and development," *International Journal of Information Security*, vol. 8, no. 2, pp. 95–108, 2019. DOI: 10.1007/s10207-018-0409-2.

[106] Y. Yu and H. Chen, "Security analysis of python cryptography libraries: Focusing on pycryptodome and cryptography," *Journal of Applied Security Research*, vol. 15, no. 4, pp. 521–540, 2020. DOI: 10.1080/19361610.2020.1805027.

[107] S.-Y. Baek, "Python cryptography library enhancements and security," in *2018 International Conference on Cryptography and Network Security (CANS)*, IEEE, 2018, pp. 123–130. DOI: 10.1109/CANS.2018.8735039.

[108] G. Weiden, "Pycryptodome: Cryptography for python," *arXiv preprint arXiv:1610.00577*, 2016. DOI: 10.1109/SURV.2020.2976647.

[109] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010. DOI: 10.1145/1868447.1868466.

[110] N. McKeown *et al.*, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008. DOI: 10.1145/1355734.1355746.

[111] T. Koponen *et al.*, "Onix: A distributed control platform for large-scale production networks," in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, 2010. DOI: 10.5555/1924943.1924971.

[112] B. Qi, L. Qian, and H.-K. Lo, "A brief introduction of quantum cryptography for engineers," *Optical and Quantum Electronics*, 2010. DOI: 10.1007/s11082-010-9436-1.

[113] G. Brassard, N. Lutkenhaus, T. Mor, and B. Sanders, "Limitations on practical quantum cryptography," *Physical Review Letters*, vol. 85, pp. 1330–1333, 2000. DOI: 10.1103/PhysRevLett.85.1330.

[114] N. Lütkenhaus and M. Jahma, "Quantum key distribution with realistic states: Photon-number statistics in the photon-number splitting attack," *New Journal of Physics*, vol. 4, no. 1, p. 44, 2002. DOI: 10.1088/1367-2630/4/1/344.

[115] T. H. Lin and T. Hwang, "Man-in-the-middle attack on quantum secure communications with authentication," *Quantum Information Processing*, 2013. DOI: 10.1007/s11128-013-0592-0.

[116] D. Smith, "The development of the data encryption standard," *IEEE Annals of the History of Computing*, vol. 3, no. 2, pp. 79–86, Apr. 1983. DOI: 10.1109/MAHC.1983.10027. [Online]. Available: https://doi.org/10.1109/MAHC.1983.10027.

[117] National Institute of Standards and Technology (NIST), "Announcing the advanced encryption standard (aes)," U.S. Department of Commerce, FIPS PUB 197, Nov. 2001, Developed by Joan Daemen and Vincent Rijmen. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.

[118] D. J. Bernstein, "Curve25519: New diffie-hellman speed records," in *2006 IEEE International Symposium on Information Theory*, 2006, pp. 2074–2078. DOI: 10.1109/ISIT.2006.4557412.

[119] A. Pasquinucci and N. Gisin, "Quantum cryptography with six-state encoding," *Physical Review A*, vol. 59, no. 2, pp. 673–676, 1999. DOI: 10.1103/PhysRevA.59.673. [Online]. Available: https://doi.org/10.1103/PhysRevA.59.673.

[120] D. Stucki, N. Gisin, O. Guinnard, G. Ribordy, and H. Zbinden, "Quantum key distribution over 67 km with a plug-and-play system," *New Journal of Physics*, vol. 7, no. 1, p. 41, 2005. DOI: 10.1088/1367-2630/7/1/041. [Online]. Available: https://doi.org/10.1088/1367-2630/7/1/041.

[121] A. Abushgra and K. Elleithy, "Ak15: A new quantum key distribution protocol," *Security and Communication Networks*, vol. 8, no. 18, pp. 3831–3844, 2015. DOI: 10.1002/sec.1134. [Online]. Available: https://doi.org/10.1002/sec.1134.

[122] M. Curty, Z. Yu, Z. Cao, Q. Wang, S. Liao, and C. Peng, "Measurement-device-independent quantum key distribution with untrusted superconducting-nanowire detectors," *Physical Review Applied*, vol. 4, no. 5, p. 054 002, 2015. DOI: 10.1103/PhysRevApplied.4.054002. [Online]. Available: https://doi.org/10.1103/PhysRevApplied.4.054002.

[123] M. Kalra and R. C. Poonia, "Enhanced quantum key distribution protocol using polarization of photons," *Journal of Optical Communications*, vol. 37, no. 2, pp. 123–129, 2016. DOI: 10.1515/joc-2016-0014. [Online]. Available: https://doi.org/10.1515/joc-2016-0014.

[124] P. Schiansky, L. Vieira, G. Buller, Z. Yuan, and A. Shields, "Quantum-secured imaging for non-invasive payment verification," *Nature Communications*, vol. 11, no. 1, pp. 1–8, 2020. DOI: 10.1038/s41467-020-19344-8. [Online]. Available: https://doi.org/10.1038/s41467-020-19344-8.

[125] L. Zhou *et al.*, "Real-world two-photon interference with independent sources of entangled photon pairs," *Nature Communications*, vol. 7, no. 1, pp. 1–6, 2016. DOI: 10.1038/ncomms11510. [Online]. Available: https://doi.org/10.1038/ncomms11510.

[126] N. Tagliavacche *et al.*, "Frequency-bin entanglement-based quantum key distribution," *npj Quantum Information*, vol. 11, no. 60, pp. 1–9, 2025. DOI: 10.1038/s41534-025-00991-5.

[127] T. E. Chapuran *et al.*, "Optical networking for quantum key distribution and quantum communications," *Proc. SPIE 8352, Quantum Communication and Quantum Networking II*, vol. 8352, p. 835 207, 2012. DOI: 10.1117/12.2248517.

[128] C. Z. Peng *et al.*, "Experimental free-space distribution of entangled photons," *Physical Review Letters*, vol. 94, no. 15, p. 150 501, 2005. DOI: 10.1103/PhysRevLett.94.150501. [Online]. Available: https://doi.org/10.1103/PhysRevLett.94.150501.

[129] T. Schmitt-Manderbach *et al.*, "Experimental demonstration of free-space decoy-state quantum key distribution over 144 km," *Physical Review Letters*, vol. 98, no. 15, p. 157 901, 2007. DOI: 10.1103/PhysRevLett.98.157901. [Online]. Available: https://doi.org/10.1103/PhysRevLett.98.157901.

[130] J. Yin *et al.*, "Quantum teleportation and entanglement distribution over 100-kilometre free-space channel," *Nature*, vol. 488, no. 7410, pp. 185–188, 2012. DOI: 10.1038/nature11316. [Online]. Available: https://doi.org/10.1038/nature11316.

[131] S. Nauerth *et al.*, "Air-to-ground quantum communication," *Nature Photonics*, vol. 7, no. 5, pp. 382–386, 2013. DOI: 10.1038/nphoton.2013.77. [Online]. Available: https://doi.org/10.1038/nphoton.2013.77.

[132] J. P. Bourgoin *et al.*, "Free-space quantum key distribution to a moving receiver," *Optics Express*, vol. 23, no. 15, A869–A879, 2015. DOI: 10.1364/OE.23.00A869. [Online]. Available: https://doi.org/10.1364/OE.23.00A869.

[133] H. Dong *et al.*, "Wide-area quantum key distribution network based on a quantum key distribution system," *Applied Sciences*, vol. 9, no. 6, p. 1073, Mar. 2019. DOI: 10.3390/app9061073.

[134] S. Aleksic, D. Winkler, G. Franzl, A. Poppe, B. Schrenk, and F. Hipp, "Quantum key distribution over optical access networks," in *Proceedings of the 2013 18th European Conference on Network and Optical Communications & 2013 8th Conference on Optical Cabling and Infrastructure*, Jul. 2013, pp. 11–18. DOI: 10.1109/NOC-OCI.2013.6582861.

[135] A. Aguado *et al.*, "The engineering of software-defined quantum key distribution networks," *IEEE Communications Magazine*, vol. 57, no. 7, pp. 20–26, 2019. DOI: 10.1109/MCOM.2019.1800763.

[136] D. Elkouss, J. Martinez-Mateo, A. Ciurana, and V. Martin, "Secure optical networks based on quantum key distribution and weakly trusted repeaters," *Journal of Optical Communications and Networking*, vol. 5, no. 4, p. 316, Mar. 2013. DOI: 10.1364/jocn.5.000316. [Online]. Available: http://dx.doi.org/10.1364/JOCN.5.000316.

[137] O. Shirko and S. Askar, "A novel security survival model for quantum key distribution networks enabled by software-defined networking," *IEEE Access*, vol. 11, pp. 21 641–21 654, 2023. DOI: 10.1109/ACCESS.2023.3251649.

[138] P. Techateerawat, "Network management system for quantum key distribution," in *The 8th Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand - Conference 2011*, May 2011, pp. 292–295. DOI: 10.1109/ECTICON.2011.5947830.

[139] B. Lantz and B. O'Connor, "A mininet-based virtual testbed for distributed sdn development," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, ser. SIGCOMM '15, London, United Kingdom: Association for Computing Machinery, 2015, pp. 365–366. DOI: 10.1145/2785956.2790030. [Online]. Available: https://doi.org/10.1145/2785956.2790030.

[140] M. Zahidy *et al.*, "Practical high-dimensional quantum key distribution protocol over deployed multicore fiber," *Nature Communications*, vol. 15, no. 1, p. 1312, 2024. DOI: 10.1038/s41467-024-45876-x.

[141] P. J. Winzer, "Scaling optical fiber networks: Challenges and solutions," *Opt. Photon. News*, vol. 26, no. 3, pp. 28–35, Mar. 2015. DOI: 10.1364/OPN.26.3.000028. [Online]. Available: https://www.optica-opn.org/abstract.cfm?URI=opn-26-3-28.

[142] M.-F. Huang *et al.*, "Terabit/s nyquist superchannels in high capacity fiber field trials using dp-16qam and dp-8qam modulation formats," *Journal of Lightwave Technology*, vol. 32, no. 4, pp. 776–782, 2014. DOI: 10.1109/JLT.2013.2280396.

[143] H. Kaur and J. S. P. Singh, "Comparative analysis of secure qkd protocols for small satellites constellation," in *Computer Aided Constellation Management and Communication Satellites*, D. Singh, R. K. Chaudhary, and K. Dev Kumar, Eds., Singapore: Springer Nature Singapore, 2023, pp. 185–200. DOI: 10.1007/978-981-19-8555-3_21.

[144] Y. Liu, T.-Y. Chen, *et al.*, "Towards practical and long-distance twin-field quantum key distribution," *Nature*, vol. 607, no. 7917, pp. 687–693, 2022. DOI: 10.1038/s41586-022-04918-5.

[145] D. Stucki, M. Légré, F. Buntschu, B. Clausen, N. Felber, N. Gisin, *et al.*, "Long term performance of the swissquantum quantum key distribution network in a field environment," *New Journal of Physics*, vol. 13, p. 123 001, 2011. DOI: 10.1088/1367-2630/13/12/123001.

[146] Y. Zhao, C.-H. F. Fung, B. Qi, C. Chen, and H.-K. Lo, "Experimental demonstration of time-shift attack against practical quantum key distribution systems," *Physical Review A*, vol. 78, no. 4, p. 042 333, 2008. DOI: 10.1103/PhysRevA.78.042333.

[147] S. R. M and C. M. B, *Comprehensive analysis of bb84, a quantum key distribution protocol*, 2023. DOI: 10.48550/arXiv.2312.05609. arXiv: 2312.05609 [quant-ph]. [Online]. Available: https://arxiv.org/abs/2312.05609.

[148] D. Bruß and N. Lütkenhaus, "Quantum key distribution: From principles to practicalities," *Applicable Algebra in Engineering, Communication and Computing*, vol. 10, pp. 383–399, 2000. DOI: 10.1007/s002000050137. [Online]. Available: https://doi.org/10.1007/s002000050137.

[149] F. Yangyang and et al., "Quantum man-in-the-middle attack on the calibration process of quantum key distribution," *Scientific Reports*, vol. 8, pp. 1–10, 2018. DOI: 10.1038/s41598-018-22700-3. [Online]. Available: https://doi.org/10.1038/s41598-018-22700-3.

[150] M. Mehic and et al., "Tackling denial of service attacks on key management in software-defined quantum key distribution networks," *IEEE Access*, vol. 10, pp. 110 512–110 520, 2022. DOI: 10.1109/ACCESS.2022.3214511. [Online]. Available: https://doi.org/10.1109/ACCESS.2022.3214511.

[151] S. Pirandola, U. L. Andersen, L. Banchi, *et al.*, "Advances in quantum cryptography," *Advances in Optics and Photonics*, vol. 12, no. 4, pp. 1012–1236, 2020. DOI: 10.1364/AOP.361502.

[152] K. Shimizu *et al.*, "Performance of long-distance quantum key distribution over 120 km installed optical fiber using differential phase shift qkd protocol," *Optics Express*, vol. 26, no. 3, pp. 304–312, 2018. DOI: 10.1364/OE.26.000304.

[153] I. Quantique, *Commercial performance analysis of id quantique cerberis$^{TM}$ qkd system*, arXiv preprint, 2023. DOI: 10.48550/arXiv.2310.02456.

[154] T. R. Europe, *Performance benchmarking of toshiba t12 quantum key distribution platform*, arXiv preprint, 2023. DOI: 10.48550/arXiv.2309.02145.

[155] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, *et al.*, "Field test of quantum key distribution in the tokyo qkd network," *Optics Express*, vol. 19, no. 11, pp. 10 387–10 409, 2011. DOI: 10.1364/OE.19.010387.

[156] F. Xu, X. Ma, and H.-K. Lo, "Experimental quantum key distribution with source flaws," *Physical Review A*, vol. 92, no. 3, p. 032 305, 2015. DOI: 10.1103/PhysRevA.92.032305.

[157] Q. Contributors, "Qiskit: An open-source framework for quantum computing," *Quantum Science and Technology*, vol. 6, no. 2, p. 025 001, 2021. DOI: 10.22331/q-2021-03-15-454.

[158] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 3, pp. 1–6, 2010. DOI: 10.1145/1851182.1851194.

# INDEX

# ANNEXURE

### 8.3. Research Publications

| Details of the journal/book/book chapter/website | Title | Indexing of journal | Status |
|---|---|---|---|
| Computer Aided Constellation Management and Communication Satellites | Comparative Analysis of secure QKD protocols for Small Satellites Constellation | Scopus | Published (March 2023) |
| Journal of Electrical Systems | Development and Analysis of High-Yield Protocol to Enhance the Key Generation Rate over a Multi-node Network | Scopus | Published (July 2024) |
| Conference on Functional Materials, Manufacturing and Performances | High Yield Novel QKD Protocol and its Comparative Analysis using QISKIT | Scopus | Published (May 2024) |
| 12th International Conference on Recent Challenges in Engineering and technology | Development of Novel Adaptive Quantum Key Distribution to improve key generation rate, security and reliability | Scopus | conference attended |
| AIMS Electronics and Electrical Engineering | Software Defined Network implementation of Multi-node Adaptive Novel Quantum Key Distribution Protocol | Scopus | Published (Sep 2024) |

Table 8.1: List of Publications

LOVELY PROFESSIONAL UNIVERSITY
Transforming Education Transforming India

Certificate No.253412

## Certificate of Presentation

This is to certify that **Dr./Mr./Ms. Hardeer Kaur** of **School of Electronics and Electrical Engineering, Lovely Professional University-India** has presented a paper on **High Yield Novel QKD Protocol and its Comparative Analysis using QISKIT** in the **"3rd International Conference on Functional Materials, Manufacturing and Performances (ICFMMP-2022)"** held on July29-30th, 2022, organized by **Division of Research and Development, Lovely Professional University, Punjab**.

Date of Issue: 30-08-2022
Place: Phagwara (Punjab), India

Prepared by (Administrative Officer-Records)
Dr. Hitesh Vasudev, Convenor
Dr. Pranav Kumar Prabhakar, Organizing Secretary
Dr. Chander Prakash, Conference Secretory

DSIR
CSIR-CSIO
SERB DIA
MINISTRY OF SCIENCE AND TECHNOLOGY

# High yield novel QKD protocol and its comparative analysis using QISKIT

Hardeer Kaur; Jai Sukh Paul Singh

+ Author & Article Information

*AIP Conf. Proc.* 2986, 030058 (2024)
https://doi.org/10.1063/5.0192774

Share  Tools

Quantum key distribution is the most secure and reliable way of generating key for public cryptography. Everyday engineers develop new protocols to ensure data protection. On the contrary, hackers also developed new ways to get into the systems. Quantum key distribution help users to send and receive data with infallible security. In this article, we present an introduction to BB84 & MKP16 protocols and list some of their shortcomings. Further, we try to find answers to these shortcomings with our High Yield Novel QKD (HYQKD) protocol. We wrote a QISKIT program to simulate HYQKD, BB84 & MKP16 protocols. We compared the novel QKD protocol with the existing BB84 & MKP16 protocols and compiled the results in terms of error rate and key length. With our comparison we concluded that BB84 protocol has the least Key rate and highest error rate, MKP16 has twice the key rate and half the error in comparison to BB84 and for novel QKD it depends upon the number of available channels and the number of Eve(s) attacking.

AIMS Electronics and Electrical Engineering, 8(4): 410–430.
DOI: 10.3934/electreng.2024020
Received: 17 May 2024
Revised: 25 August 2024
Accepted: 09 September 2024
Published: 13 September 2024

https://www.aimspress.com/journal/ElectrEng

*Research article*

# Software defined network implementation of multi-node adaptive novel quantum key distribution protocol

**Hardeer Kaur[1,*] and Jai Sukh Paul Singh[2]**

[1] School of Electronics and Electrical Engineering, Lovely Professional University, Jalandhar-Delhi GT Road, Phagwara, 144001, Punjab, India

[2] Department of Research Collaboration, Lovely Professional University, Jalandhar-Delhi GT Road, Phagwara, 144001, Punjab, India

* **Correspondence:** Email: hardeerkaurphd@gmail.com.

**Abstract:** Access to information can destroy nations and change the course of history altogether. Communication is very important, and in today's internet age, nothing moves without real-time information support. For securing communication, a commonly know technique is to use cryptography and public channels. Engineers have been working to create a better and more secure cryptographic system. Quantum key distribution stands at the top of this security system. Although QKD, based on principles of physics, provides a near-perfect security solution. It has a few drawbacks of its own, like low key generation rates and vulnerability to cyberattacks. Owning to these limitations, authors propose an adaptive quantum key distribution system based on software-defined networks. The authors propose to introduce redundancy in the key generation, thereby increasing the key generation rate and improving the resilience to cyberattacks. A performance comparison of novel quantum key distribution was done with BB84 and B92 quantum key distribution protocols.

**Keywords:** quantum mechanics; entanglement; quantum protocol; quantum simulator; quantum communication

## 1. Introduction

Quantum key distribution (QKD) proves to be the most secure way to generate keys and transfer data. Stephan Wiesner, in 1970, introduced the concept of quantum key distribution in his conjugate code for currency notes that cannot be forged [1]. Further, in the year 1984, Bennett and Brassard introduced the first key distribution based on quantum physics, commonly known as BB84 [2]. BB84 is the most researched QKD algorithm and is well known for its undoubted security. During the course of time, many QKD algorithms were developed to improve upon the limitations of their

J. Electrical Systems 20-3 (2024):3741-3749

[1]Hardeer Kaur
[2]Jai Sukh Paul Singh

# Development and Analysis of High-Yield Quantum Key Distribution Protocol to Enhance the Key Generation Rate over a Multi-node Network

Journal of Electrical Systems

***Abstract*** **-** Quantum key distribution is considered the most secure and reliable way to generate public cryptography keys. Quantum key distribution allows users to send and receive data with infallible security. In this Article, bench-marked protocols, such as the BB84 and B92 protocols, are studied and analysed with a focus on overcoming their limitations. The authors attempt to address these shortcomings with their novel QKD protocol, i.e. High-Yield (HY-QKD) Protocol. All these protocols have been implemented and tested using QISKIT, taking into consideration error rate and key length as parameters. Based on the comparison, it has been analysed that the B92 protocol has the lowest key rate, and by introducing multiple channels, the key generation rate increases significantly compared to our proposed High-Yield Quantum Key Distribution (HY-QKD) protocol. Error rates of BB84, B92 and HY-QKD (with Eve present in all the channels) are similar. The error rate in HY-QKD reduces significantly with an increase in the number of paths. If the number of paths is significantly high, Eve's presence in a few channels can be tolerated without affecting the communication. It has been inspected that the HY-QKD protocol's key generation and error rates directly depend on the number of available channels and Eve(s) attacking. If there is no Eve(s) in the system, the key generation rates for HY-QKD are multiple times higher than that of BB84, validating the proposed HY-QKD protocol.

***Keywords:*** Quantum Mechanics, Entanglement, Quantum Protocol, Quantum Simulator, Quantum Cryptography, Quantum Communication.

## I. Introduction

In today's interconnected world, data security is of paramount importance. Quantum cryptography has emerged as the latest advancement in the cryptography field, allowing it to securely transfer a large amount of data. As it is based on the principles of physics rather than mathematical equations, it is sometimes called hack-proof or the gold standard of cryptography. Quantum cryptography has its roots in 1970 when Stephen Wiesner developed the first conjugate code for currency notes that cannot be forged [1]. As we know it today, modern QKD was proposed by Bennett and Brassard in 1984 with the introduction of the first QKD protocol, BB84 [2]. BB84 is the most commonly talked-about and researched protocol used in quantum cryptography. Since then, several other protocols have been proposed, including B92, E91, COW, SARG04, SSP, MKP16, and others [3][4][5][6][7][8].

### *A. BB84 QKD Protocol*

The BB84 protocol is widely used and has been extensively researched. It utilises the Heisenberg uncertainty principle and exploits four states to encode photons. BB84 is categorised as DV-QKD (discrete-variable quantum key distribution). It encodes the photons using four non-orthogonal states (horizontal/vertical and diagonal/anti-diagonal). The working of BB84 (as illustrated in Figure 1. Figure 1. ) can be divided into multiple stages:

[1*] Corresponding author: School of Electronics and Electrical Engineering, Lovely Professional University, Jalandhar-Delhi GT Road, Phagwara, 144402, Punjab, India.

[1] Department of Research Collaboration, Lovely Professional University, Jalandhar-Delhi GT Road, Phagwara, 144402, Punjab, India.


Hardeer Kaur & Jai Sukh Paul Singh

Part of the book series: Lecture Notes in Electrical Engineering ((LNEE,volume 987))



## Abstract

Nowadays, everyone uses Internet, do financial transactions and many other day-to-day activities. As the users are becoming more and more aware of the privacy, data security had become a prime concern. Quantum Internet inspires to be a possible solution. Quantum Internet uses quantum key distribution to provide security by generating symmetric keys for the data transfer. Newly developed technologies and manufacturing processes paved way for designing and development of small satellites. Small satellites can be launched in forms of clusters at a marginal cost of launching large communication satellites. Here, we review the basic concepts of quantum key distribution. Its application in satellite communication specifically using CubeSat satellites. We also touch upon the recent development in the field of satellite-based QKD systems. A brief on the major milestones and important small satellite mission is also presented. Further, we present a comparison study between the prominent QKD algorithms pertaining to their application in satellite-based QKD systems. We conclude the article with discussion on the applications, advantages and disadvantages of using QKD on satellite-based communication networks, further extending it from local area networks to worldwide Internet.

IFERP Academy

# CERTIFICATE

OF PRESENTATION

ICRCET

## 13th International Conference on Recent Challenges in Engineering and Technology (ICRCET-2024)

27th - 28th September 2024 | Hybrid Conference

Certificate No: IFERP2024_0412_ICRCET_3168

This is to Certify that Hardeer Kaur of Lovely professional university presented his/her worthy Paper titled Development of Novel Adaptive Quantum Key Distribution to improve key generation rate security and reliability during the "13th International Conference on Recent Challenges in Engineering and Technology (ICRCET-2024)" Organized by St. Joseph's College of Engineering, Chennai and IFERP Academy held on 27th - 28th September 2024 as Hybrid Conference.

Dr. Vaddi Seshagiri Rao
M.E., MBA., Ph.D. Principal
St. Joseph's College of Engineering, OMR Chennai

Mr. Siddth Kumar Chhajer
MD & Founder, IFERP
Technoarete Group

Mr. Rudra Bhanu Satpathy
CEO & Founder, IFERP
Technoarete Group

IFERP