

Article

Not peer-reviewed version

Recovering Error-Free Cryptographic Keys from Noisy Quantum Key Distribution and Independent Eavesdropper Detection on the Receiving End

[Dina Ghanai Miandoab](#)^{*,†,‡}, [Brit Riggs](#)^{*,†,‡}, [Nicholas Navas](#)[†], [Bertrand Cambou](#)[†]

Posted Date: 3 March 2026

doi: 10.20944/preprints202603.0169.v1

Keywords: quantum key distribution; network security; secure data sharing; error correction; key recovery; intrusion detection; challenge response pair; bit error rate



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Article

Recovering Error-Free Cryptographic Keys from Noisy Quantum Key Distribution and Independent Eavesdropper Detection on the Receiving End

Dina Ghanai Miandoab ^{*,†,‡}, Brit Riggs ^{*,†,‡}, Nicholas Navas [†] and Bertrand Cambou [†]

Northern Arizona University, USA

* Correspondence: dinaghanai@nau.edu (D.G.M.); bmr298@nau.edu (B.R.)

† Current address: Northern Arizona University, School of Informatics, Computing and Cyber Systems, Flagstaff; AZ, 86001, USA

‡ These authors contributed equally to this work.

Abstract

In this paper we study the performance and feasibility of integrating a novel key encapsulation protocol into Quantum Key Distribution (QKD). The key encapsulation protocol includes a challenge-response pair (CRP). In our design, Alice and Bob derive identical cryptographic tables from shared challenges, allowing the ephemeral key to be encoded and recovered without disclosing helper data. Software simulations show error-free key recovery for quantum channel bit error rates up to 40% when using longer response lengths. Additionally, we designed the protocol to detect eavesdropping solely from the statistics of the received quantum stream, without sacrificing key bits for public comparison. We formalize the encoding and decoding model, analyze trade-offs between response length and latency, and report key recovery and error detection performance across different noise levels. The results indicate that this CRP-based multi-wavelength QKD protocol can reduce the reliance on classical reconciliation while preserving security in noisy settings.

Keywords: quantum key distribution; network security; secure data sharing; error correction; key recovery; intrusion detection; challenge response pair; bit error rate

1. Introduction

The establishment of secure and reliable keys is fundamental to protecting data from adversaries. Quantum key distribution (QKD) is a method for key establishment and has emerged as a cornerstone of quantum-safe communication, offering security that remains intact even in the presence of quantum computers. Its unconditional security and ability to detect eavesdroppers through the fundamental principles of quantum mechanics make it an attractive alternative to classical cryptography, which relies on computational hardness assumptions and is increasingly threatened by advances in quantum algorithms.

Despite its theoretical security, the practical implementation of QKD faces significant challenges, specifically regarding noise and data loss. Transmission losses [1–3], environmental interactions, and equipment imperfections introduce noise and data loss that introduce errors and therefore degrade the quantum bit error rate (QBER). These obstacles can limit practical implementation by reducing achievable transmission distances and key rates [4,5]. Another persistent challenge lies in differentiating between errors caused by noisy quantum channels and those introduced by an eavesdropper. While QKD protocols like BB84 can detect Eve's interference when error rates remain below a certain threshold, higher background noise levels can obscure this distinction and can compromise security. Accepting higher error thresholds without sacrificing security requires methods that not only tolerate noise but also maintain the ability to identify Eve's presence. Additionally, limiting the number of

reconciliation steps and the publicly shared information helps maintain security while keeping the protocol efficient.

Most QKD protocols use binary encoding with a single wavelength. However, exploring higher radix encoding and multi-wavelength schemes enables eavesdropping detection through the receiving party's statistics without sacrificing the final key.

Beyond QKD, other security mechanisms address concerns with secure key establishment. One example relevant to this work is challenge-response pair (CRP) schemes. CRP mechanisms enable reproducible, authenticated key derivation without directly storing the key, thus reducing key exposure. Instead, legitimate parties incorporate shared mechanisms such as encrypted files or cryptographic tables as a requirement during the key derivation process. This approach reduces key exposure and limits adversary capabilities because successful interception requires access to the shared mechanisms instead of the transmitted data alone. CRP mechanisms can be incorporated into existing protocols to enable reproducible, authenticated key derivation. This makes CRPs a useful addition to practical QKD, which currently lacks built-in authentication.

In this work, we propose the use of a CRP based ephemeral key encapsulation mechanism integrated with multi-wavelength QKD to address both noisy key recovery and eavesdropper detection. By using CRP-based cryptographic tables, Alice and Bob can recover cryptographic keys in noisy conditions without disclosing helper data. Our work demonstrates error-free key recovery for simulated quantum channels with error rates up to 40%. Furthermore, we show that Eve's activity can be inferred directly at the receiving end (Bob) through statistical analysis of received transmission data, eliminating the need to expose key bits for Eve's detection, as is required in other QKD protocols.

The main contributions of this work are the introduction and implementation of a key encapsulation protocol that combines a CRP mechanism with QKD using higher-radix encoding and two-wavelength transmission. The protocol is evaluated through simulation to quantify how state probabilities change under channel noise and eavesdropping. We establish a baseline for performance and detection capability without sacrificing the final key. This characterization provides practical insight into how quaternary multi-wavelength QKD systems can leverage state statistics for eavesdropping detection. Finally, we discuss the tradeoffs and limitations of the approach, including how protocol parameters influence detection performance, noise tolerance, and the ability of the key encapsulation structure to recover from data loss due to noise.

This paper is organized as follows: Section 2 provides an overview of QKD protocols and related prior work. Section 3 describes the integration of the proposed key encapsulation methodology with a multi-wavelength QKD protocol. Section 3.3 presents the mechanism enabling independent eavesdropper detection at the receiver. Section 5 details the experimental setup and procedures. Section 6 presents the experimental results, while Section 7 analyzes and interprets these findings. Finally, Section 8 discusses potential directions for future research.

2. Background Information

2.1. Quantum Key Distribution (QKD)

In theory, QKD offers unconditional security, resistance to eavesdroppers, and maintains its security even with the development of quantum computers. For these reasons, among many others, there is much interest in QKD development as part of the shift toward quantum-safe cryptography [6–8].

The security of QKD protocols is based on quantum mechanical properties. Because of these properties, QKD protocols remain secure even with commercialization of quantum computers, making it a safer method compared to alternatives that are broken by quantum computers (e.g., Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC)) [6,9–12]. The No Cloning Theorem and properties of quantum measurement ensure security. The No Cloning Theorem ensures that a single quantum state cannot be cloned, meaning that an eavesdropper cannot intercept and make a copy of the data [10]. The properties of quantum measurement ensure that a single photon polarization

state cannot be measured without changing the state in a way that Alice and Bob can detect [9,13]. For quantum entanglement protocols, eavesdropper detection is possible using the Bell Inequality [14–16]. These principles are the basis QKD security.

Noise and error rates remain major challenges for practical QKD implementations, and while post-processing error correction helps mitigate these issues, it comes at a cost to security and efficiency. Transmission distance, hardware imperfections, and environmental interactions all lead to increased communication errors, which consequently limit the achievable key rate. Optical fiber transmission is limited by signal loss over distance, whereas free-space transmission is affected by atmospheric effects. Early implementations demonstrated QKD over distances on the order of tens of kilometers [17]. Twin-Field QKD and post-measurement pairing QKD have enabled longer distances, and more recent QKD systems have achieved distances of hundreds of kilometers [18–21], which is an improvement but the methods introduce more complexity to the protocols. Furthermore, imperfections in devices can lead to security vulnerabilities that attackers could exploit. To address these errors, a variety of post-processing protocols have been developed for error reconciliation [22–24]. Some protocols include Cascade, Winnow, and low-density parity-check (LDPC) codes [25]. These methods require the exchange of redundant information over the classical channel, reducing efficiency, increasing latency, and potentially leaking partial information to adversaries. Other countermeasures also exist. For example, decoy-state protocols and active monitoring improve security but add complexity and cost [26,27]. As a result, there is strong motivation to explore protocol designs that limit the need for extensive error correction.

2.1.1. QKD Protocols

The goal of QKD protocols is to securely generate matching cryptographic keys between two parties Alice (sender) and Bob (receiver). While protocols differ in their details, generally, they aim to securely establish a shared key over an insecure channel.

QKD protocols can be divided into entanglement-based and prepare-and-measure schemes. Entanglement-based protocols use shared entangled states, while prepare-and-measure protocols involve transmitting prepared quantum states for measurement. Despite these differences, both rely on quantum measurement disturbance to detect interference during key distribution.

Generally, QKD protocols consist of two stages using two different channels: the quantum channel communication and the classical channel communication. The quantum channel is used to transmit quantum states, and the classical channel is used for reconciliation and error correction. The quantum channel is affected by noise and loss, while the classical channel is public but must be authenticated. QKD systems rely on distinct quantum and classical channels.

There are many notable QKD protocols [16,28–34,34,35,35–42]. Among them, BB84 [14] is one of the earliest and most extensively studied protocols and has served as a reference point for many subsequent developments in QKD. To illustrate how the proposed protocol can be integrated with existing QKD schemes, we focus on BB84 due to its wide adoption and clear structure. While polarization-based protocols such as BB84 have been associated with limited key rates [43,44], they provide a reference for demonstrating protocol integration. The following section will provide a general overview of BB84, as a reference for discussing our protocol.

2.1.2. BB84 Protocol

BB84 was proposed by Charles Bennett and Gilles Brassard in 1984 [9]. In BB84, Alice sends data to Bob encoded in single-photon polarization states. The data is encoded in a way that allows Alice and Bob to generate matching secret keys from the transmitted photons [14].

The BB84 protocol encodes data in single-photon polarization states, using two bases: rectilinear ($|+\rangle$) with 0° and 90° , and diagonal ($| \times \rangle$) with 45° and 135° [45]. Alice prepares a random bit string, assigns each bit a random basis, and polarizes the photons accordingly. Bob independently measures each photon using a randomly chosen basis. After transmission, Alice and Bob reveal their basis choices over a classical channel and retain only the bits measured in matching bases, which ideally

produce identical values. Figure 1 shows the BB84 encoding scheme. This part of the protocol is called sifting, and the resulting key is called the sifted key.

The next step is error rate estimation. Security is ensured by sacrificing a subset of the key to assess its error rate. If the observed error rate remains below a predefined threshold, the remaining bits are distilled into the final cryptographic key. Otherwise, the session is aborted because errors above the threshold indicate the possible presence of an eavesdropper. This process ensures that any interception leaves detectable traces, and allows QKD protocols such as BB84 to provide secure key establishment under realistic conditions [46].

Basis	Bit	State	
[+]	0	$ 0\rangle$	 0°
	1	$ 1\rangle$	 90°
[x]	0	$\frac{1}{\sqrt{2}} 0\rangle + \frac{1}{\sqrt{2}} 1\rangle$	 45°
	1	$\frac{1}{\sqrt{2}} 0\rangle - \frac{1}{\sqrt{2}} 1\rangle$	 135°

Figure 1. BB84 photon polarization encoding scheme.

Error rate estimation works because Eve’s interception affects the error rate. For Eve to interfere, she must choose one of the two bases and measure the photon. If Eve guesses the same basis as Alice, then Eve should not produce detectable errors. On average, Eve will guess correctly 50% of the time. However, if she guesses incorrectly, Eve will introduce detectable errors. This should result in $50\% \times 50\% = 25\%$ error in Bob’s final key. If enough bits are sent and noise is low, Alice and Bob can detect eavesdropper error when they compare a subsection of their key [47].

2.1.3. QKD Equipment Vulnerabilities

Equipment limitations and imperfections introduce weaknesses in QKD protocols and motivate the need for approaches that can tolerate noise and vulnerabilities [48]. Equipment like single-photon sources may emit multiple or zero photons per pulse. Multi-photon pulses are vulnerable to photon number splitting (PNS) attacks, where Eve measures or stores extra photons for herself and forwards the remaining photons to Bob without detection [8,13,49]. Single-photon detectors can also introduce vulnerabilities, because Eve may exploit imperfect timing biases or manipulate detectors in blinding attacks to extract information [8,49]. Countermeasures such as decoy states and photocurrent monitoring have been proposed to mitigate these threats [49–52], yet equipment imperfections continue to limit practical QKD performance, so methods to address noise and security are needed.

2.2. Error Correction in QKD

In QKD systems, the raw keys created after quantum transmission often have errors because of noise, signal loss, and imperfect equipment. To obtain identical keys, the communicating parties perform a series of classical post-processing steps over the classical channel. These steps include reconciliation and privacy amplification. A fundamental challenge in QKD is error correction because high error rates limit transmission distance, key rates, and feasibility of widespread implementation. Transmission loss increases exponentially over longer physical communication channel distances [2,3,53–55]. Methods to address errors include post-processing error correcting codes, quantum error correction, utilizing decoherence-free states, and quantum repeaters.

2.2.1. Error Correcting Codes

Post-processing error correcting codes and the use of helper data can mitigate errors [56]. These methods can require redundant information for correcting errors, which, in turn, can limit the overall

efficiency and capacity of the quantum channel [57–60]. However, such post-processing methods often add complexity, increase computation time, and can leak information to adversaries [61–63].

Some QKD implementations rely on interactive error reconciliation algorithms such as Cascade and Winnow. These methods work by exchanging information over multiple communication rounds to find and correct bit errors in the key. Despite being effective at error correction, these schemes reveal additional information during communication, which can increase latency and information leakage. Winnow reduces the number of iterations compared to Cascade by using Hamming codes instead of binary search operation, but still requires multiple communication rounds. As a result, there has been significant interest in developing more efficient and less interactive reconciliation methods [22].

Other QKD implementations rely on forward error reconciliation protocols such as low-density parity-check (LDPC) codes. Unlike iterative protocols such as Cascade and Winnow, forward error correction schemes should only require one interaction in theory, and thus, reduce communication rounds and latency. While the LDPC decoding process requires greater computational and memory resources, it significantly reduces communication overhead by limiting reconciliation to a single information exchange. These post-processing tradeoffs motivate improvements that are addressed by our proposed key encapsulation approach.

2.2.2. Other Methods for Handling Errors

Beyond classical post-processing, several approaches attempt to address noise and loss in QKD systems at the physical and network levels. These include quantum error correction, trusted nodes, and quantum repeaters. The following summarizes each method:

1. **Quantum Error Correction:** Quantum error correction takes into account quantum information constraints and is necessary for systems that use quantum data because existing classical computing error correction is not directly applicable due to differences in the behavior of quantum bits [64–66]. Some quantum error correction methods make assumptions about the location of errors in the system, while others do not require these assumptions, at the cost of complexity and number of quantum bits needed [65]. Decoherence-free states are robust against environmental interaction errors and offer another solution [4,5,67].
2. **Trusted Nodes:** Another option for handling errors is to introduce trusted or untrusted relay nodes to read and transmit data between communicating parties and help overcome signal loss, but such a solution comes at the cost of security [55,68,69]. Trusted nodes introduce a security vulnerability because security is breached if any nodes are compromised. Ideally, security should be preserved even if the adversary controls a node. For this reason, untrusted relay node protocols are in development [68]. Because each relay node adds equipment cost and information vulnerability, minimizing the number of nodes is ideal for scalability. Wavelength-division multiplexing (WDM), entanglement QKD protocols, and satellite nodes can assist in the scalability and cost [8,54,55]. However, despite cost improvement, the major drawback of relay nodes is the security risk added by each node.
3. **Quantum Repeaters:** Quantum repeaters are a technology theorized to restore the transmitted quantum information without measurement, and if developed, can enable longer transmission distances [54,68,69]. However, such technology continues to be a challenge that will likely rely on further development of quantum memory [54,55,68–71]. Additionally, even with ideal quantum repeaters, they also introduce additional costs and can be a target for adversary attacks.

With noise as a fundamental obstacle in QKD, error correction methods are vital for widespread implementation, and further research is needed for developing such methods.

2.3. Challenge Response Pair Mechanism

The Challenge Response Pairs (CRP) mechanism is used to verify the identity of a user or device, ensuring that access is granted only to authorized entities. CRP systems consist of input parameters referred to as the challenge and corresponding output values referred to as the response [72,73]. In

typical usage, the authenticating entity (e.g., server) sends a challenge to the entity that is being authenticated (e.g., client). The entity that is being authenticated uses a secret, such as a cryptographic table or encrypted file, to compute a response based on the challenge. The authenticating entity then verifies this response. If it matches the expected response, authentication is successful. This process ensures that the responding entity knows the secret without transmitting it directly.

3. CRP-Based Ephemeral Key Encapsulation and Obfuscation Method for Independent Eavesdropping QKD Systems

In this section, we describe how the CRP-based key encapsulation mechanism proposed in [74,75] can be integrated with QKD. The CRP-based key encapsulation approach provides a mechanism to withstand errors in transmitted ephemeral keys. By embedding this method within a QKD framework, we aim to provide a theoretically grounded approach that can support more reliable key recovery and error correction in future practical QKD implementations.

As a preliminary step, Alice and Bob independently generate matching response tables following a CRP procedure. In this work, both parties start from a shared encrypted file, which serves as the seed material for generating a reproducible set of responses. This shared file guarantees that, when the same CRP function is applied, the resulting response sets are identical.

This property of generating identical responses at both ends provides two key advantages within the protocol:

1. First, the availability of matching responses enables an efficient and reliable key recovery process because the receiving party can independently reconstruct the same cryptographic key without helper data.
2. Second, the CRP procedure acts as a source verification mechanism. This means only a party in possession of the correct shared encrypted file can produce the valid response, which authenticates the origin of the transmitted responses.

Figure 2 illustrates the key distribution protocol. In summary, Alice and Bob generate the same set of responses using the same CRP mechanism. Using her CRP mechanism, Alice generates and encapsulates the ephemeral key in a subset of the full responses and transmits these to Bob by encoding the responses in single photon polarization states. These photons are sent through the quantum channel. Bob extracts the ephemeral key from the transmitted photons using his CRP mechanism. The end result of the protocol is a secure and identical cryptographic key between both parties, Alice and Bob. The next sections will describe the protocol in more detail.

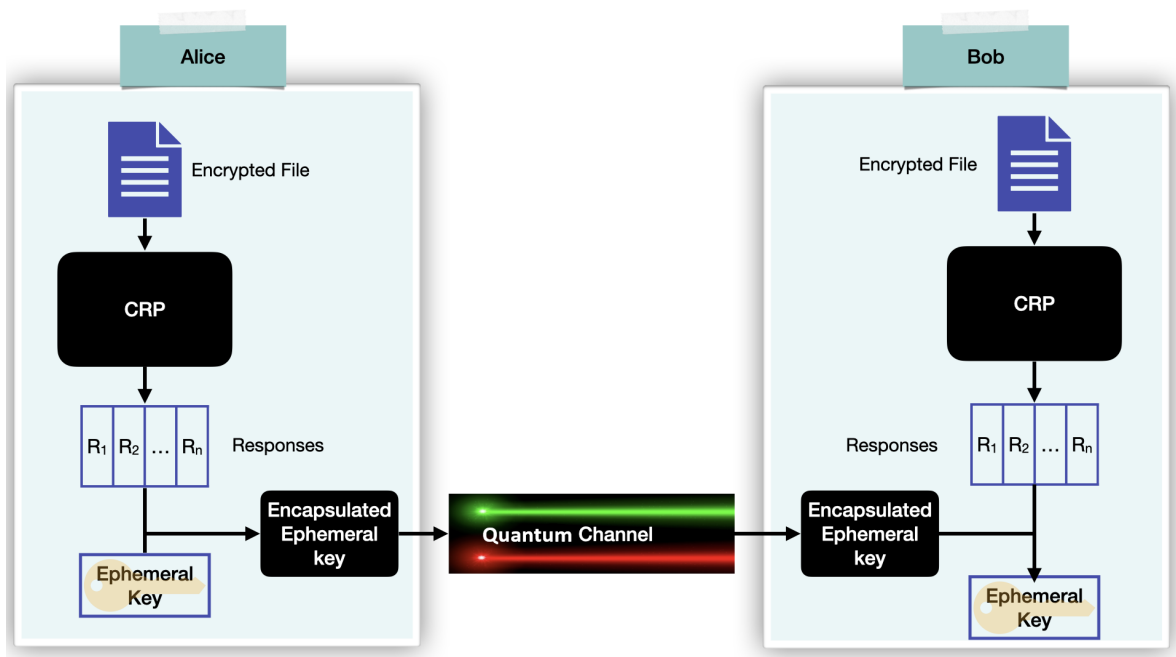


Figure 2. Diagram of the QKD protocol using the CRP mechanism.

3.1. Overview of the CRP-Based Ephemeral Key Encapsulation Protocol

The proposed protocol enables Alice and Bob to establish a shared secret key by combining QKD with a CRP-based key encapsulation mechanism. This approach combines the security of QKD with the reproducibility and authentication properties of CRP-based key generation.

First, Alice produces a random ephemeral key whose length matches the number of responses produced from the CRP process. She then applies the encapsulation method introduced in [74] to securely transmit this ephemeral key to Bob.

Instead of sending the key directly, Alice transmits only a subset of the CRP-generated responses that correspond to the positions of “1” bits in the ephemeral key. Each selected response is encoded into qubits according to a randomly chosen quantum basis and then sent to Bob over the quantum channel.

Upon receiving the qubits, Bob independently generates his own set of responses using the same CRP process, and then proceeds to match the received responses against his locally generated table. After the classical post-processing phase, including error correction, Bob reconstructs the ephemeral key, ensuring both correctness and confidentiality. Figure 2 illustrates the complete flow of this process.

3.1.1. Key Encapsulation and Reconstruction of the Ephemeral Key

1. **Step 1 – Key and Response Generation**
Alice begins by generating a random ephemeral key K , ensuring its bit length matches the number of responses obtained from the CRP process. Each response R_i is derived from the shared encrypted file using a challenge–response mechanism, where the challenge defines the input to the CRP function, and the encrypted file ensures consistency and reproducibility.
2. **Step 2 – Key Encapsulation**
To encapsulate K , Alice selects only those responses whose positions correspond to bits set to “1” in the ephemeral key. This filtered subset $\{R''_{s_j}\}$ represents the encapsulated form of K . Each element of this subset is encoded into qubits and transmitted to Bob over the quantum channel.
3. **Step 3 – Reception and Preliminary Processing**
Bob receives the qubits and measures them in his randomly chosen bases. After publicly exchanging the basis, he disregards all measurement outcomes obtained in mismatched bases and keeps only the outcomes where his measurement basis matches Alice’s basis. This yields a sifted

subset of responses. However, Bob initially does not know which specific bit positions in K these responses represent.

4. Step 4 - Error Detection via Response Matching

After receiving the qubits and converting them into classical responses, Bob performs key reconstruction by comparing each received response against the complete set of CRP-generated responses stored locally. To automate this comparison, Bob employs the error detection algorithm (Algorithm 1), which classifies every received response according to how well it correlates with his locally generated responses.

Algorithm 1 Error Detection Protocol

Input: Subset of responses $\{R''_{s_1}, \dots, R''_{s_n}\}$, full set $\{R_1, \dots, R_n\}$, window γ_0 , threshold T
Output: Lists NoMatches, OneMatch, Collisions
Initialization
1: Set $i = 1, j = 1, \gamma = \gamma_0$
2: Initialize NoMatches, OneMatch, Collisions $\leftarrow \emptyset$
Main Comparison Loop
3: **while** $i \leq n$ **do**
4: Compute $H(R''_{s_j}, R_{i+k})$ for $k = 0, \dots, \gamma - 1$
5: **if** no $H(R''_{s_j}, R_{i+k}) \leq T$ **then**
6: Append all $i + k$ to NoMatches
7: $i \leftarrow i + 1, j \leftarrow j + 1, \gamma \leftarrow 2\gamma_0 - 1$
8: **else if** exactly one $H(R''_{s_j}, R_{i+k}) \leq T$ **then**
9: Append that index to OneMatch
10: $i \leftarrow i + k + 1, j \leftarrow j + 1, \gamma \leftarrow \gamma_0$
11: **else**
12: Identify $k_{\min}, k_{\max}$ where $H(R''_{s_j}, R_{i+k}) \leq T$
13: Append all corresponding indices to Collisions
14: $i \leftarrow i + k_{\min} + 1, \gamma \leftarrow \gamma_0 + (k_{\max} - k_{\min})$
15: $j \leftarrow j + 1$
16: **end if**
17: **end while**
Return: NoMatches, OneMatch, Collisions

The algorithm operates within a dynamically adjustable search window and computes the Hamming distance between each received response R''_{s_j} and every candidate R_i in that window. If no distances fall below the predefined threshold T , the response is tagged as a *No Match*; if exactly one candidate is within the threshold, it is recorded as a *One Match*; and if multiple candidates qualify, a *Collision* is logged.

The output of the algorithm consists of three lists—NoMatches, OneMatch, and Collisions—that summarize the classification results for all received responses.

Each OneMatch directly maps to a bit value of “1” in the ephemeral key, revealing a reliable correspondence between Alice’s transmitted subset and Bob’s generated table. The NoMatches and Collisions entries, on the other hand, identify error bits that require further processing in the next phase.

5. **Step 5 - Key Recovery** Following the initial reconstruction, Bob analyzes the algorithm’s output to identify uncertain bits in the key. Positions recorded in the No Match or Collision categories indicate potential bit errors or alignment uncertainties introduced by channel noise, photon loss, or any other environmental effect. To correct these errors, Bob employs bit flipping and hash comparison. He sequentially flips one or more of the erratic bits within the uncertain regions and recomputes the hash of the reconstructed key $H(K')$. This hash is then compared against the reference hash of Alice’s ephemeral key $H(K)$, which was securely shared through the classical channel. If the hashes do not match, Bob continues flipping different combinations of the uncertain bits, re-evaluating the hash after each iteration. The process repeats until $H(K') = H(K)$, indicating that the correct key has been successfully reconstructed. See Algorithm 2 for further details.

Algorithm 2 Key Recovery

```
1: Input:  $N$  (length),  $C$  (ciphertext),  $NoMatches[]$ ,  $OneMatch[]$ ,  $Collisions[]$ 
2: Initialize  $K$  as an array of zeros with length  $N$ 
3: for each position  $pos$  in  $OneMatch[]$  do
4:   Flip the bit at position  $pos$  in  $K$ 
5: end for
6: if  $NoMatches[]$  and  $Collisions[]$  are empty then
7:   Attempt to decrypt  $C$  with  $K$  to get  $M$ 
8:   Output:  $K$  if  $M$  is a valid plaintext
9: else
10:  Combine  $NoMatches[]$  and  $Collisions[]$  into a list of lists  $Combined[]$ 
11:  for each combination  $combo$  of bits from  $Combined[]$  do
12:    Flip the bits in  $K$  according to  $combo$ 
13:    Attempt to decrypt  $C$  with new  $K$  to get  $M$ 
14:    if  $M$  is a valid plaintext then
15:      Output:  $K$ 
16:      break
17:    end if
18:  end for
19: end if
20: Output: Failed to find the key
```

3.1.2. Transmission of Quatrits with Two Wavelengths

While many QKD implementations encode binary (two states) information into single photon polarization using a single wavelength, our work encodes quatrit (four states) information using two wavelengths. To support the four state quatrits ($z_i \in \{0, 1, 2, 3\}$), a single wavelength is insufficient because polarization alone offers only two orthogonal states within each measurement basis. To overcome this limitation, the proposed protocol employs a two-wavelength encoding scheme that doubles the number of distinguishable quantum states to encode data in. Each wavelength carries two orthogonal polarizations, allowing the combined system to represent four quatrits.

In this scheme, the two wavelengths are represented by distinct colors (green and red). Within each wavelength, photons are polarized according to one of two bases: the rectilinear basis (+), corresponding to horizontal (0°) and vertical (90°) polarizations, and the diagonal basis ($\times$), corresponding to 45° and 135° polarizations.

Table 1. Polarization states associated with qubit polarization for encoding, where the color green and red represents wavelength color.

z_i	+	$\times$
0	0°	45°
1	90°	135°
2	0°	45°
3	90°	135°

3.2. Data Obfuscation

The protocol employs an obfuscation mechanism that masks the transmitted data stream and enables eavesdropper detection without revealing any portion of the secret key. This obfuscation technique is based on modular arithmetic and operates on subset of responses generated from the ephemeral key encapsulation. The input to the obfuscation process can be represented as binary sequence of length k bits,

$$\{x_1, x_2, \dots, x_k\}, \quad x_i \in \{0, 1\}.$$

This binary sequence is combined with a pre-shared stream of k quatrils (sometimes referred to as the nonce)

$$\{y_1, y_2, \dots, y_k\}, \quad y_i \in \{0, 1, 2, 3\},$$

which is known to both Alice and Bob. This pre-shared stream or nonce can also be replaced with a physical unclonable function or cryptographic table.

For each position i , the transmitted symbol z_i is computed as

$$z_i \equiv (x_i + y_i) \bmod 4. \quad (1)$$

The resulting sequence $\{z_1, z_2, \dots, z_k\}$ therefore represents an obfuscated stream of quatrils to then be encoded into qubits (i.e., single photon polarization states) and sent through the quantum channel.

This modular transformation maps the original binary response state space into four state system, thereby masking the underlying key information.

When $x_i = 0$, there is no effect of modular addition with y_i , and the transmitted value is

$$z_i = y_i.$$

When $x_i = 1$, the operation shifts the shared quatril value by one position within the symbol set $\{0, 1, 2, 3\}$,

$$z_i \equiv (y_i + 1) \bmod 4.$$

with the modulo addition, ensuring that the result remains within the valid quatril range.

Without knowledge of the shared stream y_i , an eavesdropper cannot distinguish whether a given transmitted symbol corresponds to $x_i = 0$ or $x_i = 1$, preventing direct inference of the original data.

At the receiver's end, the inverse operation is applied to obtain the original stream:

$$x_i \equiv (z_i + 3y_i) \bmod 4. \quad (2)$$

The equation 2 corresponds to combining the received sequence z with the shared stream $3y$ (the modular inverse of y) to retrieve the original bit sequence x .

Figure 3 illustrates the complete encoding and decoding process within the proposed QKD architecture. The shared quatril stream y functions as a cryptographic mask that conceals the transmitted key information while simultaneously introducing a structured dependency between the transmitted and recovered symbols. Any unauthorized measurement or interception disrupts this structure and results in error patterns exceeding those predicted by the estimated quantum channel noise. Such deviations provide a quantifiable indicator of eavesdropper activity on the channel.

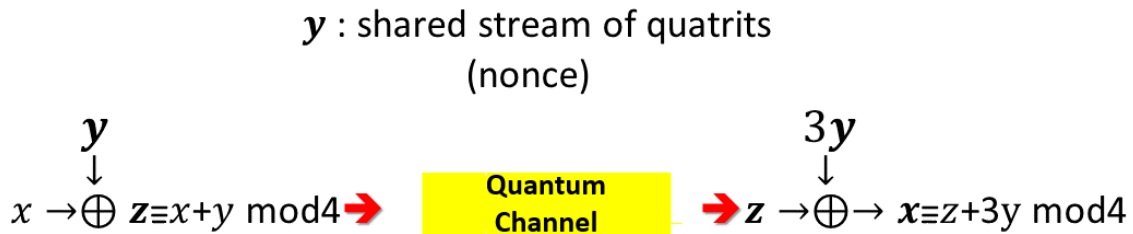


Figure 3. Schematic representation of the key encoding and decoding process in the obfuscation protocol. The shared stream of quatrils, denoted as y , is combined with the initial key stream x using the XOR operation to create z , which is transmitted through the quantum channel (QC). The receiver then applies the reverse operation with the shared stream $3y$ to retrieve the original key stream x .

3.3. Eavesdropper Detection Based on Probability Analysis

One of the challenges in QKD is distinguishing between errors caused by quantum channel noise and those deliberately introduced by an eavesdropper [76]. In protocols such as BB84, eavesdropper

detection is performed by measuring the error rate introduced by an intercept-resend attack, which typically increases the QBER by about 25%. However, noise due to factors such as the channel, equipment, or environment can obscure Eve's presence, making it difficult to determine whether the observed errors arise from non-eavesdropper sources or from eavesdropping because both cause the QBER to increase.

In this work, eavesdropper detection is performed through a probability analysis that exploits the structure of the proposed encoding scheme.

3.3.1. Quantum Channel Error Analysis

In this work, we assume a noisy quantum channel model in which polarization flips between complementary states. Under this model, quatrit symbols $\{0,1\}$ form one complementary pair and $\{2,3\}$ form the other. A channel error therefore causes a transmitted symbol to flip only to its complementary counterpart.

These transitions are deterministic for a given wavelength and measurement basis: each transmitted state can only transition to one other state. For example, in the rectilinear basis, a transmitted symbol $z_i = 0$ (0°) can only flip to $z_i^* = 1$ (90°), and vice versa. Similarly, in the diagonal basis, $z_i = 2$ can only flip to $z_i^* = 3$, and $z_i = 3$ can only flip to $z_i^* = 2$.

Therefore, these transition limitations result in a reduced set of potential combinations that Bob can receive, which will be elaborated upon in the following sections. As shown in Table 2, $\beta_i = 1$ denotes a forward flip (e.g., $0 \rightarrow 1$ or $2 \rightarrow 3$), and $\beta_i = -1$ denotes a backward flip (e.g., $1 \rightarrow 0$ or $3 \rightarrow 2$). A transmission without error corresponds to $\beta_i = 0$.

Table 2. Representing the effect of a polarization flip on each transmitted photon. The Bits in and Bits out columns are the binary representation of the Quatrit in and Quatrit out representation, respectively. The value of β_i corresponds to a single bit flip error.

Bits in	Quatrit in (z_i)	Quatrit out (z_i^*)	Bits out	β_i
00	0	1	01	1
01	1	0	00	-1
10	2	3	11	1
11	3	2	10	-1

The transmitted symbol is computed by Alice as

$$z_i \equiv (x_i + y_i) \bmod 4,$$

where x_i is the binary response bit and y_i is the pre-shared quatrit stream. In the presence of a channel error, Bob receives

$$z_i^* \equiv (z_i + \beta_i) \bmod 4.$$

Bob then applies the inverse obfuscation step to recover the received output

$$x_i^* \equiv (z_i^* + 3y_i) \bmod 4.$$

Substituting for z_i^* and simplifying yields

$$\begin{aligned} x_i^* &\equiv (z_i + \beta_i + 3y_i) \bmod 4 \\ &\equiv (x_i + y_i + \beta_i + 3y_i) \bmod 4 \\ &\equiv (x_i + \beta_i) \bmod 4. \end{aligned}$$

Table 3 shows all combinations of the input bit x_i and the pre-shared nonce y_i , along with the corresponding outcomes of x_i^* . For each case, the error term β_i takes a value of either +1 or -1, resulting in exactly two possible erroneous quatrit outputs.

Table 3. Outcome of single bit flip errors on quatrit ($x_i^* \equiv x_i + \beta_i \text{mod} 4$). The table details the final output quatrits (x_i^*) and their corresponding error values (β_i) when each initial quatrit (z_i) is subject to a one-bit error, given a specific pre-shared nonce (y_i).

$\downarrow x_i$	$y_i = 0$	$y_i = 1$	$y_i = 2$	$y_i = 3$
0	$z_i = 0$	$z_i = 1$	$z_i = 2$	$z_i = 3$
	$\beta_i = 1$	$\beta_i = -1$	$\beta_i = 1$	$\beta_i = -1$
	$x_i^* = 1$	$x_i^* = 3$	$x_i^* = 1$	$x_i^* = 3$
1	$z_i = 1$	$z_i = 2$	$z_i = 3$	$z_i = 0$
	$\beta_i = -1$	$\beta_i = 1$	$\beta_i = -1$	$\beta_i = 1$
	$x_i^* = 0$	$x_i^* = 2$	$x_i^* = 0$	$x_i^* = 2$

Figure 4 presents a expansion of this mapping in Table 3 that models all possible transitions from a transmitted input x_i to a received output x_i^* under a quantum channel error rate α . The diagram explicitly accounts for basis selection, measurement outcomes, and polarization flips. The conditional probabilities $P(x_i^* | x_i)$ are obtained by summing the probabilities of all paths leading from x_i to x_i^* , resulting in:

$$\begin{aligned}
 P_{noeve}(x_i^* = 0 | x_i = 0) &= \frac{1 - \alpha}{2}, \\
 P_{noeve}(x_i^* = 1 | x_i = 0) &= \frac{\alpha}{4}, \\
 P_{noeve}(x_i^* = 2 | x_i = 0) &= 0, \\
 P_{noeve}(x_i^* = 3 | x_i = 0) &= \frac{\alpha}{4}, \\
 P_{noeve}(x_i^* = 0 | x_i = 1) &= \frac{\alpha}{4}, \\
 P_{noeve}(x_i^* = 1 | x_i = 1) &= \frac{1 - \alpha}{2}, \\
 P_{noeve}(x_i^* = 2 | x_i = 1) &= \frac{\alpha}{4}, \\
 P_{noeve}(x_i^* = 3 | x_i = 1) &= 0.
 \end{aligned}$$

Using this distribution, the expected number of occurrences of each received quatrit x_i^* in Bob's sifted sequence can be estimated. The conditional probability distribution $P(x_i^* | x_i)$ enables estimation of the expected occurrence rate of each received symbol in Bob's sifted sequence. Under normal operating conditions, the appearance of invalid states is determined by the quantum channel error rate α and follows a predictable distribution. This expected behavior serves as a reference model, allowing deviations beyond the predicted noise-induced error to be identified as potential evidence of eavesdropping.

The algorithm A1 in Section A Appendix shows how we implemented the quantum channel error. For simplicity and this initial work, we decided to simulate quantum channel error under the assumption that a quatrit will always flip to the incorrect value; and the quantum channel error rate determines the percentage of quatrits that flip to the incorrect value.

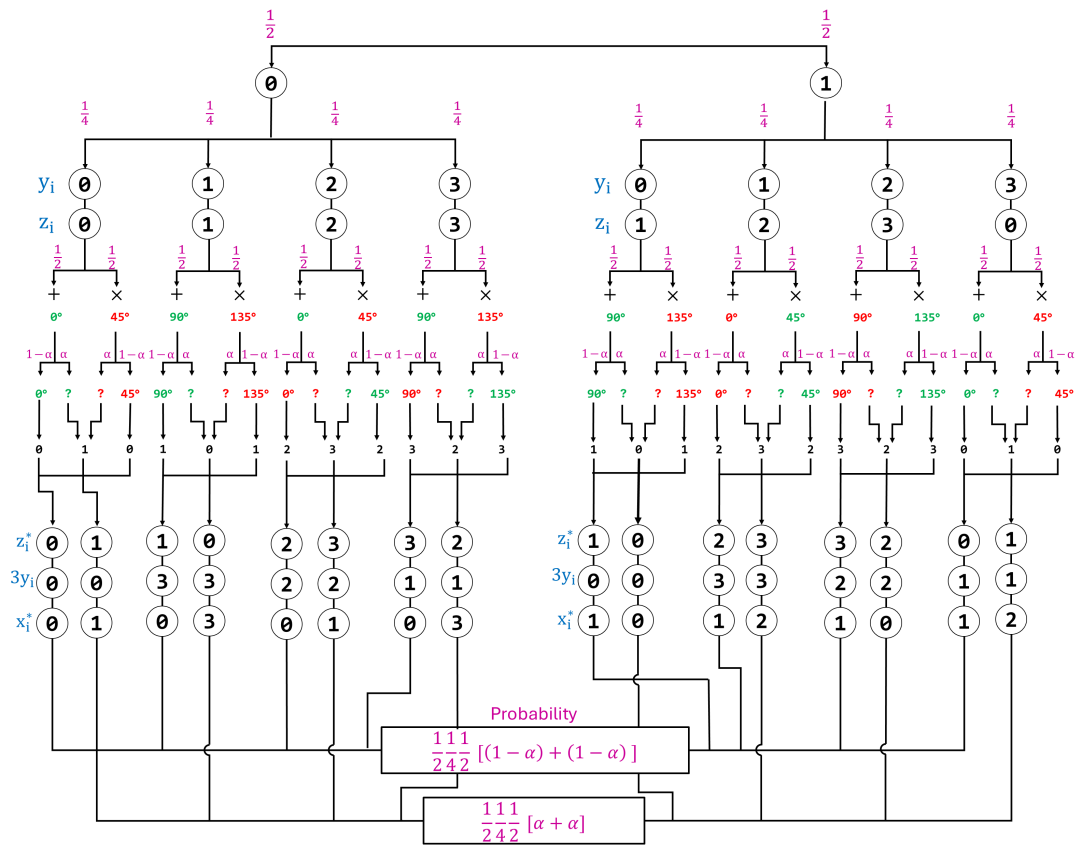


Figure 4. Non-eavesdropper scenario probabilistic decision tree illustrating all possible transitions from a transmitted input x_i to a received output x_i^* under the quantum channel error model, accounting for basis selection, polarization flips, and measurement outcomes.

3.3.2. Differentiating Eavesdropping-Induced Errors from Channel Noise

The goal of the eavesdropper detection mechanism is to reliably distinguish errors caused by inherent quantum channel noise from those introduced by an intercept-resend adversary. This distinction is achieved by analyzing the statistical distribution of the recovered quatrils after de-obfuscation and comparing the observed error behavior to the expected distribution under normal channel conditions. The deviation between these two distributions, denoted as ΔBER , serves as a decision metric that indicates whether the observed errors are consistent with channel noise alone or suggest the presence of an eavesdropper.

In this work, we adopt an error model in which quantum channel errors are treated as deterministic bit-flip events. Specifically, we assume that any channel error always results in an incorrect bit value at the receiver. Additionally, we model an intercept-resend eavesdropper who measures every transmitted qubit using a randomly chosen basis. When Eve's measurement basis matches that of the sender, no error is introduced; when the bases differ, Eve's measurement outcome is random, leading to a bit error with probability 50%. Under these assumptions, channel-induced errors and eavesdropper-induced errors contribute independently to the overall bit error rate.

The impact of an intercept-resend attack on the received symbol distribution is analyzed using the probabilistic decision tree shown in Figure 5. When Eve intercepts a quantum state, she performs a measurement in a randomly selected basis and resends the resulting state to Bob. If her basis choice

coincides with Alice's, the transmitted state remains unchanged; otherwise, the disturbance introduced by the measurement propagates through the channel and affects Bob's final observation.

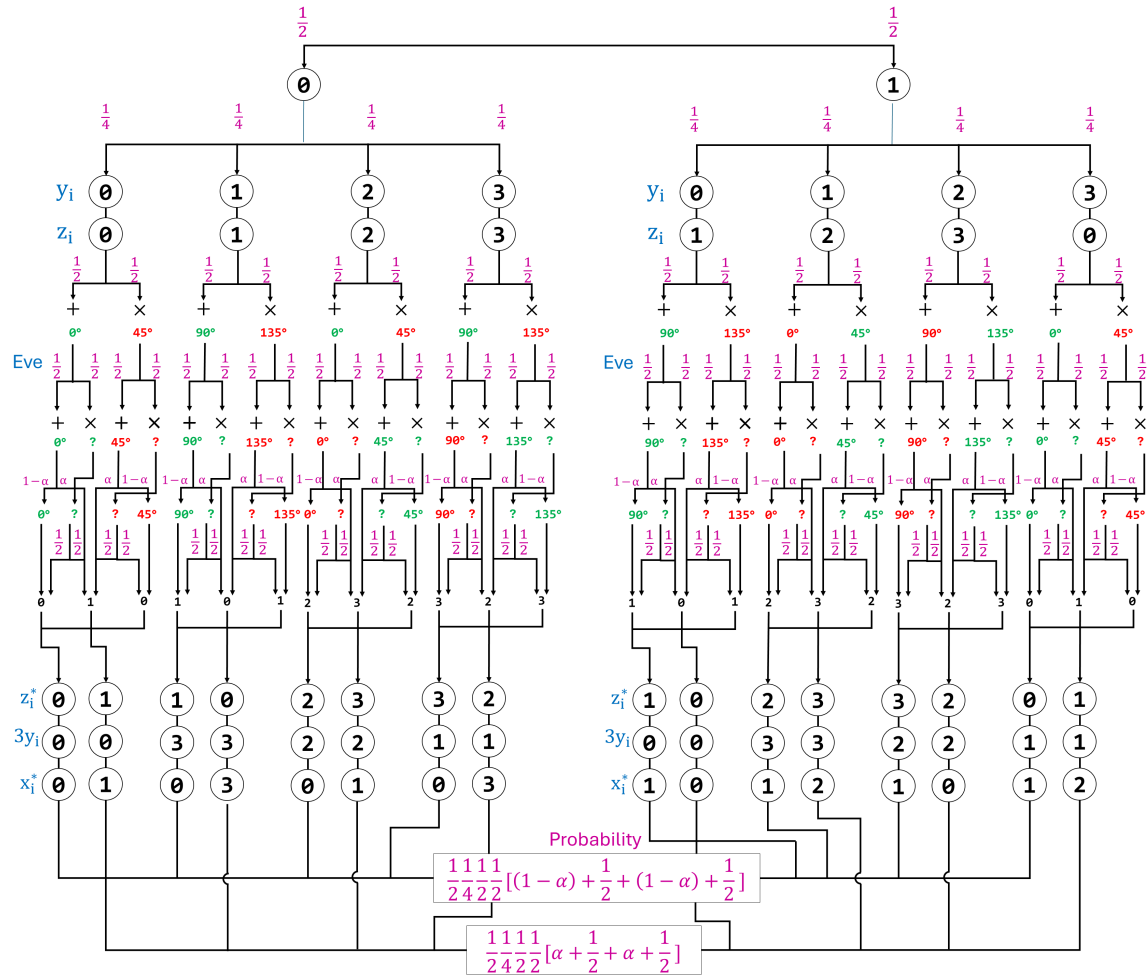


Figure 5. Eavesdropper scenario probabilistic decision tree showing all possible transitions from a transmitted input x_i to a received output x_i^* under the quantum channel error model, accounting for basis selection, polarization flips, and measurement outcomes.

To compute the probability of each possible received symbol in the presence of eavesdropping, we follow the same tree-based probability analysis introduced in the previous section. By summing the probabilities of all paths that lead from a transmitted input x_i to a recovered output x_i^* , we obtain the conditional probability distribution $P(x_i^* | x_i)$. The resulting probabilities are given by

$$\begin{aligned}
P_{yeseve}(x_i^* = 0 \mid x_i = 0) &= \frac{3 - 2\alpha}{8}, \\
P_{yeseve}(x_i^* = 1 \mid x_i = 0) &= \frac{1 + 2\alpha}{16}, \\
P_{yeseve}(x_i^* = 2 \mid x_i = 0) &= 0, \\
P_{yeseve}(x_i^* = 3 \mid x_i = 0) &= \frac{1 + 2\alpha}{16}, \\
P_{yeseve}(x_i^* = 0 \mid x_i = 1) &= \frac{1 + 2\alpha}{16}, \\
P_{yeseve}(x_i^* = 1 \mid x_i = 1) &= \frac{3 - 2\alpha}{8}, \\
P_{yeseve}(x_i^* = 2 \mid x_i = 1) &= \frac{1 + 2\alpha}{16}, \\
P_{yeseve}(x_i^* = 3 \mid x_i = 1) &= 0.
\end{aligned}$$

From this distribution, the probability of observing symbols in the invalid state set $\{2, 3\}$ at the receiver can be obtained by summing the corresponding terms. In the presence of eavesdropping, the probability of observing these invalid states is $(\alpha + 0.5)/4$. In the absence of an eavesdropper, their occurrence is due only to channel noise and occurs with probability $\alpha/2$.

The difference between these two quantities defines the detection metric ΔBER . This metric captures the excess error introduced by eavesdropping beyond what is expected from channel noise alone. If the observed rate of invalid states in Bob's recovered sequence exceeds the noise-only expectation by more than ΔBER , the deviation can be attributed to adversarial interference rather than normal channel imperfections. The expression for this difference is given by

$$\Delta\text{BER} = P(x_i^* \in \{2, 3\} | \text{Eve}) - P(x_i^* \in \{2, 3\} | \text{NoEve}) = \frac{\alpha + 0.5}{4} - \frac{\alpha}{2} = \frac{1}{8} - \frac{\alpha}{4} \quad (3)$$

This equation provides a quantitative measure for distinguishing errors introduced by an eavesdropper from those arising solely due to quantum channel noise. By comparing the observed rate of invalid states in Bob's recovered sequence with the noise-only expectation, the receiver can determine whether the observed deviation is consistent with normal channel behavior or indicative of external interference.

For a given channel error rate α , the expected occurrence of invalid states in the absence of eavesdropping is known analytically. Any excess beyond this baseline can be directly compared with the predicted value of ΔBER . When the measured deviation approaches the theoretical ΔBER corresponding to the estimated channel noise, the discrepancy cannot be explained by channel imperfections alone and instead indicates the presence of an intercept-resend attack.

This comparison allows Alice and Bob to infer eavesdropping activity without revealing any portion of their secret key. Rather than relying on absolute error values, the protocol exploits the structured difference between noise-induced and eavesdropping-induced error patterns, enabling reliable detection even in highly noisy channels.

3.4. Full Protocol Description

Combining the steps and information from the previous sections, the full protocol steps have been summarized here for clarity and completeness.

1. Step 1: Ephemeral Key Generation

At the beginning of each protocol execution, Alice generates a new random ephemeral key. This key is uniformly sampled from $\{0, 1\}^l$, where the bit-length l can be selected based on the desired security level. The ephemeral key is session-specific, meaning it is never reused across protocol runs and is discarded after successful completion. This ephemeral key plays two role in the protocol. First, it serves as the secret key that Alice intends to convey implicitly to Bob through

the protocol without transmitting it directly. Second, it acts as a selector that determines how Alice filters and encodes responses derived from the CRP mechanism.

2. **Step 2: CRP-Based Challenge and Full Response Generation**

After generating the ephemeral key, Alice applies the CRP mechanism explained in section 3. Alice first selects or constructs a challenge C and computes the full set of responses based on the protocol.

3. **Step 3: Ephemeral Key Encapsulation via Subset of Responses Selection**

In the third step, Alice encapsulates the ephemeral key K by using it to select a subset of responses from the full response set. Concretely, the bits of K define a deterministic selection rule that maps the ephemeral key to a subset index set.

4. **Step 4: Obfuscation via Modular Combination with Shared Quatrit Stream**

Alice combines x with a pre-shared quatrit masking stream y , which is known only to Alice and Bob. This operation lifts the binary stream x into a quatrit stream z , effectively hiding the original value of x . As a result, even if an adversary gains partial information about the transmitted quantum states, they cannot directly infer the underlying binary data without knowledge of the shared quatrit stream. See section 3.2

5. **Step 5: Encoding into Polarization States**

In the next step, Alice encodes each quatrit symbol z into a corresponding quantum polarization state according to table (referred to the table).

6. **Step 6: Transmission over a Two-Wavelength Quantum Channel**

Finally, Alice transmits the encoded polarization states over the quantum channel to Bob. The transmission is performed using two distinct optical wavelengths, enabling parallel or interleaved encoding of the polarization states across wavelength channels.

7. **Step 7: Measurement of Incoming Quantum States**

Upon receiving the transmitted quantum states, Bob performs measurements on the incoming qubits using randomly chosen measurement bases, independently selected for each received state.

8. **Step 8: Basis Sifting**

After measurement, Alice and Bob engage in a classical sifting procedure over an authenticated public channel. During this step, Alice reveals the basis information associated with each transmitted quantum state, while Bob discloses the bases he used for measurement. No symbol values are revealed during this exchange.

Only the positions for which Alice's encoding basis matches Bob's measurement basis are retained. All other positions are discarded.

9. **Step 9: Regeneration of the Full CRP Response Set**

Using the challenge C previously communicated by Alice, Bob locally regenerates the full set of CRP responses.

10. **Step 10: Recovery of the Binary Symbol Sequence**

For each retained symbol in the sifted sequence, Bob applies the inverse modular operation to recover the original binary symbol x . Using the shared quadrature masking stream y and the measured quatrit stream z , Bob computes $x \equiv (3y + z) \bmod 4$. Since $3 \equiv -1 \pmod{4}$, this operation effectively removes the masking applied by Alice in Step 4, yielding the original response bits. When no eavesdropping or excessive noise is present, the recovered values of x match Alice's original symbols with high probability.

11. **Step 11: Security Evaluation**

Before proceeding to key reconstruction, Bob evaluates the security of the transmission by computing the BER over a designated subset of the recovered symbols. The observed BER is compared against an expected baseline, and the deviation is used as a decision metric to assess whether the channel behavior is consistent with normal quantum noise or indicative of adversarial interference. If $\Delta(\text{BER})$ exceeds a predefined threshold, the session is deemed

compromised and the protocol is aborted. Otherwise, Bob concludes that the transmission is secure and proceeds to the final key recovery stage, during which the encapsulated ephemeral key is reconstructed from the recovered response subset. See section 3.3

12. Step 12: Key Recovery

Upon successful completion of the security check, Bob reconstructs the encapsulated ephemeral key.

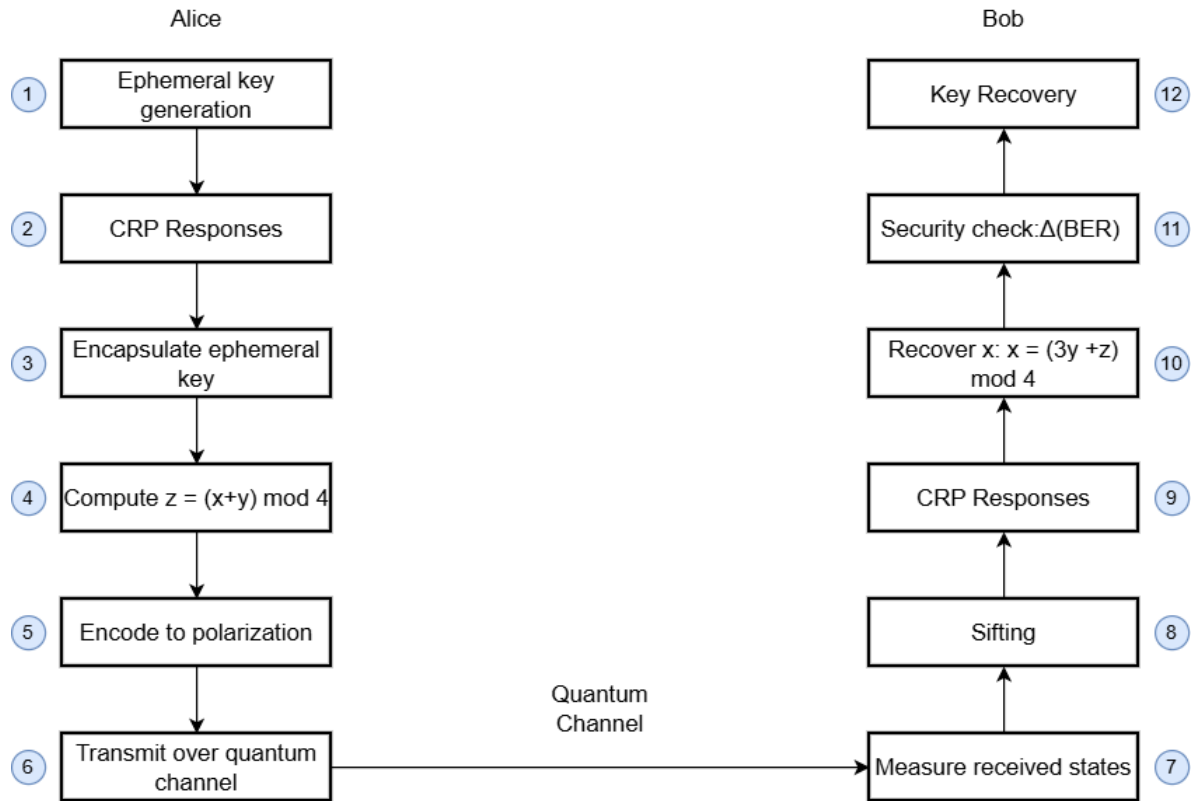


Figure 6. Overview of the proposed protocol workflow from key generation to recovery.

4. Security Proof

4.1. Quantum Transmission Security

To show the security of the quantum transmission section of the protocol we can treat sending photons encoded with two wavelengths as two separate instances of the BB84 protocol, which each instance has been shown to be secure [45,77]. From here we'll note that for single channel BB84 protocol Shor and Preskill showed that you can guarantee security so long as the measured phase and bit errors rates of the qubits are less than about 11% [45]. Our goal is to derive the error bound that we can guarantee security. We'll use the same mechanism by seeing when the quantity $1 - 2H(x)$ goes to zero with

$$H(X) = - \sum_{x \in \chi} P(x) \log_b(P(x)), \quad (4)$$

being the Shannon entropy. To do this let's look at how we can encode a quatrit $z_i \in \mathbb{Z}_4$. We can encode z_i by using a pair of bits that can be generated randomly. We can use the assumption that the two bits are generated independently of each other, thus given that a 0 could be generated with probability $P_{\text{bit}}(0) = p$ and generating a 1 with probability $P_{\text{bit}}(1) = 1 - p$, we get the following:

Table 4. Encoding table for every possible quatrit $z_i \in \mathbb{Z}$

Quatrit z_i	Binary Encoding	Probability $P(z_i)$
0	00	p^2
1	01	$p(1-p)$
2	10	$p(1-p)$
3	11	$(1-p)^2$

From here we get our Shannon entropy to be

$$H(p) = -p^2 \log_4(p^2) - 2p^2 \log_4(p(1-p)) - (1-p)^2 \log_4((1-p)^2). \quad (5)$$

A side note is we can make our logarithms base 4 as the information being sent over is in base 4. Using equation (5) we get that the quantity $1 - 2H(x)$ go to zero at the exact same point 11%, which is equivalent to [45].

5. Methodology

The goal of the simulation was to implement the proposed protocol in Python and evaluate its ability to recover a cryptographic key across varying channel noise conditions. We represented our protocol in a Python program and ran 10,000 Monte Carlo simulations of Alice and Bob distributing 256-bit long keys using six different response lengths: 16, 32, 64, 128, 256, and 512. For each response length, the protocol was tested across increasing channel error to study how noise affects key recovery performance. Simulations were run for both the case with an eavesdropper and the case without one. In the simulation, the eavesdropper (Eve) is modeled as a random variable that interacts with each photon independently using an intercept and resend strategy. The pre-shared stream is implemented using a pseudo-random number generator.

Channel errors are modeled as independent and identically distributed random events. In each trial, a transmitted photon may experience a channel error with a given probability. This will result in an incorrect photon state. In this simulation, a channel error event guarantees an incorrect measurement outcome when measured in the same basis as Alice. For example, if Alice transmits a photon polarized at 90 degrees, a channel error event will replace it with a photon polarized at 0 degrees and ensures an incorrect measurement outcome when the same measurement basis is used. Each photon can experience one channel error event at most. In practice, channel noise may be more complex depending on the setup.

The key recovery step is computationally intensive because it requires evaluating possible candidate keys. To address this, the simulation code was parallelized and executed on Northern Arizona University's high-performance computer (HPC) in order to distribute the computations across nodes and speed up the run time. Computer nodes were equipped AMD EPYC 7542 32-Core processors @2.9 GHz. Each simulation job utilized up to 64 CPU cores with single-threaded execution per core. During the key recovery step, the number of possible keys to try can become large. If the number of possible keys reaches or exceeds $1e27$, the simulation run is terminated, and the run is classified as a failed key recovery. This approximate threshold is used to ensure a practical computation time. Simulation results were analyzed using statistical tools in R.

6. Results

We implemented the two-wavelength key encapsulation QKD protocol described in Section 3 using software simulation as described in Section 5. We studied the impact of noise on error rates in the cryptographic key and considered performance during an eavesdropper attack. This result section aims to show the empirical findings of our investigation.

6.1. Key Recovery in the Presence of Noise

Our first experiment aims to quantify the ability to recover the ephemeral key in the presence of channel noise. The relationship between the successful key recovery rate and channel error rate is shown in Figure 7. The graph was produced by running the simulation 10,000 times for each response length (32, 64, 128, 256, and 512 bits) across channel error rates ranging from 0 to 0.5 in increments of 0.01. The count of successful key recoveries out of 10,000 iterations was recorded for each response length and channel error rate pairing.

Figure 7 shows that as channel error increases, shorter response lengths lose the ability to recover the key earlier, while longer response lengths continue to succeed at higher channel noise levels, demonstrating greater tolerance to channel error. In other words, longer response lengths are able to withstand more channel error. While increased response length improves tolerance to channel error, key recovery sharply declines once the error rate exceeds a critical channel error threshold. For example, in Figure 7, the key recovery rate for the 128-bit response length falls sharply at a channel error rate of approximately 0.3 (30%). The protocol exhibits the best performance with a response length set at $p = 512$, demonstrating that the system can withstand noise levels as high as 40% at this response length. Unfortunately, increasing the response lengths p comes at the cost of increasing latency (see Section 7 for more details). Overall, this experiment assessed key recovery performance at various response lengths across channel error levels and demonstrates that the longest response length can recover the key even with noise as high as 40%. Eavesdropping was not included in this stage of evaluation, and the following section addresses eavesdropper detection.

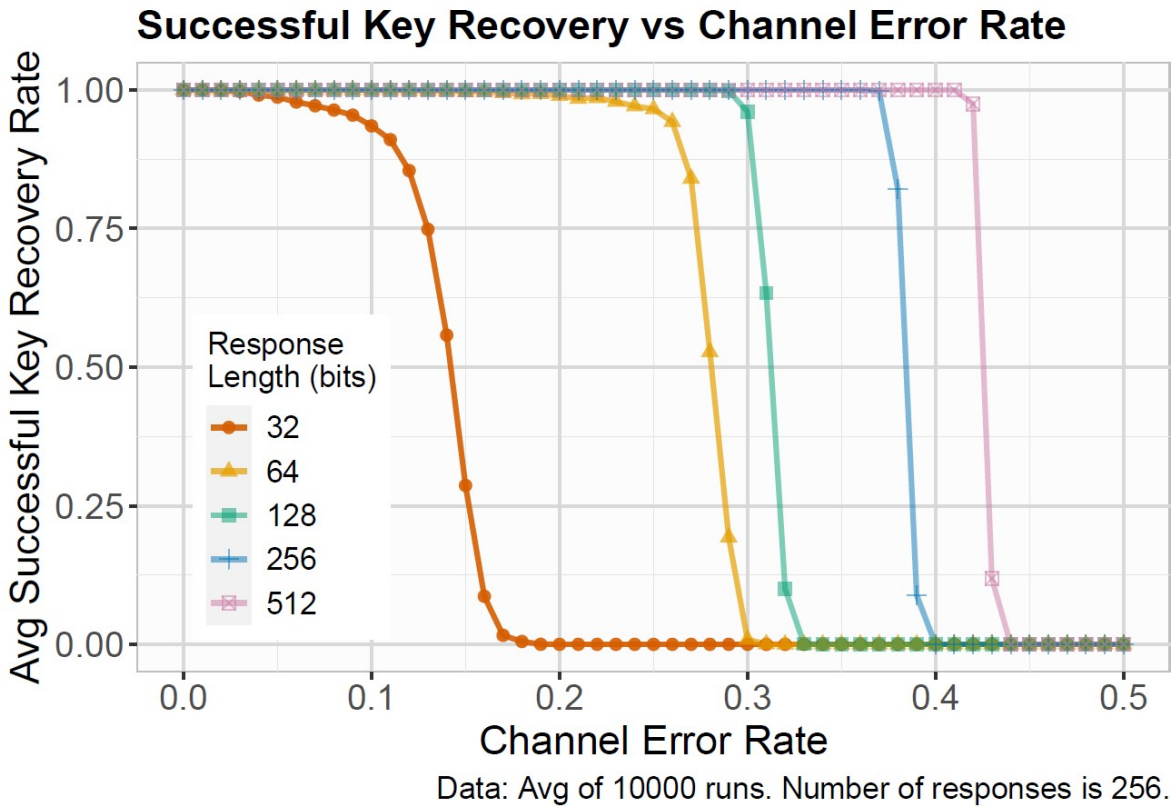


Figure 7. Successful key recovery rate averaged over 10,000 trial runs for response lengths 32, 64, 128, 256, and 512 bits over incrementing simulated quantum channel error from 0 to 50%. The test was run with no eavesdropper.

6.2. Differentiating Eavesdropper From Channel Error

To assess the protocol’s effectiveness in accurately detecting the presence of an eavesdropper in the presence of noise, we ran 10,000 simulations under varying noise conditions (BER), both with and without the eavesdropper. Figure 8 illustrated in this section uses Kernel Density Estimation (KDE) to visualize the probability density function of the difference in error rates (see Equation 3) in a

transmitted stream, comparing scenarios with and without the presence of Eve. Data was collected under three noise conditions, 10%, 30%, and 40%. The peaks of the KDE plots correspond to the most frequent values of the difference in error rate. The distribution of error rate differences without Eve presents a more narrow shape at lower noise levels, suggesting that most of the errors have minimal variation. The presence of Eve introduces a wider distribution, indicating more varied error differences. As the noise level increases, the distributions for scenarios with and without Eve begin to converge, making it more difficult to distinguish Eve’s presence from noise. This convergence is particularly evident at higher noise levels, where the error rate differences between the Eve and No Eve scenarios become less evident.

Kernal Density of Delta: Impact of Channel Error Rate on Eavesdropper Detection

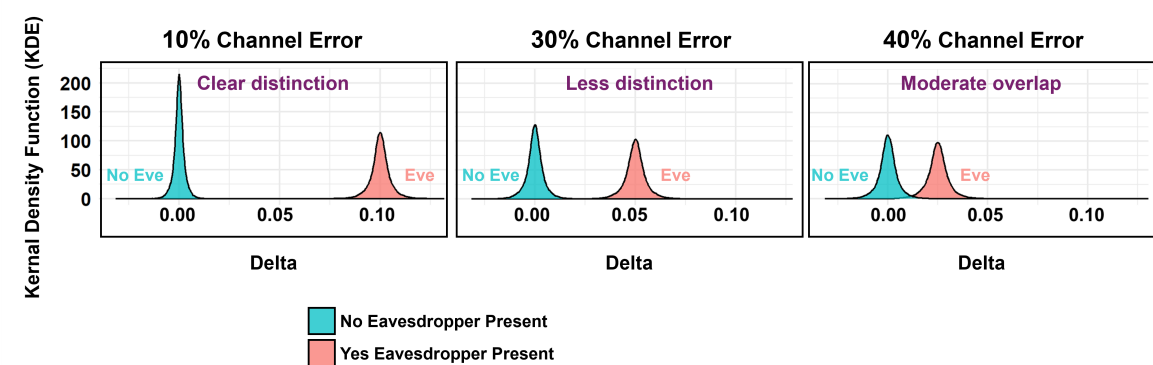


Figure 8. This plot illustrates the effect of increasing channel error rates on the separability of error distributions with and without an eavesdropper present. At low error rates, the distributions remain well separated, allowing a clear distinction between normal channel noise and eavesdropping-induced disturbances. As the channel error rate increases, the two distributions remain largely distinguishable; however, partial overlap begins to emerge, which can lead to occasional misclassification of the eavesdropper’s presence under higher-noise conditions.

Figure 9 shows the impact of response length on the error rate distribution at a 40% noise level for response lengths 64 and 512. The plot reveals distinctive variations in density distribution height and width for response lengths 64 and 512, suggesting that longer response lengths may result in less error rate variability, as evidenced by the narrower distribution for the No_Eve_P512 condition. The long tail on the red plot labeled as Eve_P64 which overlaps with the blue plot labeled as No_Eve_P64 suggests some instances where eavesdropping was in place but might not have been detected due to the minimal amount of error introduced by the eavesdropping activities. Consequently, this could lead to false negatives.

In assessing the efficacy of the proposed eavesdropping method, we quantified key performance indicators: False Negative Rate (FNR), False Positive Rate (FPR), and accuracy. These metrics serve as benchmarks to evaluate the reliability and robustness of our method under different noise levels.

Table 5 summarizes a framework for categorizing detection instances into true-positives (TP), false-negatives (FN), false-positives (FP), and true-negatives (TN) [76]. We define them according to the following formulas:

Table 5. Terminologies for Eavesdropper Detection

P	Judgment	Terminology
Exist	Exist	True-positive (TP)
	Not exist	False-negative (FN)
Not exist	Exist	False-positive (FP)
	Not exist	True-negative (TN)

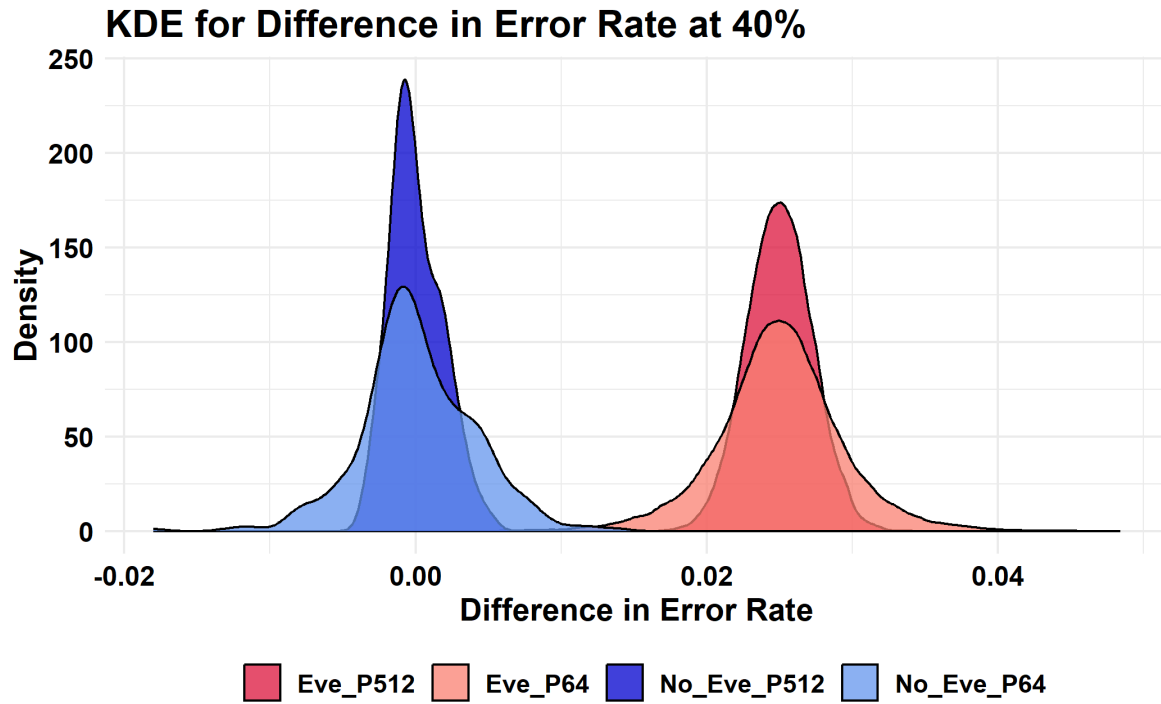


Figure 9. Kernel Density Estimation (KDE) of Protocol Performance for 64-bit vs. 512-bit Response Lengths at 40% Noise Level.

$$FNR = \frac{FN}{FN + TP} \quad (6)$$

$$FPR = \frac{FP}{FP + TN} \quad (7)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

The FNR and FPR of our eavesdropping detection system as functions of BER are displayed in Tables 6 and 7 for different P (Response length), ranging from 32 to 512, for different channel error rates. Table 7 shows that the protocol exhibits zero FPR when the channel error is below 30% across all response lengths. Notably, for longer response lengths exceeding 128 bits, the protocol maintains robust performance, effectively discerning the absence of eavesdropping even with channel error rates up to 40%. This demonstrates the system's competent eavesdropping detection capabilities under noisy conditions.

Table 6 showcases the FNR which represents the protocol's efficacy when an eavesdropper is present. Across all response lengths, the protocol consistently identifies Eve's activities up to a 35% channel error rates. Longer response lengths exhibit a tolerance surpassing 40%, indicating a higher resilience in detection capabilities against noise in the channel.

The reason why the FPR and FNR values appear identical stems from the symmetrical distribution of errors observed in scenarios involving eavesdropping versus those without. However, in real-world settings, this symmetry might not hold, potentially leading to variations in the FPR and FNR values.

Table 6. False Negative Rate (FNR) Across Channel Error Rate for Different Response Lengths (p)

Channel Error Rate	p=32	p=64	p=128	p=256	p=512
0	0	0	0	0	0
0.1	0	0	0	0	0
0.2	0	0	0	0	0
0.3	0.0013	0	0	0	0
0.4	0.0809	0.0232	0.0035	0	0
0.49	0.4539	0.4239	0.3939	0.3484	0.2904

Table 7. False Positive Rate (FPR) Across Channel Error Rate for Different Response Lengths (p)

Channel Error Rate	p=32	p=64	p=128	p=256	p=512
0	0	0	0	0	0
0.1	0	0	0	0	0
0.2	0	0	0	0	0
0.3	0.0002	0	0	0	0
0.4	0.0590	0.0135	0.0007	0	0
0.49	0.4346	0.4183	0.3807	0.3322	0.2916

To evaluate the precision of our protocol across various response lengths, we calculated accuracy according to Equation 8. Figure 10 depicts that the protocol maintains an accuracy rate of 100% when the BER is below 35%. High accuracy at relatively high noise levels supports that the protocol is robust and is a promising indicator of its reliability in practical, noisy scenarios.

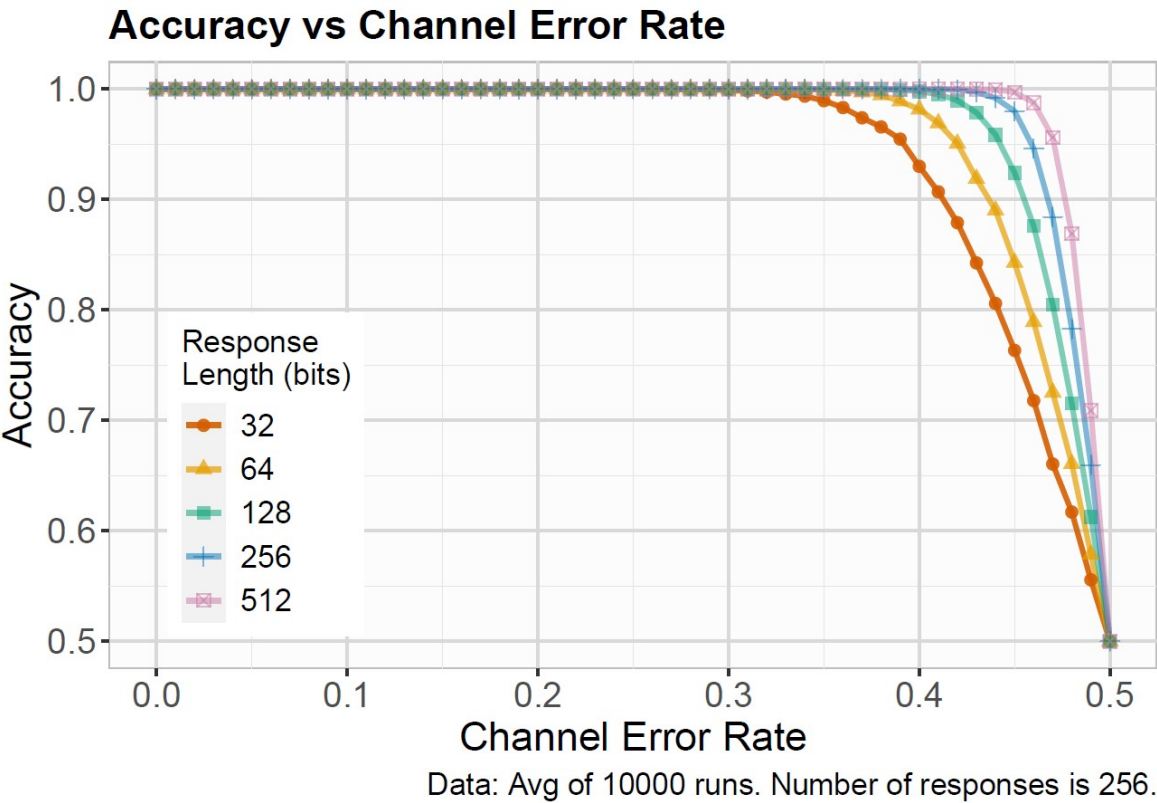


Figure 10. Performance Accuracy Rate Across Various Noise Levels and Response Lengths.

7. Discussion

7.1. Computational Overhead, Complexity, and Limitations

Although the CRP-based error correction protocol removes the need for multi-round classical reconciliation, it shifts part of the cost to local computation. This overhead mainly stems from the generation and handling of cryptographic tables and from the receiver-side search and recovery procedure. Increasing the response length enhances tolerance to channel noise, up to approximately 40%, but naturally leads to higher processing time and latency. In practice, this behavior reflects a tunable design trade-off rather than a fundamental limitation. Selecting moderate response lengths (e.g., 128–256 bits) allows the protocol to maintain strong robustness while remaining computationally practical for real-world QKD deployments with limited resources.

Table 8 quantifies how the choice of response length impacts the number of qubits required to transmit an encapsulated ephemeral key of different sizes. For a fixed key size, increasing the response length results in a proportional increase in the total transmitted qubits. This relationship highlights the practical scaling behavior of the protocol: shorter response lengths minimize transmission overhead, while longer responses provide improved robustness against channel noise at the cost of increased quantum resources.

Table 8. Number Of Qubits Needed for Transmission of Encapsulated Ephemeral Keys with Various Length

Response Length	Key Size: 128	Key Size: 256	Key Size: 512
32	≈ 256 B	≈ 512 B	≈ 1 KB
64	≈ 512 B	≈ 1 KB	≈ 2 KB
128	≈ 1 kB	≈ 2 KB	≈ 4 KB
256	≈ 2 kB	≈ 4 KB	≈ 8 KB
512	≈ 4 kB	≈ 8 KB	≈ 16 KB

7.2. Comparison with Other Work

Compared to classical reconciliation techniques such as Cascade, LDPC, and Winnow, our CRP-based protocol achieves a markedly different balance between communication and computation. Cascade requires multiple interactive rounds and leaks information during reconciliation, whereas our protocol eliminates communication rounds entirely and avoids information leakage. LDPC performs well under moderate noise levels (25% QBER) but its efficiency degrades under burst error distributions; our method, by contrast, depends only on total errors per block, enabling robustness against non-uniform error patterns. Winnow and related codes tolerate only low error rates (<10%), making them less suitable in noisy channels. In contrast, our simulations demonstrate reliable key recovery at quantum bit error rates as high as 40%, without helper data exchange.

Table 9. Comparison of Error Correction Mechanisms in QKD

Metric	Our Protocol	LDPC	Cascade	Winnow
Communication Rounds	0	1	3–6	1–3
Bits Revealed	0	Moderate	High	Moderate
Latency	Low	Low	High	Medium
Decoding Method	Crypto-table reference	Belief Propagation	Binary Search	Hamming Code Based
Leakage via Reconciliation	None	Yes	Yes	Yes
Error Tolerance (QBER)	High (<40%)	High (<25%)	Medium (<15%)	Medium (<10%)
Adaptivity to Channel Noise	High	Medium	Low	Medium

7.3. Importance of Independent Eavesdropping Detection

An important implication of this work is the protocol's ability to support independent eavesdropping detection without relying on additional classical exchanges or partial key disclosure. Unlike many existing QKD schemes that depend on public reconciliation steps, our approach enables Alice and Bob to locally construct identical cryptographic tables using a shared CRP mechanism. As a result, an adversary attempting to compromise the system must not only interfere with the quantum channel but also reconstruct this independently generated structure, substantially increasing the attack complexity.

Another important aspect of this work is that Bob can assess the integrity of the quantum transmission solely from his measurement outcomes and local computations, without requiring explicit confirmation or auxiliary data from Alice. This self-contained verification preserves key confidentiality while minimizing information exposure during post-processing. By embedding consistency checks, appearing as measurable statistical deviations, directly into the encoding and recovery stages, the protocol effectively decouples eavesdropping detection from classical reconciliation.

8. Future Work

Our work has demonstrated the ability to recover keys in the presence of higher noise using preliminary error model simulations. Future work includes evaluating the protocol using more complex quantum simulation and real implementation, optimizing protocol parameters, and quantifying the distance improvements for long-distance communication.

The next steps involve evaluating the protocol using more complex quantum simulation tools that better resemble the behavior of real equipment such as photon loss, detector imperfections, and decoherence. These studies will provide a more realistic analysis of the protocol's performance and help improve performance in practical settings. Beyond the simulations, future work includes implementing the protocol in a real system to assess its performance and consider environmental variability. Additionally, further optimization of protocol parameters such as the response length can help improve efficiency and scalability. This will provide a more realistic evaluation of the protocol.

Finally, future work will also focus on evaluating protocol performance across varying transmission distances using more realistic simulation models and assessing scalability and efficiency. The current simulation can be extended to model the relationship between transmission distance, photon loss, and key recovery performance. Establishing this connection will allow protocol performance to be evaluated as a function of distance, but requires more complex simulation and modeling distance-dependent attenuation and other error rates related to distance. Because photon loss increases exponentially with distance and is a known limiting factor in QKD systems, the protocol's demonstrated ability to recover keys in scenarios with up to 40% errors, suggests potential for improved distance tolerance beyond the tens of kilometers range typically reported for conventional QKD protocols without advanced techniques. Further improvements might be achievable by integrating our protocol into existing long-distance QKD frameworks which report distances around hundreds of kilometers. Future research will focus on a comprehensive evaluation of the compatibility and performance of this integration, in addition to incorporating distance-dependent models to quantify the improvement and determine practical operating ranges.

9. Conclusion

The methodology and analysis presented in this paper highlight the efficacy and benefits of the CRP-based ephemeral key encapsulation technique employed in QKD, specifically for effective error detection, key recovery, and eavesdropper identification. We demonstrate successful error-free key recovery with noise in the transmitted information blocks as high as 40%. Our results verify that, by adjusting the response length and ephemeral key sizes, a secure key exchange can be facilitated between two parties.

10. Patents

1. Cambou BF, Hoffstein J, Ghanaimiandoab D, Riggs B, Burke I, Heynssens JB, inventors; Northern Arizona University, Brown University, assignee. Error management for long distance quantum key distribution and symmetrical key distribution. United States patent application US 18/907,586. 2025 Apr 10. [78]
2. Cambou BF, Garrett ML, Partridge M, Ghanaimiandoab D, inventors; Northern Arizona University, assignee. Protocols with noisy response-based cryptographic subkeys. United States patent application US 18/885,226. 22 May 2025. [79]

Author Contributions: Conceptualization, B.C.; methodology, D.G. and B.R; software, D.G. and B.R; validation, D.G., B.R. and B.C.; formal analysis, D.G., B.R. and N.N.; investigation, D.G., B.R.; resources, B.C.; data curation, D.G. and B.R.; writing—original draft preparation, D.G., B.R.; writing—review and editing, D.G., B.R., N.N. and B.C; visualization, D.G., B.R.; supervision, B.C.; project administration, B.C.; funding acquisition, B.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Castle Shield Holdings, LLC.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: This research was carried out in collaboration with Northern Arizona University and Castle Shield Holdings, LLC. The authors gratefully acknowledge the support of Jeffrey Roney for enabling this research initiatives. The authors also thank Julie Heynssens and Ian Burke for their assistance in the project development.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

QKD	Quantum Key Distribution
CRP	Challenge-Response Pair
BB84	Bennett–Brassard-84
BER	Bit Error Rate
QBER	Quantum Bit Error Rate
RSA	Rivest-Shamir-Adleman
ECC	Elliptic Curve Cryptography
WDM	Wavelength-Division Multiplexing
LDPC	Low-Density Parity-Check
PNS	Photon Number Splitting
QC	Quantum Channel
XOR	Exclusive OR
HPC	High Performance Computer
KDE	Kernel Density Estimation
FPR	False Positive Rate
FNR	False Negative Rate
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

Appendix A

Algorithm A1 Simulation of Quantum Channel Error

1: **Input:** QKD Input $Q_i = \{q_0, q_1, \dots, q_n\}$, Quantum Channel Error Rate α

2: Compute how many numbers to flip; $N = n \times \alpha$

3: Randomly choose N indices from $[0, n - 1]$; $I_i : \{i_0, i_1, \dots, i_N\}$

4: Change the polarization of $q_{i_i} \pm 90^\circ = q'_{i_i}$

5: Replace new q_{i_i} with q'_{i_i}

6: **Output:** Q'_i

References

1.

Zhao, X.; Xu, X.; Qi, L.; Xia, X.; Bilal, M.; Gong, W.; Kou, H. Unraveling quantum computing system architectures: An extensive survey of cutting-edge paradigms. *Information and Software Technology* **2024**, *167*, 107380.

2.

Liu, Y.; Zhang, W.J.; Jiang, C.; Chen, J.P.; Zhang, C.; Pan, W.X.; Ma, D.; Dong, H.; Xiong, J.M.; Zhang, C.J.; et al. Experimental twin-field quantum key distribution over 1000 km fiber distance. *Physical Review Letters* **2023**, *130*, 210801.

3.

Liu, M.; Li, Z.; Cai, K.; Allcock, J.; Zhang, S.; Lui, J. Quantum BGP with Online Path Selection via Network Benchmarking. In Proceedings of the IEEE INFOCOM, 2024.

4.

Wang, M.; Yan, F.; Gao, T. Generation of four-photon polarization entangled decoherence-free states with cross-Kerr nonlinearity. *Scientific Reports* **2016**, *6*, 38233.

5.

Zanardi, P.; Rasetti, M. Noiseless quantum codes. *Physical Review Letters* **1997**, *79*, 3306.

6.

Priyanka, M.; Sinha, U. Study of BB84 QKD protocol: Modifications and attacks. *Retrieved August 2020*, *8*.

7.

Bennett, C.H.; Bessette, F.; Brassard, G.; Salvail, L.; Smolin, J. Experimental quantum cryptography. *Journal of cryptology* **1992**, *5*, 3–28.

8.

Diamanti, E.; Lo, H.K.; Qi, B.; Yuan, Z. Practical challenges in quantum key distribution. *npj Quantum Information* **2016**, *2*, 1–12.

9.

Fickler, R.; Prabhakar, S. Quantum communication with structured photons. In *Structured Light for Optical Communication*; Elsevier, 2021; pp. 205–236.

10.

McMahon, D. *Quantum computing explained*; John Wiley & Sons, 2007.

11.

Ecker, S.; Pseiner, J.; Piris, J.; Bohmann, M. Advances in entanglement-based QKD for space applications. *arXiv preprint arXiv:2210.02229* **2022**.

12.

Cambou, B.F.; Montano, I.; Behunin, R.; Rodriguez, V. Secure multi-state quantum key distribution with wavelength division multiplexing, 2021. US Patent App. 16/951,760.

13.

Hong, K.W.; Foong, O.M.; Low, T.J. Challenges in quantum key distribution: a review. In Proceedings of the Proceedings of the 4th International Conference on Information and Network Security, 2016, pp. 29–33.

14.

Bennett, C.H.; Brassard, G. Quantum cryptography: Public key distribution and coin tossing. *Theoretical computer science* **2014**, *560*, 7–11.

15.

Alshehri, O.; Li, Z.H.; Al-Amri, M. Chapter 1 - Basics of quantum communication. In *Structured Light for Optical Communication*; Al-Amri, M.D.; Andrews, D.L.; Babiker, M., Eds.; Nanophotonics, Elsevier, 2021; pp. 1–36. <https://doi.org/https://doi.org/10.1016/B978-0-12-821510-4.00007-8>.

16.

Kong, P.Y. A review of quantum key distribution protocols in the perspective of smart grid communication security. *IEEE Systems Journal* **2020**.

17.

Hughes, R.J.; Morgan, G.L.; Peterson, C.G. Quantum key distribution over a 48 km optical fibre network. *Journal of Modern Optics* **2000**, *47*, 533–547.

18.

Boaron, A.; Boso, G.; Rusca, D.; Vulliez, C.; Autebert, C.; Caloz, M.; Perrenoud, M.; Gras, G.; Bussi eres, F.; Li, M.J.; et al. Secure quantum key distribution over 421 km of optical fiber. *Physical review letters* **2018**, *121*, 190502.

19.

Hiskett, P.A.; Rosenberg, D.; Peterson, C.G.; Hughes, R.J.; Nam, S.; Lita, A.; Miller, A.; Nordholt, J. Long-distance quantum key distribution in optical fibre. *New Journal of Physics* **2006**, *8*, 193–193.

20.

Hajomer, A.A.; Derkach, I.; Jain, N.; Chin, H.M.; Andersen, U.L.; Gehring, T. Long-distance continuous-variable quantum key distribution over 100-km fiber with local local oscillator. *Science Advances* **2024**, *10*, eadi9474.

21. Pittaluga, M.; Lo, Y.S.; Brzosko, A.; Woodward, R.I.; Scalcon, D.; Winnel, M.S.; Roger, T.; Dynes, J.F.; Owen, K.A.; Juárez, S.; et al. Long-distance coherent quantum communications in deployed telecom networks. *Nature* **2025**, *640*, 911–917.
22. Buttler, W.T.; Lamoaux, S.K.; Torgerson, J.R.; Nickel, G.; Donahue, C.; Peterson, C.G. Fast, efficient error reconciliation for quantum cryptography. *Physical Review A* **2003**, *67*, 052303.
23. Yan, H.; Peng, X.; Lin, X.; Jiang, W.; Liu, T.; Guo, H. Efficiency of winnow protocol in secret key reconciliation. In Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering. IEEE, 2009, Vol. 3, pp. 238–242.
24. Zhu, M.; Cui, K.; Li, S.; Kong, L.; Tang, S.; Sun, J. A Code Rate-Compatible High-Throughput Hardware Implementation Scheme for QKD Information Reconciliation. *Journal of Lightwave Technology* **2022**, *40*, 3786–3793. <https://doi.org/10.1109/JLT.2022.3149567>.
25. Johnson, J.S. An analysis of error reconciliation protocols for use in quantum key distribution **2012**.
26. Zapatero, V.; Navarrete, Á.; Curty, M. Implementation security in quantum key distribution. *Advanced Quantum Technologies* **2025**, *8*, 2300380.
27. Ashkenazy, A.; Idan, Y.; Korn, D.; Fixler, D.; Dayan, B.; Cohen, E. Photon number splitting attack–proposal and analysis of an experimental scheme. *Advanced Quantum Technologies* **2024**, *7*, 2300437.
28. Ekert, A.K. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.* **1991**, *67*, 661–663. <https://doi.org/10.1103/PhysRevLett.67.661>.
29. Yuan, Z.S.; Bao, X.H.; Lu, C.Y.; Zhang, J.; Peng, C.Z.; Pan, J.W. Entangled photons and quantum communication. *Physics Reports* **2010**, *497*, 1–40.
30. Nandal, R.; Nandal, A.; Joshi, K.; Rathee, A.K. Comparison of Some of the Most Prominent QKD Protocols: A Review. *IJIP Journal of Telecommunications* **2021**, *13*.
31. Ilic, N. The ekert protocol. *Journal of Phy334* **2007**, *1*, 22.
32. Kumar, A.; Garhwal, S. State-of-the-art survey of quantum cryptography. *Archives of Computational Methods in Engineering* **2021**, *28*, 3831–3868.
33. Bennett, C.H. Quantum cryptography using any two nonorthogonal states. *Physical review letters* **1992**, *68*, 3121.
34. Shukla, M.; Patel, S. Prominent Security of the Quantum Key Distribution Protocol. *International Journal of Science and Research* **2018**, *8*.
35. Pirandola, S.; Andersen, U.L.; Banchi, L.; Berta, M.; Bunandar, D.; Colbeck, R.; Englund, D.; Gehring, T.; Lupo, C.; Ottaviani, C.; et al. Advances in quantum cryptography. *Advances in optics and photonics* **2020**, *12*, 1012–1236.
36. Abushgra, A.A. Variations of QKD Protocols Based on Conventional System Measurements: A Literature Review. *Cryptography* **2022**, *6*. <https://doi.org/10.3390/cryptography6010012>.
37. Krithika, S. Quantum key distribution (QKD): a review on technology, recent developments and future prospects. *Research Journal of Engineering and Technology* **2017**, *8*, 291–294.
38. Bechmann-Pasquinucci, H.; Gisin, N. Incoherent and coherent eavesdropping in the six-state protocol of quantum cryptography. *Physical Review A* **1999**, *59*, 4238.
39. Lo, H.K.; Chau, H.F.; Ardehali, M. Efficient quantum key distribution scheme and a proof of its unconditional security. *Journal of Cryptology* **2005**, *18*, 133–165.
40. Sharifi, M.; Azizi, H. A simulative comparison of bb84 protocol with its improved version. *Journal of Computer Science & Technology* **2007**, *7*.
41. Li, H.W.; Zhang, C.M.; Jiang, M.S.; Cai, Q.Y. Improving the performance of practical decoy-state quantum key distribution with advantage distillation technology. *Communications Physics* **2022**, *5*, 1–8.
42. Yunakovsky, S.E.; Kot, M.; Pozhar, N.; Nabokov, D.; Kudinov, M.; Guglya, A.; Kiktenko, E.O.; Kolycheva, E.; Borisov, A.; Fedorov, A.K. Towards security recommendations for public-key infrastructures for production environments in the post-quantum era. *EPJ Quantum Technology* **2021**, *8*, 14.
43. Ding, Y.Y.; Chen, H.; Wang, S.; He, D.Y.; Yin, Z.Q.; Chen, W.; Zhou, Z.; Guo, G.C.; Han, Z.F. Polarization variations in installed fibers and their influence on quantum key distribution systems. *Optics Express* **2017**, *25*, 27923–27936.
44. Fitzke, E.; Bialowons, L.; Dolejsky, T.; Tippmann, M.; Nikiforov, O.; Walther, T.; Wissel, F.; Gunkel, M. Scalable network for simultaneous pairwise quantum key distribution via entanglement-based time-bin coding. *PRX Quantum* **2022**, *3*, 020341.
45. Shor, P.W.; Preskill, J. Simple proof of security of the BB84 quantum key distribution protocol. *Physical review letters* **2000**, *85*, 441.

46. Riggs, B.; Partridge, M.; Cambou, B.; Burke, I.; Rios, M.A.; Heynssens, J.; Ghanaimiandoab, D. Multi-Wavelength Quantum Key Distribution Emulation with Physical Unclonable Function. *Cryptography* **2022**, *6*, 36.
47. Riggs, B.; Miandoab, D.G.; Cambou, B. Ternary Two Wavelength Quantum Key Distribution Protocol With Physical Unclonable Function Error Models and Eavesdropper Detection. In Proceedings of the 2024 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). IEEE, 2024, pp. 175–181.
48. Shu, H. Solving single photon detector problems. *arXiv preprint arXiv:2203.02905* **2022**.
49. Portmann, C.; Renner, R. Security in quantum cryptography. *Reviews of Modern Physics* **2022**, *94*, 025008.
50. Lo, H.K.; Ma, X.; Chen, K. Decoy state quantum key distribution. *Physical review letters* **2005**, *94*, 230504.
51. Huang, A.; Sun, S.H.; Liu, Z.; Makarov, V. Quantum key distribution with distinguishable decoy states. *Physical Review A* **2018**, *98*, 012330.
52. Sekar, R. A Report on Decoy State Quantum Key Distribution. *Raman Research Institute* **2017**.
53. Lovic, V. Quantum key distribution: Advantages, challenges and policy **2020**.
54. Bedington, R.; Arrazola, J.M.; Ling, A. Progress in satellite quantum key distribution. *npj Quantum Information* **2017**, *3*, 30.
55. Kaltwasser, J.; Seip, J.; Fitzke, E.; Tippmann, M.; Walther, T. Reducing the number of single-photon detectors in quantum-key-distribution networks by time multiplexing. *Physical Review A* **2024**, *109*, 012618.
56. Gümüş, K.; Eriksson, T.A.; Takeoka, M.; Fujiwara, M.; Sasaki, M.; Schmalen, L.; Alvarado, A. A novel error correction protocol for continuous variable quantum key distribution. *Scientific reports* **2021**, *11*, 10465.
57. Gallager, R. Low-density parity-check codes. *IRE Transactions on information theory* **1962**, *8*, 21–28.
58. Phromsa-ard, T.; Sangwongngam, P.; Sripimanwat, K.; Kaemarungsri, K.; Vanichchanunt, P.; Wuttisittikulkij, L. Low-complexity key reconciliation algorithm using LDPC bit-flipping decoding for quantum key distribution. In Proceedings of the 2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE, 2014, pp. 1–5.
59. Benletaief, N.; Rezig, H.; Bouallegue, A. Toward efficient quantum key distribution reconciliation. *arXiv preprint arXiv:2002.04887* **2020**.
60. Nguyen, K.C.; Van Assche, G.; Cerf, N.J. Side-information coding with turbo codes and its application to quantum key distribution. *arXiv preprint cs/0406001* **2004**.
61. Matsumoto, W.; Watanabe, Y. Quantum key delivery method and communication device, 2008. US Patent 7,461,323.
62. Choi, J.W. Permutation method for correcting bit error in quantum key distribution protocol, 2018. US Patent 10,103,881.
63. Johnson, J.S.; Grimaila, M.R.; Humphries, J.W.; Baumgartner, G.B. An analysis of error reconciliation protocols used in quantum key distribution systems. *The Journal of Defense Modeling and Simulation* **2015**, *12*, 217–227.
64. Devitt, S.J.; Munro, W.J.; Nemoto, K. Quantum error correction for beginners. *Reports on Progress in Physics* **2013**, *76*, 076001.
65. Roffe, J. Quantum error correction: an introductory guide. *Contemporary Physics* **2019**, *60*, 226–245.
66. Nielsen, M.A.; Chuang, I.L. *Quantum computation and quantum information*; Cambridge university press, 2010.
67. Duan, L.M.; Guo, G.C. Preserving coherence in quantum computation by pairing quantum bits. *Physical Review Letters* **1997**, *79*, 1953.
68. Cao, Y.; Zhao, Y.; Wang, Q.; Zhang, J.; Ng, S.X.; Hanzo, L. The evolution of quantum key distribution networks: On the road to the qinternet. *IEEE Communications Surveys & Tutorials* **2022**, *24*, 839–894.
69. Wang, J.; Huberman, B.A.; et al. An overview on deployment strategies for global quantum key distribution networks. *Wireless Communications and Mobile Computing* **2022**, 2022.
70. Takeoka, M.; Guha, S.; Wilde, M.M. Fundamental rate-loss tradeoff for optical quantum key distribution. *Nature communications* **2014**, *5*, 5235.
71. Lo, H.K.; Zhao, Y. Quantum cryptography. *arXiv preprint arXiv:0803.2507* **2008**.
72. Herder, C.; Yu, M.D.; Koushanfar, F.; Devadas, S. Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE* **2014**, *102*, 1126–1141.
73. Gao, Y.; Al-Sarawi, S.F.; Abbott, D. Physical unclonable functions. *Nature Electronics* **2020**, *3*, 81–91.
74. Ghanai Miandoab, D.; Garrett, M.L.; Alam, M.; Jain, S.; Assiri, S.; Cambou, B. Secure Cryptographic Key Encapsulation and Recovery Scheme in Noisy Network Conditions. *Applied Sciences* **2025**, *15*, 2732.

75. Cambou, B.; Philabaum, C.; Hoffstein, J.; Herlihy, M. Methods to Encrypt and Authenticate Digital Files in Distributed Networks and Zero-Trust Environments. *Axioms* **2023**, *12*, 531.
76. Lee, C.; Sohn, I.; Lee, W. Eavesdropping detection in BB84 quantum key distribution protocols. *IEEE Transactions on Network and Service Management* **2022**, *19*, 2689–2701.
77. Biham, E.; Boyer, M.; Boykin, P.O.; Mor, T.; Roychowdhury, V. A proof of the security of quantum key distribution. *J. Cryptol.* **2006**, *19*, 381–439.
78. Cambou, B.F.; Hoffstein, J.; Ghanaimiandoab, D.; Riggs, B.; Burke, I.; Heynssens, J.B. Error management for long distance quantum key distribution and symmetrical key distribution, 2025. US Patent App. 18/907,586.
79. Cambou, B.F.; Garrett, M.L.; Partridge, M.; Ghanaimiandoab, D. Protocols with noisy response-based cryptographic subkeys, 2025. US Patent App. 18/885,226.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.