Baghdad Science Journal

Volume 22 | Issue 7 Article 26

7-25-2025

The Effect of QBER Estimation on the Efficiency and Security of Quantum Key Distribution

Sufyan Al-Janabi
Department of Computer Network Systems, College of Computer Science and IT, University of Anbar, Ramadi, Iraq, sufyan.aljanabi@uoanbar.edu.iq

How to Cite this Article

Al-Janabi, Sufyan (2025) "The Effect of QBER Estimation on the Efficiency and Security of Quantum Key Distribution," *Baghdad Science Journal*: Vol. 22: Iss. 7, Article 26.
DOI: https://doi.org/10.21123/2411-7986.5006

BAGHDAD SCIENCE JOURNAL 2025;22(7):2412–2428

Scan the QR to view
the full-text article on
the journal website

RESEARCH ARTICLE

The Effect of QBER Estimation on the Efficiency and Security of Quantum Key Distribution

Sufyan Al-Janabi

Department of Computer Network Systems, College of Computer Science and IT, University of Anbar, Ramadi, Iraq

ABSTRACT

Quantum Key Distribution (QKD) is a prominent solution to achieve proven secure data transmission between legitimate users despite any assumptions about the eavesdropper's computational power (given that the laws of quantum mechanics are correct). This remains true even under attacks by quantum computers. In recent years, there has been a great research interest in developing enhanced security constructions based on QKD. This is mainly driven by the envisioned threats of quantum computers to the traditional information security infrastructure based on public key cryptography. This research lies under the wide umbrella of the so-called post-quantum cryptography. However, for any QKD protocol, an accurate estimation of the Quantum Bit Error Rate (QBER) before the information reconciliation phase is crucial for both the efficiency and security perspectives. In spite of the importance of the QBER estimation step, the state-of-the-art literature still lacks a detailed investigation of the practical aspects of this issue. Therefore, this work aims to fill this research gap. An extension of the EnQuad simulator has been used to study the effect of QBER estimation on the efficiency and security of the BB84 QKD protocol for depolarizing quantum channels and individual Intercept/Resend Eve attacks. Various parameters affecting the QBER estimation process have been carefully investigated. Our evaluation has shown the significant importance of the QBER estimation step. Indeed, an empirical formula has been derived for calculating the required size of the sifted key subset (the so-called sharing rate) needed for the QBER estimation. This is a major contribution of the presented work.

Keywords: Key reconciliation, QBER, QKD, Quantum cryptography, Unconditional security

Introduction

It is quite prudent now to expect the appearance of universal quantum computers in the near to mid-future. These quantum computers can solve certain mathematical problems much more efficiently than their classical counterparts. This particularly applies to the cryptographic systems based on discrete logarithm and integer factorization problems. Therefore, there is a paradigmatic change toward the so-called quantum-safe technologies, which include the security tools that have proven to be invulnerable to attacks by quantum computers. Such techniques are usually divided into two groups:[1,2]

- Information-theoretically (or unconditionally) secure techniques that make no assumptions about the computational power of an eavesdropper (e.g., the one-time pad encryption and Wegman-Carter authentication codes)
- Post-quantum cryptography primitives that include tools based on certain computational problems that are believed to be immune against attacks from both classical and quantum computers.

This work is related to the first group of techniques mentioned above that include the use of tools providing unconditional security such as the one-time pad encryption scheme. However, the one-time pad requires using a random cryptographic key as long as the message. Indeed, the key should be used only once and then must be carefully destroyed. Furthermore, the one-time pad as a symmetric

Received 22 January 2024; revised 31 March 2024; accepted 2 April 2024.
Available online 25 July 2025

E-mail address: sufyan.aljanabi@uoanbar.edu.iq (S. Al-Janabi).

https://doi.org/10.21123/2411-7986.5006

system requires sharing initial secret key information in advance between the correspondents. Therefore, key management and distribution is significantly difficult for this kind of system.[3]

Fortunately, it is possible to overcome this difficulty using Quantum Key Distribution (QKD). QKD is an elegant technique for generating secret random keys between legitimate users (Alice and Bob) using individual quantum systems (e.g., photons). It is important to notice that there is no mechanism in the classical world to prevent an eavesdropper (Eve) from copying the cryptographic keys during their transmission. In contrast, the QKD technology uses quantum objects as information carriers. Because of the quantum no-cloning theorem, Eve cannot keep a copy of the sent quantum signals. In other words, QKD can offer information-theoretic (unconditional) security based on the laws of physics.[1]

The significant attention to QKD is mainly related to the fact that the security of public-key cryptosystems is justified based on the complexity of certain mathematical problems. Meanwhile, some of these problems can be solved in polynomial time using Shor's algorithm on a quantum computer. In fact, there is no mathematical proof of the unavailability of efficient classical (non-quantum) algorithms for breaking public-key systems.[4] In the recent two decades or so, there has been a noticeable advancement in the design of QKD devices for practical deployment of one-time pad encryption. Moreover, QKD devices have been combined with other (classical) cryptographic primitives to produce hybrid classical-quantum encryption systems. Furthermore, some commercial QKD products are now widely available on the global market.[2–5]

For the time being, on the basis of the current market demand, there is a consensus among market analysts and researchers that QKD will be important for many communication technologies including the Internet security infrastructure. Such a relationship between classical and quantum cryptography is predicted to affect several sectors shortly.[6,7] It is important to emphasize that QKD is used to distribute the encryption keys for symmetric ciphers, yet QKD is not used to transmit any message data between legitimate users. A vast of QKD protocols have been suggested. However, the first invented protocol called the BB84 (after Bennet and Brassard) protocol, which is based on single quantum particles (e.g., polarized photons), is the most widely applied solution for QKD in practice. Many other protocols (e.g., the BBM92 and SARG04) can be considered as modified versions of this protocol.[8]

In the BB84 QKD protocol, Alice and Bob need to test the Quantum Bit Error Rate (QBER) to decide whether Eve is trying to gain information about the secret key. The QBER value can be influenced by two-factor types. The first is Eve's eavesdropping strategies. The second is the combined effect of the imperfections of the (quantum) transmission channel and physical devices used for sending and receiving the quantum states.[9] It is obvious that the accuracy of estimating the QBER can strongly affect both the efficiency and security of the whole QKD protocol. On the one hand, if the QBER is underestimated, Eve's actions might result in a significant increase in the required communication rounds for key reconciliation (as will be explained later). On the other hand, if QBER is overestimated, the efficiency of the protocol will be much less than the theoretical rate.[5]

It is well-known that a typical experimental QKD implementation requires special and costly hardware. Such hardware is unavailable in most university lab environments. Hence, this highly motivates the development of simulation packages for studying QKD. In this way, the researchers can obtain first insights into different aspects of the protocol before proceeding into experimental implementations. Over recent years, several QKD simulator frameworks have been introduced. These simulators have mainly focused on the BB84 protocol as being the first and the most explored QKD protocol.[10]

The aim of this work is to investigate the accuracy of QBER estimation and its effect on the efficiency and security of QKD. The BB84 protocol is considered in this study and an extension of the EnQuad Simulator[11] is used to achieve the required evaluation. Based on the analysis of the obtained results, an empirical formula has been derived for the best size of the permuted part of the sifted key used for QBER estimation.

Literature survey

Recently, there has been an increasing research and industrial interest in QKD. Plenty of papers studied the various aspects of QKD including modeling, simulation, protocol design, key reconciliation, authentication, experimental evaluation, etc. Many of these studies mentioned the importance of the QBER estimation issue. However, to the best of our knowledge, there is almost no detailed study of the effect of QBER estimation accuracy on the QKD protocol nor the best approach to increase this accuracy.

Some literature focused on mathematical modeling for studying QKD parameters. In reference,[9] the authors developed a simple quantum channel model with polarization switching to investigate the effects on the QBER in BB84 implementation. Other researchers modeled a high-bit rate QKD system and used numerical investigation to show the impact of chromatic on QBER and the obtained secret key rate.[12] Inspired by the similarity between QKD and mobile ad-hoc networks (MANETs), the work in reference[13] proposed a model for the Quality of Service (QoS) provisioning in trusted relay QKD networks. Moreover, some researchers considered modelling QKD over satellite networks. This is quite interesting but still a challenging issue. In this respect, a double-layer quantum satellite network (QSN) has been proposed to relay cryptographic keys for ground stations.[14]

The first successful quantum hacking on commercial QKD systems was reported in 2010. Such attacks are possible mainly due to device imperfections. Therefore, based on analyzing quantum hacking, new QKD devices were developed resulting in the so-called Measurement-Device-Independent (MDI) QKD. Indeed, this gives ignition to security-proof studies that warned about the existence of some problems in QKD theories. Hence, standardized security criteria were proposed. One of the most widely adopted criteria now is the trace distance between an ideal quantum state and a quantum state in actual situations (i.e., when the quantum system of Eve is correlated to the shared quantum system between Alice and Bob).[15]

The development and implementation of QKD networks is another interesting research direction in this area. Besides stand-alone systems, QKD can also be integrated within the physical layer of optical free-space communication networks. This can facilitate the current key-management infrastructures. Various studies considered the integration of QKD capabilities within the Internet (or intranet) security architecture based on extending well-known security protocols like SSL/TLS and IPSec.[7] Here, one can notice that there is still a subtle distinction between open “classical” networks (e.g., the Internet) and “closed” QKD networks, mainly because of the physical limitations of QKD settings. Thus, for the time being, these extended security protocols can be more applicable in private intranets rather than the public Internet.[16] Nevertheless, researchers are investigating the use of QKD for enhancing various networking paradigms and technologies including wireless sensor networks (WSNs),[17] cloud computing,[18] software-defined networking (SDN),[19] and the smart grid.[20]

The QBER estimation has a direct effect on the reconciliation stage, which is one of the classical post-processing phases of QKD. Here, the iterative Cascade protocol is the most widely implemented reconciliation protocol because of its simplicity and efficiency.[3,21] Other recently developed reconciliation protocols are based on using Low-Density Parity Check (LDPC) codes that can achieve improvement of the achievable secret key rate.[4,21] For reconciliation based on LDPC codes, syndrome-based algorithms for QBER estimation can be developed.[5] Also, it is possible to use the knowledge of the QBER on the basis of the maximum-likelihood estimating algorithm respecting the obtained syndrome information.[22] In addition, it was shown that information reconciliation based on polar codes can increase the reconciliation efficiency.[23,24] Furthermore, some researchers used eavesdropping detection based on a QBER threshold in order to analyze the upper bounds of the false-positive and false-negative eavesdropping detection ratios in the BB84 protocol.[25]

Due to the obvious advantages that simulation can offer for QKD experimentation and validation, researchers developed several QKD simulation packages; however, the majority of them are not publicly available. Some of them followed the object-oriented approach using C++ programming language to maintain the modularity of the software.[16,26] The EnQuad is a publicly available simulator that was developed using MatLab to facilitate the process of designing QKD setups.[11] Another modular QKD simulator was presented using open-source LabView and Python codes with the possibility of modifying optical schemes.[1] Moreover, the NuQKD simulation framework is a Python script package that was built to simulate the BB84 protocol between two computer terminals.[10]

At last but not at least, several research groups worked on experimental QBER testing and validation. In reference,[27] the QBER was investigated as a function of the optical quantum channel length based on three-stage multi-photon protocols. Indeed, the work in reference[28] studied the effect of the time-jitter on the QBER for the case of coupling a commercial single-photon avalanche diode with multimode fibers. They concluded that using graded-index multimode fiber can be beneficial for a free-space high-rate QKD receiver. Finally, a real-time implementation using FPGA was presented for an underwater BB84 QKD system. They successfully achieved a secret key distribution rate of 100 bps over a 7-meter link distance in their experiments.[29] This rate is quite lower than the achievable rates in fiber-optic or free-space networks; however, it still promising for underwater communications.

Materials and methods

The security of QKD is based on two quantum principles, which are the Heisenberg uncertainty principle and the no-cloning theorem. The first principle declares the impossibility of precisely measuring two uncorrelated physical quantities. For example, when the position of a particle is measured in some dimension, its momentum can only be measured with a limited precision along that dimension. The second one states the impossibility of creating a device that can clone the state of a quantum bit (qubit) without changing its original state. Thus, quantum systems can be protected from passive eavesdropping.[26]

Since the introduction of the BB84 protocol in 1984 and its first experimental implementation in 1989, a myriad of variant QKD protocols around BB84 were proposed such as SARG04 and Decoy-state protocol.[11,16] For implementing the BB84 protocol, the required resources are the preparation, transmission, and measurement of single-qubit states. However, the BB84 protocol has an equivalent entanglement-based implementation. For security proof formalization, the entanglement-based version of the BB84 protocol played a key role. Nevertheless, it is possible to show that there is an equivalent implementation based on single-qubit states.[30] Thus, the polarization states of photons are usually utilized in most of the QKD protocols today.

The following points can be provided by quantum physics:[27]

- In general, it is impossible to obtain any precise information about the single photon's unknown polarization state.
- When the polarization of the photon is measured, its polarization value is irreversibly altered.
- Based on the no-cloning theorem, it is impossible to copy an unknown quantum state of a single photon.

In order to obtain a single photon per laser pulse, attenuation of the laser beam is usually used. Therefore, in practice, there is a probability that an emitted laser pulse can contain no photons, a single photon, or more than one photon. Thus, there is a possibility of using photon number splitting attacks (PNS) on the BB84 protocol when more than one photon is emitted per bit in a practical setting. For such practical considerations, hacking was reported on some commercial BB84-based products despite that patches were soon implemented by the manufacturers.[27]

Ideally, QKD has the ability to generate information-theoretical secure keys between distant legitimate parties (Alice and Bob). Assume that the sifted keys (the keys after the quantum transmission phase, as will be explained in the next subsection) of Alice and Bob are $K^A{}_s$ and $K^B{}_s$, respectively with length of n. Hence, $K^A{}_s \neq K^B{}_s$ with the QBER (ϵ) that is introduced due to the imperfections of QKD systems and potential Eve attacks. The aim of information reconciliation post-processing phase is to reconcile $K^A{}_s$ and $K^B{}_s$ to be an equally weak secure key (K_r). Finally, privacy amplification is used to produce the final (shorter) secret key (K).[23]

The BB84 protocol

In general, it is possible to describe the BB84 protocol by the following main steps:[16,30]

1. *Quantum transmission:* Alice uses the quantum channel to send a random string of bits (single photons). She randomly (with equal probability) chooses one of the orthogonal alphabets for each photon. To perform the photon measurement, Bob (randomly and independently) uses one of the orthogonal polarizations for each photon sent by Alice. The sequence of used bases and his measurement results are recorded by him.
2. *Sifting (Raw key extraction):* Bob uses the public channel to announce his measurement operators used for each of the received photons. Then, Alice publicly tells Bob to keep the bits for which of the measurement operators were correct. This step results in the sifted keys $K^A{}_s$ and $K^B{}_s$.
3. *Parameter estimation:* Alice and Bob sacrifice m bits in order to estimate their average correlation (estimating the QBER) and decide whether to abort or proceed with the protocol (They may select a random sample of raw key and publicly compare these sample bits). The remaining bits will constitute the updated version of the sifted keys ($K^A{}_s$ and $K^B{}_s$) when the protocol is not aborted.
4. *Advantage distillation* (optional): More classical post-process can be used here to increase the correlation between $K^A{}_s$ and $K^B{}_s$ and get an advantage over Eve.
5. *Information reconciliation:* Using the public channel, Alice and Bob implement an information reconciliation protocol to produce an error-free common key, which is called the reconciled key (K_r).
6. *Privacy amplification:* Finally, Alice and Bob apply a privacy amplification protocol to extract the common final secret key (K) from a partially secret one (K_r).

A schematic of a typical experimental BB84 QKD setup is shown in Fig. 1, where Alice and Bob have access to a quantum channel and an authentic classical public channel (for classical post-processing). A Single Photon Source (SPS) is assumed to be used by Alice to generate the quantum pulses, while Bob

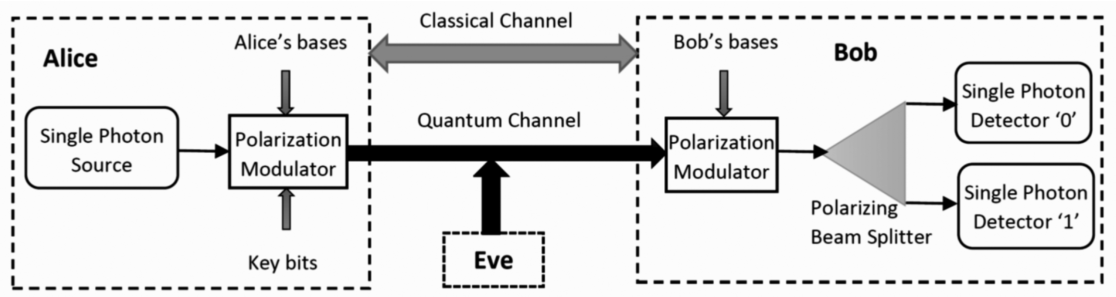


Fig. 1. Schematic of a typical experimental BB84 setup.

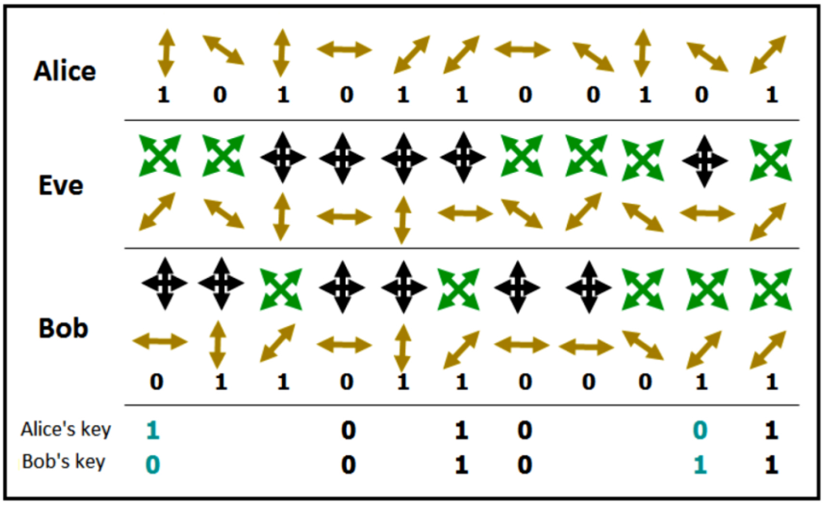


Fig. 2. Eavesdropping effect in the BB84 protocol.[8]

uses a Polarizing Beam Splitter (PBS) and two Single Photon Detectors (SPDs) for detection. Both of them use polarization modulators to encode and decode the photon polarization states.

When Alice starts establishing a new key with Bob, they both define two alphabets, e.g., rectilinear and diagonal. They also may define photons with horizontal polarization (0°) to mean bit 0 and photons with vertical polarization (90°) to mean bit 1 in the rectilinear polarization alphabet. Similarly, they can consider photons with polarization – 45° to mean bit 0 and photons with polarization 45° to mean bit 1 in the diagonal polarization alphabet. In Fig. 2, two crossed double-headed (black) arrows represent a rectilinear basis, while two crossed double-headed diagonal (green) arrows represent a diagonal basis (green). Indeed, the double-headed (brown) arrows mean the polarization states of single photons. For example, when a photon with diagonal polarization 45° is observed using the rectilinear basis, the measurement produces either horizontal or vertical polarization with probability 1/2. This means that the measurement in the rectilinear basis causes losing information about diagonally polarized sent photons. A similar argument applies to the case of using the diagonal basis for observing horizontally and vertically polarized photons.

The protocol starts when Alice sends Bob a string of bits that is encoded using photon polarization (qubits). She uses randomly chosen bases to send the bits through the quantum channel. Bob randomly and independently chooses the basis to measure the received string. Then, he informs Alice about the bases he used via the public channel, keeping the results of the measurement secret. Next, Alice uses the public channel to inform Bob to only keep those bits for which he has chosen the same basis of her. Ideally, such an algorithm results in that the obtained key consists of approximately 50% of the bits sent by Alice.[8]

However, when Eve eavesdrops on the quantum channel, she can try to randomly measure the photons' polarization using either a rectilinear or diagonal basis (In fact, more general measurement attacks are possible, but Alice and Bob can always deal with them). If the chosen basis by Eve is incorrect, the polarization will irreversibly be changed, as depicted in Fig. 2. Hence, Alice and Bob can uncover eavesdropping by comparing a part of the key using the public channel. This means that passive eavesdropping is not possible because Eve will irreversibly change the quantum states of the sent photons. Moreover, Eve cannot clone an unknown quantum state of the photon. This is why the BB84 QKD protocol can offer a significantly high level of security.[3]

Eavesdropping causes errors in the quantum channel. However, errors during quantum transmission can occur due to other reasons such as optical misalignment, channel disturbance, or noise in SPDs. The number of errors of the raw quantum transmission in QKD systems (i.e., QBER) can be calculated as the ratio of the number of mismatched bits to the total number of bits in sifted keys as Eq. (1). Hence,

$$\varepsilon = \frac{Number\ of\ mismatched\ bits}{Total\ number\ of\ bits\ of\ the\ sifyed\ key} \times 100\% \tag{1}$$

Based on the estimated QBER, Alice and Bob will decide if there is an eavesdropper or not. This can be done by comparing a small portion of the sifted key using the public channel and computing the QBER (They must discard the compared part). When the QBER exceeds a certain predetermined threshold, it may indicate that eavesdropping happened on a large part of the quantum transmission. Thus, the QKD protocol is to be aborted and initialized again. However, if the QBER value is acceptable, Alice and Bob can continue to apply key distillation protocols. Such protocols typically involve information reconciliation and privacy amplification protocols.[8] Information reconciliation aims to correct the erroneous key bits due to eavesdropping or channel noise. In privacy amplification, more key bits are sacrificed to maximize the security against Eve's information gained from the bits revealed during information reconciliation. Thus, the final secret key (K) of Alice and Bob is obtained.[11]

The common procedure for estimating the QBER in the BB84 protocol involves using the public channel to compare a random permuted subset of the sifted keys between Alice and Bob. It has been shown in the literature that the most common QBER threshold for deciding the continuum of the QKD protocol is about 15%. At this QBER value, Eve could have intercepted more than half of the qubits transmitted. Some researchers proposed a modification of the QBER estimation process by using two samples of qubits, one from each polarization basis, and then obtaining an estimate for each. Thus, QBER estimation is an important part of QKD protocols.[31] Finally, one has to note that the authenticity of the public channel is crucial for the BB84 protocol. Hence, authentication is necessary for sifting, error estimation, reconciliation, and privacy amplification. However, encryption for information reconciliation is optional depending on the later privacy amplification procedure.[32]

The EnQuad simulator

EnQuad is a publicly available QKD simulator for QKD protocols, where its v1.0 code is available on GitHub. EnQuad simulates the BB84-based QKD stack with single-photon quantum signals. Indeed, it does security testing and gives guidance to the researcher on the required efficiency of information reconciliation to be used. Moreover, EnQuad can recommend some changes to the assumed QKD setting in order to satisfy certain security levels or target key rates when any of these are not fulfilled. EnQuad v1.0 considers depolarizing quantum channels and individual qubits' Intercept-and-Resend attacks. The simulator is also featured with nine modular functions and 24 main parameters. Thus, it is possible to expand its functionality into other variants of QKD protocols. In addition, EnQuad outcomes were validated against comparable simulators and the theory. Furthermore, a set of 11 experiments showed that EnQuad runs 6.12× to 12.2× faster than a comparable simulator. EnQuad v1.0 was built using MATLAB with clear interfaces and relatively fast simulation speed. The EnQuad v1.0 parameters and their allowable values are shown in Table 1.[11]

Error rate estimation in QKD

The aim of a QKD protocol between Alice and Bob is to share an unconditionally secure secret key despite the power of the eavesdropper Eve. In the quantum transmission phase of the protocol, Alice sends Bob quantum signals (e.g., single photons) that carry classical information. Alice can encode a classical bit onto the photon's polarization and send it to Bob for measurement. Repeating this step r times, Alice and Bob will share two r-bit strings, A and B. However, Eve can also have access to a random variable E that is possibly in correlation to A and B.

In realistic QKD implementation, A and B would suffer discrepancies due to various reasons such as channel losses, noise of Bob's detectors, and conservatively attributed to Eve's actions. Hence, QKD protocols include classical post-processing phases to extract the final secret key (K) from the correlated strings A and B. This is achieved using a public authenticated channel. If the protocol succeeds, K will be uncorrelated with E. The theoretical secret

Table 1. Parameters' values of EnQuad v1.0.[11]

Parameter name	Allowable values
Number of photons	Real positive integer
Key bits	1D array of zeros/ones
Alice basis selection	1D array of zeros/ones
Alice polarized photons states	1D array of 1/2/3/4
Eve's attack level	Real positive number from 0 to 1
Photons states after Eve's attack	1D array of 1/2/3/4
Channel depolarization probability	Real positive number between 0 and 0.25
Photons states after channel	1D array of 1/2/3/4
Bob basis selection	1D array of zeros/ones
Photon states after beam splitter	1D array of 1/2/3/4
Alice photos states after polarizing beam splitter	1D array of 1/2/3/4
Measured bits	1D array of zeros/ones
Alice sifted key	1D array of zeros/ones
Bob sifted key	1D array of zeros/ones
QBER	Real positive number from 0 to 1
Simulation time	Real positive number
Theoretical key rate	Real positive number from 0 to 1
Target key rate	Real positive number from 0 to 1
Mutual information between Alice and Bob	Real positive number from 0 to 1
Mutual information between Alice and Eve	Real positive number from 0 to 1
Security Decision	Text
Lower bound of reconciliation efficiency for a target key rate	Real positive number from 0 to 1
Recommendation on depolarizing channel parameter change to achieve security (if not already satisfied)	Text
Recommendation on depolarizing channel parameter change to achieve target key rate (if not already achievable)	Text

capacity K_{th} is given by Eq. (2):[21]

$$K_{th} = H(A|E) - H(A|B) \tag{2}$$

The definition of H(A|E) is changed depending on the attack type, whereas H(A|B) is the conditional entropy of A given B. This capacity is theoretical and can only be achieved in the case of a perfect information reconciliation scheme. Note that H(A|B) represents the minimum amount of (classical) information that Alice is required to send to Bob so that he corrects his string B. However, in any realistic implementation, the actual secret key rate (K_{real}) can be given by Eq. (3):[21]

$$K_{real} = H(A|E) - fH(A|B) \tag{3}$$

where f is an efficiency parameter with a value greater than 1 related to the actual reconciliation procedure.

Quantum technologies are still suffering from imperfections. Thus, QKD real implementations are subjected to noise even in the absence of Eve. Therefore, it is interesting to design protocols that are able to tolerate the noise levels of current technology and simultaneously result in the highest possible secret key rates.[30] In order to estimate the quantum channel's error rate (QBER), Alice and Bob reveal to each other a sample of the sifted keys ($K^A{}_s$ and $K^B{}_s$). QBER is a significantly important feature of QKD because it can be used to measure the joint probability distribution among Alice, Bob, and Eve. This cannot be decided within the scenarios of classical key agreement.[22] As mentioned previously, QBER is calculated when obtaining the sifted keys according to Eq. (1).

It is prudent to mention that several coding-based reconciliation schemes have been proposed. To optimize the coding rate of these schemes, one has to use sample keys for estimating the QBER. Moreover, some researchers proposed using an interactive LDPC-based reconciliation method that can work without needing a priori QBER estimation. However, this method requires heavy interactive communications.[22] As LDPC-based reconciliation methods need QBER estimation, inaccurate QBER estimation can either decrease the scheme efficiency or result in a higher number of communication rounds. It is possible to use a QBER estimate from the previous QKD stages or to employ a typical QBER default value, but anyway, this decreases the key generation rate because legitimate parties have to disclose some bits of the sifted keys used for QBER estimation. In some LDPC-based schemes, an on-the-fly QBER estimation technique that uses syndromes of the parties can be applied to estimate the QBER. However, to increase the accuracy of such estimation, an a priori QBER distribution is employed and/or this approach is combined with the QBER estimation using previous rounds.[4,5]

In a typical BB84 protocol, the sifted key size is statistically about 50% of the raw key length. During the parameter estimation phase, additional bits will be shared to identify mismatches between $K^A{}_s$ and $K^B{}_s$. When the number of shared bits is increased, a more accurate QBER estimation is obtained. However, this results in keys shorter in size. This ultimately reduces the QKD key generation rate. After the information reconciliation and privacy amplification stages, the total reduction of the key can be expressed by the secret-key rate (SKR), which is an important protocol parameter that describes the performance of the whole QKD scheme. SKR is usually defined as:[10]

$$SKR = \frac{length\ of\ K}{lenght\ of\ K_s^A (or\ K_s^B)} \times R_s \tag{4}$$

where R_s is the resultant key rate after the sifting phase. Practically, the final secret key (K) is always less than the sifted keys ($K^A{}_s$ and $K^B{}_s$) because some bits must be shared (and sacrificed) between Alice and Bob. When the noise in the quantum channel and/or the attack level are higher, more bits need to be shared to obtain an error-free final key. This definitely reduces SKR. The theoretical secret-key rate can be defined as:[11]

$$K_{th} = h\left(\frac{1}{2} - q_e\right) - h(\varepsilon) \tag{5}$$

where q_e represents the level of attack, ϵ is the QBER, and the terms on the right-hand side in the equation are expressed using Shannon binary entropy.

The quantum channel noise (e.g., depolarization), devices' imperfections and/or Eve's actions introduce inhomogeneity between Alice and Bob's sifted keys. Thus, classical post-processing is used to match the final two keys. In the parameter estimation phase, Alice and Bob use the public channel to share and compare part of the sifted bits. This part of the sifted key will be sacrificed and cannot be used for encryption/authentication afterward. The result of comparing the values of the shared bits is an estimated value of the QBER that can be denoted as ϵ_{est}, which is described by Eq. (6):

$$\varepsilon_{est} = \frac{Number\ of\ erroneous\ bits\ in\ the\ shared\ part}{Total\ number\ of\ bits\ of\ the\ shared\ key\ part} \times 100\% \tag{6}$$

Comparing Eqs. (1) and (6), it would be desirable to have the value of ϵ_{est} be as close as possible to ϵ. In other words, it is quite desirable to have a QBER estimation procedure that results in an estimated value of QBER that is almost equal to the actual value of the QBER. Hence, an important question is raised here about the minimum fraction of shared bits required to obtain an accurate QBER estimation on the remaining bits of the key. On the one hand, as the length of the shared part of the sifted key increases, the estimation precision improves. On the other hand, this would decrease the key distribution rate because it reduces the number of remaining bits. Thus, modeling this behavior is important to determine the optimal ratio of shared bits' fraction versus the length of the sifted key. In this respect, it is possible to define the sharing rate (F_{sh}) as the number of bits shared between Alice and Bob for the QBER estimation as a fraction of the size of the sifted key.[10] Thus,

$$F_{sh} = m/n \tag{7}$$

A necessary step in all QKD protocols is to check whether Eve has performed any eavesdropping actions. Since errors in the raw quantum transmission can occur not only due to eavesdropping but also because of various system and channel imperfections, it is crucial to know in advance the threshold QBER (ϵ_{max}) for the quantum channel under various operating conditions without the existence of Eve. The value of ϵ_{max} depends on various QKD system imperfections related to polarization state preparation at the source, polarization reference misalignment, quantum channel disturbance, imperfect PBSs, single-photon detectors' dark counts and noise, and stray background light.[33] These issues apply to both fiber-optic and free-space QKD settings. However, the calculation of ϵ_{max} can be a little bit trickier in the latter case due to the possible changes in such an environment. After completing the parameter estimation phase, the estimated QBER value (ϵ_{est}) is compared with the already known ϵ_{max} (QBER threshold value). When the estimated value of QBER is higher than the threshold ($\epsilon_{est} > \epsilon_{max}$), this will be evidence of Eve's presence. Then, depending on the estimated level of eavesdropping, a decision can be made whether to continue the classical post-processing procedure (reconciliation and privacy amplification) or to abort the protocol and restart the process from the beginning.

Considering the well-known reconciliation protocol 'Cascade' for a sifted key $K^A{}_s$ (or $K^B{}_s$) length of n bits, the number of bits (m) used for estimating the QBER will depend on the sample block length used for QBER estimation, which is defined by a parameter called "level of security" S(m). According to reference,[8] it is possible to define two levels of security which are basic and advanced. Selecting the suitable level of security, Alice and Bob can use Eq. (8) to calculate m:[34]

$$S(m) = \frac{-\sum_{m=1}^{n} \frac{m}{n} log\frac{m}{n}}{n} \tag{8}$$

Algorithm 1: The algorithm of the 'QBER_Estimation' modular function.

```
Input: The sifted keys of Alice and Bob ($K^A{}_s$ and $K^B{}_s$), and the sharing rate ($F_{sh}$)
Output: The estimated QBER ($\epsilon_{est}$)
Start
Step 1: Alice calculates a random permutation of $K^A{}_s$
Step 2: Bob calculates a random permutation of $K^B{}_s$
Step 3: Calculate the size (m) of the key subset to be compared round to the least integer; $m = \lfloor F_{sh} \times n \rfloor$
Step 4: Use the public channel to compare randomly chosen m bits of the sifted keys
Step 5: Apply Eq. (6) to calculate $\epsilon_{est}$
End
```

As Alice and Bob must sacrifice the m bits used for QBER estimation, the sifted key will be shortened further. Empirical studies showed that the optimal value of the initial block length is $0.73/\epsilon$.[34]

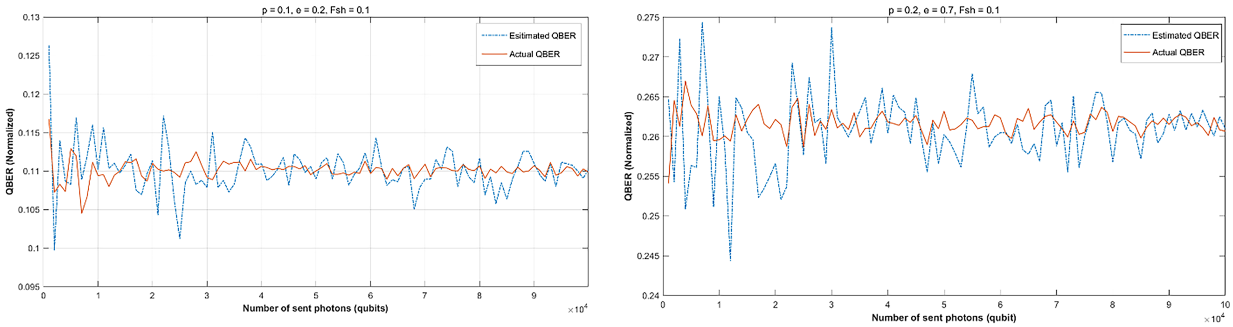
Based on the analysis above, one can deduce the importance of QBER estimation for any QKD protocol. In particular, choosing a suitable sharing rate value for F_{sh} can significantly affect the efficiency and security of the whole protocol. This effect is multifold. At first, if m is large (F_{sh} is large) it will give good QBER estimation ($\epsilon_{est} \approx \epsilon$), but this will considerably reduce the (actual) length of the sifted key as these m bits need to be sacrificed. Hence, the key rate generation of the QKD protocol would be decreased. This is a major efficiency concern. Secondly, if the chosen value of F_{sh} is small, this might result in an inaccurate QBER estimation. Therefore, Eve could avoid uncovering when Alice and Bob reach the wrong estimation that $\epsilon_{est \leq}$ ϵ_{max}, while the actual case is that $\epsilon > \epsilon_{max}$. Hence, they will continue the protocol without noticing the presence of Eve. This is a crucial security concern and could be the worst scenario for a QKD protocol. Finally, some important parameters of information reconciliation are calculated based on the estimated QBER (ϵ_{est}). Thus, inaccurate QBER estimation would also degrade the efficiency and security of the subsequent classical post-processing steps in QKD.

Results and discussion

The EnQuad simulator has been used for studying the accuracy of QBER estimation by changing the value of the number of shared bits (m) required for QBER estimation (and hence the sharing rate F_{sh}) with respect to the number of shared photons (r) and the sifted key size (n). EnQuad assumes sending single-photon signals on a depolarizing quantum channel, where Eve can do individual intercept/resend attacks. Thus, the simulation study has considered various possible values for eavesdropping level (e) and channel depolarization factor (p). Theoretically, it can be shown that when Eve interacts with the whole quantum transmission, it will unavoidably cause about 25% errors. Thus, the eavesdropping level in the simulation can take values in the range [0, 1] ($0 \leq e \leq 1$). Indeed, for all practical settings, the value of the depolarization factor was shown to be less than 0.25 ($p < 0.25$). In this latter case, QBER cannot exceed 0.5 when there is no eavesdropping.[11]

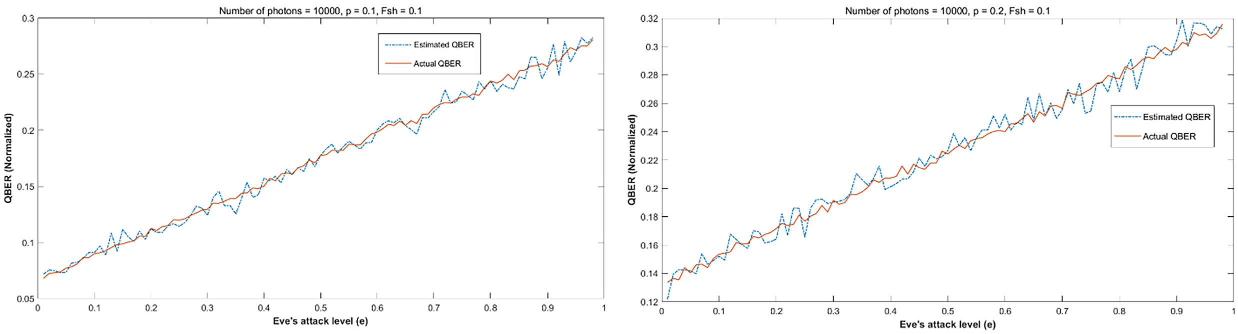
EnQuad v1.0 calculates the actual QBER value (ϵ) in the 'Sifting' modular function, as described previously in Eq. (1). However, in a realistic QKD scenario, Alice and Bob cannot calculate ϵ. Instead, they have to estimate its value and calculate ϵ_{est} as described by Eq. (6). This is done in the parameter estimation phase using the public channel. For this reason, EnQuad v1.0 has been extended by adding the modular function 'QBER_Estimation' that calculates ϵ_{est}. This function estimates the QBER based on comparing a permuted subset of the sifted keys of Alice and Bob. The 'QBER_Estimation' function starts by randomly permuting the sifted keys to prevent Eve from taking advantage of choosing any eavesdropping strategy that involves acting on certain parts of the quantum transmission. Then, the function takes the (negotiated) sharing rate (F_{sh}) as an input to decide the size of the sifted key fraction to be compared by Alice and Bob. This is described in Algorithm 1.

The 'QBER_Estimation' function has been included in the calculation of the total simulation time using 'tic-toc' built-in MATLAB function, which is the same function used by EnQuad. However, the time overhead required for the communication on the public channel has not been included as it is mainly dependent on the actual realization. Furthermore, the simulation has used the same properties of the quantum channel implemented by EnQuad which is a polarized channel without other channel losses and with no dark detector counts. Since the only attack strategy assumed for Eve is the intercept/resend attack, this means all the qubits sent by Alice will reach Bob regardless of the possible change in their polarization states. In such a simulation environment, the length (n) of the sifted key ($K^A{}_s$ or $K^B{}_s$) will statistically be 50% of the number of the sent photons



(a) p = 0.1 and e = 0.2 **(a) p = 0.2 and e = 0.7**

Fig. 3. The effect of eavesdropping level (e) and channel depolarization factor (p) on the normalized QBER value (actual and estimated) for a constant sharing rate (F_{sh} = 0.1) and different numbers of sent photons.



(a) p = 0.1 **(a) p = 0.2**

Fig. 4. The effect of eavesdropping level (e) on the normalized QBER value (actual and estimated) for a constant sharing rate (F_{sh} = 0.1) and a number of sent photons (r = 10000).

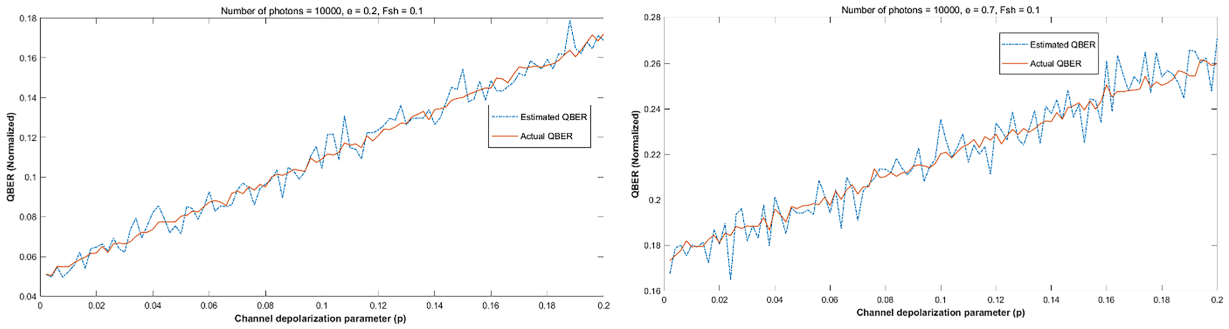
(r). Nevertheless, the value of the QBER will change depending on the values of e and p, hence affecting the length of the final key.

During experimentation, all simulation experiments have been repeated several times and the average values have been recorded. The shown values of actual QBER and estimated QBER are normalized in the range (0, 1) and the number of sent photons (r) has mostly been used rather than the length of sifted key (n) because r is typically a main input parameter for QKD simulators and hence our results are kept consistent with the previous literature. However, statistically $n \approx r/2$ as mentioned earlier.

Fig. 3 shows the effect of changing the eavesdropping level (e) and the channel depolarization factor (p) on the actual and estimated (normalized) QBER value for a constant sharing rate of F_{sh} = 0.1 and different numbers of sent photons r = (0, 100000]. One can note that when the values of p and e increased from 0.1 and 0.2 in (a) to 0.2 and 0.7 in (b) respectively, the normalized QBER values also increased from levels around 0.11 in (a) to levels around 0.26 in (b) regardless of the value of r.

The effect of changing the value of e on the actual and estimated (normalized) QBER value for a constant sharing rate F_{sh} = 0.1 and a number of sent photons r = 10000 is shown in Fig. 4. The case when p = 0.1 is shown in (a), while the case of p = 0.2 is illustrated in (b). For both cases, as the value of e is increased from 0 to 1, the normalized QBER values also increase as expected in the theoretical analysis of the BB84 protocol. Next, Fig. 5 illustrates the effect of changing the value of p on the actual and estimated (normalized) QBER value for a sharing rate F_{sh} = 0.1, a number of sent photons r = 10000, and two eavesdropping levels e = 0.2 in (a) and e = 0.7 in (b). In both cases, the increase in the p value has resulted in a similar effect to that of increasing the eavesdropping level on the QBER.

In this work, the most important parameter that needs to be investigated is the sharing rate. Thus, the effect of all its possible values in the range F_{sh} = (0, 1) has been considered on the actual and estimated (normalized) QBER values when e = 0.2 and p = 0.1 in Fig. 6. These two values for the eavesdropping level and channel depolarization have been chosen as typical because the effect of changing e and p on the QBER is behaving like the theoretical expectations, as shown in Figs. 3 to 5. The cases of different values of the number of sent photons



(a) e = 0.2

(a) e = 0.7

Fig. 5. The effect of channel depolarization factor (p) on the normalized QBER value (actual and estimated) for a constant sharing rate (F_{sh} = 0.1) and a number of sent photons (r = 10000).

r = 500, 1000, 2500, 5000, 10000, and 100000 are depicted in Fig. 6(a)–(f), respectively. It is obvious that as F_{sh} value increases from 0 (no subset is chosen for comparison, which is meaningless) to 1 (the whole sifted key is used for comparison, which is not practical), the values of the estimated QBER (ϵ_{est}) are becoming closer and closer to the actual values (ϵ). This simply happens because as the ratio of the compared subset size increases, the accuracy of estimation indeed increases. However, this can cause a degradation in the resultant secret key rate.

In Fig. 6(a)–(f), one can notice that the values of ϵ_{est} are close to ϵ almost in all cases when $F_{sh} > 0.1$. However, there are noticeable deviations in the range when $F_{sh} < 0.1$. Therefore, more investigation has been done in Fig. 7 for the effect of changing the sharing rate in the range F_{sh} = (0, 0.1) on the normalized QBER values (ϵ_{est} and ϵ) for the same conditions of Fig. 6. In Figs. 6 and 7, a greater emphasis has been put on the cases where the number of sent photons is low because the statistical discrepancy can be more obvious in samples of low size. For the cases r = 500, 1000, 2500, 5000, 10000, and 100000 shown in Fig. 7(a)–(f) respectively, it is possible to note that there are still some values when $F_{sh} < 0.1$ that give acceptable QBER estimation levels. However, some numerical analysis is required to better understand this issue and thus deduce the suitable range of F_{sh} that can produce practically good estimation levels.

In this respect, the statistical measures of variance and standard deviation can be helpful. The standard deviation can be defined as a measure of the variation amount of a random variable about its mean (μ). The standard deviation (σ) is the square root of its variance and can be expressed as follows:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \qquad (9)$$

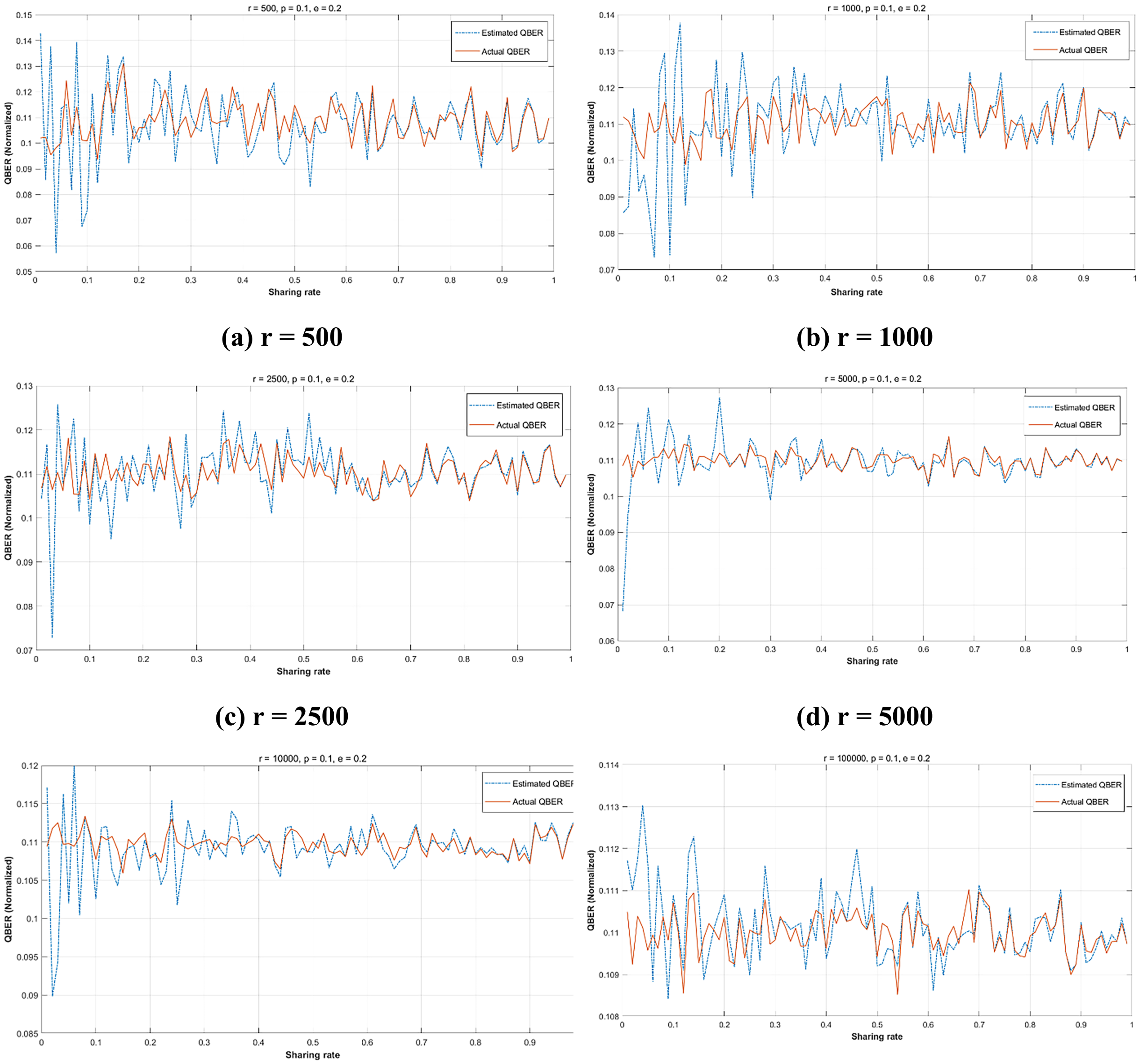
where $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$ is the mean of the N data samples (x_i).

Hence, the mean, variance, and standard deviation of the (normalized) estimated QBER values have been calculated for p = 0.1, e = 0.2, N = 100, and sharing rate values in the range F_{sh} = (0, 0.1) as shown in Table 2. Different values of the number of the sent photons have been considered from r = 1000 to r = 200000. From this table, it is possible to see that the value of the standard deviation has decreased from the level of about 4% (keeping in mind that the estimated values of the QBER are normalized) for r = 1000 to about 0.2% for r = 200000. This supports the assumption that the higher the number of sent photons (and hence the higher sifted keys' sizes) the better the accuracy of QBER estimation. Thus, from the perspective of QBER estimation accuracy, longer batches of the photons sent on the quantum channel are recommended. However, how long would be these batches is an issue that can be affected by other system design parameters and operation conditions.

Furthermore, the values of the standard deviation of the (normalized) estimated QBER have been also visualized as shown in Fig. 8 for p = 0.1, e = 0.2, and F_{sh} = (0, 0.1). Then, the Matlab 'polyfit' function was used to calculate the best linear fit for the standard deviation data. The result is the following linear polynomial shown in Eq. (10):

$$y_{lin} = -1.84 \times 10^{-7}x + 0.0363 \qquad (10)$$

Quadratic, cubic, and higher order polynomial fitting have also been tried but without noticeable advantage. Hence, the 'lsqcurvefit' Matlab function has been used for fitting the exponential decay model to the QBER estimation data by adjusting the 'initialGuess' values for better fitting. The best exponential decay formula that has been obtained is



(a) r = 500

(b) r = 1000

(c) r = 2500

(d) r = 5000

(e) r = 10000

(f) r = 100000

Fig. 6. The effect of sharing rate $F_{sh} = (0, 1)$ on the normalized QBER value (actual and estimated) for eavesdropping level ($e = 0.2$), channel depolarization factor ($p = 0.1$), and different numbers of sent photons (r).

described in Eq. (11):

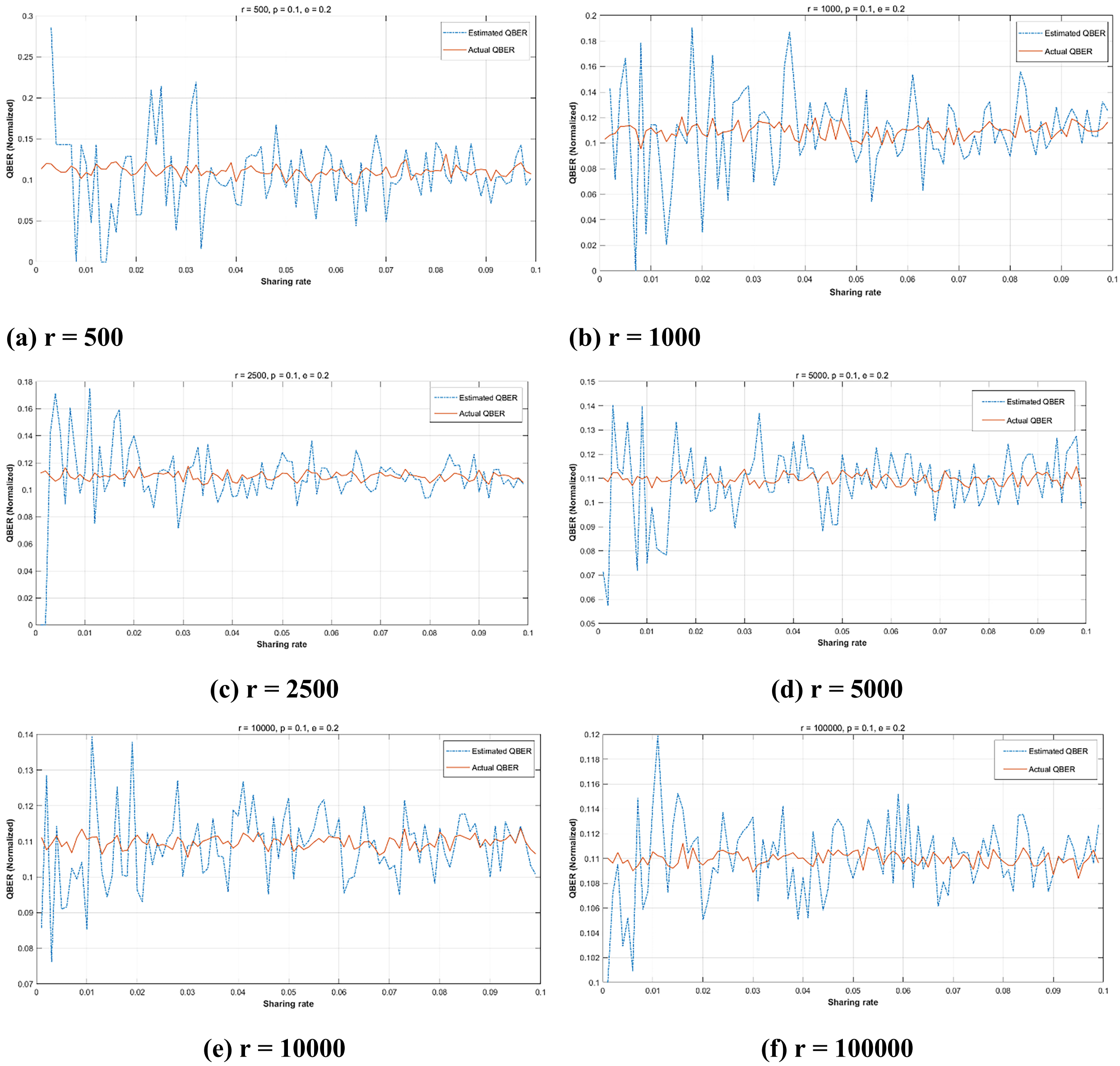
$$y_{exp} = 0.045 \times e^{-5.5\times10^{-5}x} \tag{11}$$

However, both fitting formulas in Eqs. (10) and (11) do not accurately describe the QBER estimation's standard deviation data. Therefore, a better fitting strategy that combines both linear fitting and exponential decay formulas has been concluded. The combined fitting formula is shown by Eq. (12):

$$y_{comb} = \frac{\frac{y_{lin}}{2} + y_{exp}}{2} = 9.075 \times 10^{-3} - 4.6 \times 10^{-8}x + 0.0225 \times e^{-5.5\times10^{-5}x} \tag{12}$$

All the fitting formulas represented by Eqs. (10) to (12) are illustrated in Fig. 8, where it is clear that Eq. (12) gives the best fitting among them.

Based on the above analysis, it is prudent now to propose an empirical formula for calculating the sharing rate value for various sifted key lengths (Since the sifted key length statistically equals 50% of the number of sent photons in our simulation setting). Thus, the following empirical formula described in



(a) r = 500

(b) r = 1000

(c) r = 2500

(d) r = 5000

(e) r = 10000

(f) r = 100000

Fig. 7. The effect of sharing rate F_{sh} = (0, 0.1) on the normalized QBER value (actual and estimated) for eavesdropping level (e = 0.2), channel depolarization factor (p = 0.1), and different numbers of sent photons (r).

Eq. (13) is proposed:

$$F_{sh} = \begin{cases} 0.09 & for\ n < 100 \\ -0.000065\ n + 0.096 & for\ 100 \le n < 1000 \\ 9.075\times10^{-3} - 4.6\times10^{-8}n \\ \quad +0.0225 \times e^{-5.5\times10^{-5}\ n} & for\ 1000 \le n < 50000 \\ 0.0052 + 0.045 \times e^{-5.5\times10^{-5}n} & for\ n > 50000 \end{cases} \tag{13}$$

where n is the sifted key length as defined previously. Note that the second, third, and fourth terms in Eq. (13) are simple manipulations of Eqs. (10) to (12) respectively. These amendments have been made based on the behavior of the results illustrated in Figs. 6 and 7. In order to prevent the value of F_{sh} from falling to negative values or values very close to zero, Eq. (11) has been manipulated by adding a suitable constant term such that for very large sifted key sizes, the sharing rate value will approach 0.0052 approximately. For small sifted key sizes (n < 1000), a simple linear formula has been suggested to avoid the rapid degradation of exponential decay. However, for very small sifted key sizes of less than 100 bits (which is not recommended), a constant sharing rate value of 0.09 is suggested due to the rapid

Table 2. The mean, variance, and standard deviation of the (normalized) estimated QBER. (p = 0.1, e = 0.2, N = 100, and F_{sh} = (0, 0.1))

Number of sent photons (r)	Mean (μ)	Variance	Standard deviation (σ)
1000	0.105931	0.001654	0.040669
1500	0.109241	0.000909	0.030150
2000	0.110683	0.000896	0.029933
2500	0.109744	0.000850	0.029154
3000	0.111331	0.000655	0.025593
5000	0.111237	0.000329	0.018138
7500	0.108294	0.000235	0.015330
10000	0.109247	0.000132	0.011091
12500	0.110662	0.000092	0.009592
15000	0.110308	0.000076	0.008718
20000	0.110847	0.000064	0.008000
25000	0.110241	0.000061	0.007810
30000	0.110002	0.000046	0.006782
50000	0.110473	0.000030	0.005477
100000	0.109686	0.000023	0.004796
200000	0.109950	0.000004	0.002000

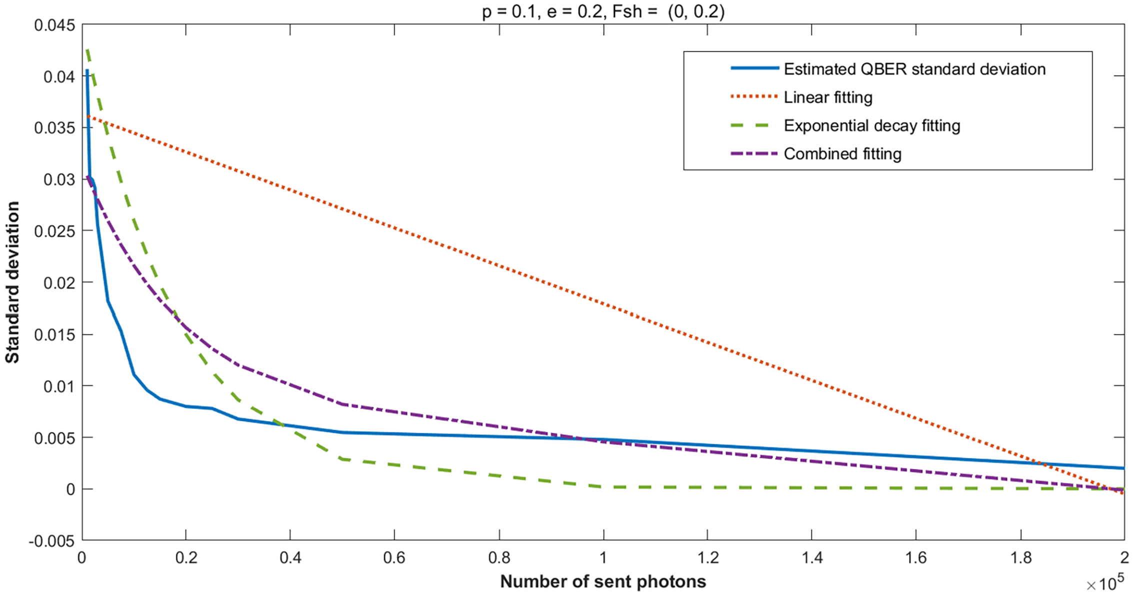


Fig. 8. Linear, exponential decay, and combined fitting of the standard deviation of the estimated QBER for different numbers of sent photons.

Table 3. The values of the sharing rate calculated based on the proposed empirical formula.

Sifted key size (bits)	Sharing rate	Sifted key size (bits)	Sharing rate
50	0.09	20000	0.0156
90	0.09	49990	0.0082
100	0.0895	50000	0.0081
999	0.031	100000	0.0054
1000	0.0303	200000	0.0052
10000	0.0216	300000	0.0052

fluctuations in estimating the QBER for a small number of sent photons. To better clarify the proposed empirical formula, the values of the sharing rate have been calculated for various sifted key lengths based on Eq. (13) as shown in Table 3.

Conclusion

In this work, the issue of the QBER estimation in QKD has been thoroughly studied. The study has emphasized the subtle consequences of the accuracy of QBER estimation on the efficiency and security of

QKD protocols in general. One of the main contributions of the work is the proposal of an empirical formula for calculating the suitable sharing rate for various sifted ley sizes, as shown in Eq. (13). based on this formula, it becomes straightforward to calculate the size of the permuted sifted key part required for QBER estimation based on Eq. (7). As far as the QBER estimation is concerned, the larger sifted key size is the better for estimation accuracy, and hence the resultant secret key rate. One possible future work direction is studying the QBER estimation for QKD protocols other than BB84 and/or other information reconciliation techniques based on LDPC codes. Indeed, one can also consider the extension of the EnQuad simulator for other security attacks rather than individual intercept/resend attacks.

Acknowledgment

The author wants to thank the anonymous reviewers for their helpful comments.

Authors' declaration

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Furthermore, any Figures and images, that are not ours, have been included with the necessary permission for republication, which is attached to the manuscript.
- No animal studies are present in the manuscript.
- No human studies are present in the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee at University of Anbar.

References

1. Rodimin VE, Kiktenko EO, Usova VV, Ponomarev MY, Kazieva TV, Miller AV, *et al.* Modular quantum key distribution setup for research and development applications. J Russ Laser Res. 2019 May;40(3):221–229. https://doi.org/10.1007/s10946-019-09793-5.
2. Kabanov IS, Yunusov RR, Kurochkin YV, Fedorov AK. Practical cryptographic strategies in the post-quantum era. In Int Conf Quant Tech- AIP Conf Proc. 2017;1936:020021-1–020021-5. https://doi.org/10.1063/1.5025459.
3. Faraj ST, Al-Naima F, Ameen S. Quantum cryptographic key distribution in multiple-access networks. In Proc 16th IFIP Int Conf Comm Tech. 2000 Aug;I:42–49. https://doi.org/10.1109/ICCT.2000.889166.
4. Kiktenko EO, Trushechkin AS, Kurochkin YV, Fedorov AK. Post-processing procedure for industrial quantum key distribution systems. arXiv. 2016 Sep [cited 2023 Nov 19]; 1603.08387v2. https://doi.org/10.1088/1742-6596/741/1/012081.
5. Kiktenko EO, Malyshev AO, Bozhedarov AA, Pozhar NO, Anufriev MN, Fedorov AK. Error estimation at the information reconciliation stage of quantum key distribution. J Russ Laser Res. 2018 Nov;39(6):558–567. https://doi.org/10.1007/s10946-018-9752-y.
6. Adu-Kyere A, Nigussie E, Isoaho J. Quantum key distribution: Modeling and simulation through BB84 protocol using Python3. Sensors. 2022;22:6284. https://doi.org/10.3390/s22166284.
7. Faraj ST. A novel extension of SSL/TLS based on quantum key distribution. In Proc Int Conf Comp Comm Eng (ICCCE08). 2008 May;I:919–922. https://doi.org/10.1109/ICCCE.2008.4580740.
8. Niemiec M, Pach AR. The measure of security in quantum cryptography. In IEEE Glob Comm Conf (GLOBECOM). 2012:967–972. https://doi.org/10.1109/GLOCOM.2012.6503238.
9. Schranz A, Udvary E. Quantum bit error rate analysis of the polarization based BB84 protocol in the presence of channel errors. In Proc 7th Int Conf Photon Opt Laser Tech (PHOTOPTICS). 2019:181–189. https://doi.org/10.5220/0007384101810189.
10. Gkouliaras K, Theos V, Evans P, Chatzidakis S. NuQKD: A modular quantum key distribution simulation framework for engineering applications. arXiv. 2023 Oct.; 2310.12351. https://doi.org/10.48550/arXiv.2310.12351.
11. Abdelgawad MS, Shenouda BA, Abdullatif SO. EnQuad: A publicly-available simulator for quantum key distribution protocols. Cyber Inform Tech. 2020 Mar;20(1):21–35. https://doi.org/10.2478/cait-2020-0002.
12. Mlejnek M, Kaliteevskiy NA, Nolan DA. Modeling high quantum bit rate QKD systems over optical fiber. In Proc SPIE Photon Eur 10674, Quant Tech. 2018 May;1067416. https://doi.org/10.48550/arXiv.1804.07722.
13. Mehic M, Fazio P, Rass S, Maurhart O, Peev M, Poppe A, *et al.* A novel approach to quality-of-service provisioning in trusted relay quantum key distribution networks. IEEE/ACM Trans Net. 2020 Feb;28(1):168–181. https://doi.org/10.1109/TNET.2019.2956079.
14. Huang D, Zhao Y, Yang T, Rahman S, Yu X, He X, *et al.* Quantum key distribution over double-layer quantum satellite networks. IEEE Access. 2020;8:16087–16098. https://doi.org/10.1109/ACCESS.2020.2966683.
15. Iwakoshi T. Bit-error-rate guarantee for quantum key distribution and its characteristics compared to leftover hash lemma. Proc Quant Tech Quant Infor Sci. 2018 Oct;IV(10803):1080309. http://dx.doi.org/10.1117/12.2500457.
16. Al-Janabi STF. Quantum key distribution networks. In multidisciplinary perspectives in cryptology and information security. Sadkhan S, Abbas N, editors, Switzerland: IGI Global; 2014;61–96. https://doi.org/10.4018/978-1-4666-5808-0.ch003.
17. Alhasnawy LH, AL-Mashanji AK. Improving wireless sensor network security using quantum key distribution. Baghdad Sci J. 2023 Oct. 28;20(5(Suppl.):2077–2085. https://doi.org/10.21123/bsj.2023.7460.
18. Kumar J, Saxena V. Cloud data security through bb84 protocol and genetic algorithm. Baghdad Sci. J. 2022 Dec. 5;19(6(Suppl.):1445–1453. https://doi.org/10.21123/bsj.2022.7281.
19. Mehic M, Rass S, Dervisevic E, Voznak M. Tackling denial of service attacks on key management in software-defined quantum key distribution networks. IEEE Access. 2022;10:110512–110520. https://doi.org/10.1109/ACCESS.2022.3214511.

20. Kong P-Y. A review of quantum key distribution protocols in the perspective of smart grid communication security. IEEE Sys J. 2022 Mar;16(1):41–54. https://doi.org/10.1109/JSYST.2020.3024956.
21. Elkouss D, Leverrier A, All´eaume R, Boutros JJ. Efficient reconciliation protocol for discrete-variable quantum key distribution. In IEEE Inter Symp Inform Theory. 2009 Jul;1879–1883. https://doi.org/10.1109/ISIT.2009.5205475.
22. Treeviriyanupab P, Phromsaard T, Zhang CM. Rate-adaptive reconciliation and its estimator for quantum bit error rate. In Inter Symp Comm Inform Tech (ISCIT). 2014;351–355. http://dx.doi.org/10.1109/ISCIT.2014.7011930.
23. Tang BY, Liu B, Yu WR, Wu CQ. Shannon-limit approached information reconciliation for quantum key distribution. Quant Inform Process. 2021;20(113):1–16. https://doi.org/10.1007/s11128-020-02919-8.
24. Zhou H, Tang BY, Chen H, Yu HC, Li SC, Yu WR, *et al.* Appending information reconciliation for quantum key distribution. arXiv. 2022 Apr. 2204;06971v1. https://doi.org/10.48550/arXiv.2204.06971.
25. Lee C, Sohn I, Lee W. Eavesdropping detection in BB84 quantum key distribution protocols. IEEE Trans Net Serv Manag. 2022 Sep;19(3):2689–2701. https://doi.org/10.1109/TNSM.2022.3165202.
26. Niemiec M, Romański L, Świę̧ty M. Quantum cryptography protocol simulator. In Inter Conf Multimedia Comm Serv Security. Dziech A, Czyżewski A, editors. MCSS CCIS 149. Germany: Springer; 2011:286–292. http://dx.doi.org/10.1007/978-3-642-21512-4_34.
27. Khodr M. Evaluations of quantum bit error rate using the three stage multiphoton protocol. CECTA. 2017;1–4. http://dx.doi.org/10.1109/ICECTA.2017.8251929.
28. Lee A, Castillo AT, Whitehill C, Donaldson R. Quantum bit error rate timing jitter dependency on multi-mode fibers. Opt Express. 2023;31(4):6076–6087. https://doi.org/10.1364/OE.477156.
29. Kebapci B, Levent VE, Ergin S, Mutlu G, Baglica I, Tosun A, *et al.* FPGA-based implementation of an underwater quantum key distribution system with BB84 protocol. IEEE Photon J. 2023 Aug;15(4):1–10. https://doi.org/10.1109/JPHOT.2023.3287493.
30. Murta G, Rozpedek F, Ribeiro J, Elkouss D, Wehner S. Key rates for quantum key distribution protocols with asymmetric noise. Phys Rev. 2020;A 101(062321). https://doi.org/10.1103/PhysRevA.101.062321.
31. Calver T, Grimaila M, Humphries J. An empirical analysis of the cascade error reconciliation protocol for quantum key distribution. In Proc 7th Ann Work Cyber Sec Inform Intell Research (CSIIRW '11), ACM. 2011;58(1). http://dx.doi.org/10.1145/2179298.2179363.
32. Pizzorno A. Quantum key distribution simulation on a distributed infrastructure. MSc [thesis]. Italy: Politecnico di Torino. 2022. http://webthesis.biblio.polito.it/id/eprint/22714.
33. Mehic M, Niemiec M, Siljak H, Voznak M. Error reconciliation in quantum key distribution protocols. In Lect Notes Comp Sci (LNCS) 12070. Ulidowski I, Lanese I, Schultz U, Ferreira C, editors. Germany: Springer; 2020:222–236. https://doi.org/10.1007/978-3-030-47361-7_11.
34. Mehic M, Niemiec M, Voznak M. Calculation of the key length for quantum key distribution. Elektron ir Elektrotech. 2015;21(6):81–85. http://dx.doi.org/10.5755/j01.eee.21.6.13768.

تأثير تقدير معدل الخطأ الكمي على كفاءة وأمان توزيع المفتاح الكمي

سفيان تايه فرج الجنابي

قسم أنظمة شبكات الحاسوب، كلية علوم الحاسوب وتكنولوجيا المعلومات، جامعة الأنبار، الرمادي، العراق.

الخلاصة

يعد توزيع المفتاح الكمي (QKD) حلاً مهماً لتحقيق نقل آمن للبيانات بين المستخدمين الشرعيين بغض النظر عن أي افتراضات حول القوة الحوسبية للمتنصت (مع الأخذ في الاعتبار صحة قوانين ميكانيكا الكم). ويظل هذا صحيحًا حتى في ظل الهجمات التي تشنها أجهزة الكمبيوتر الكمومية. وفي السنوات الأخيرة، كان هناك اهتمام بحثي كبير بتطوير إنشاءات أمنية معززة تعتمد على QKD. ويرجع ذلك أساسًا إلى التهديدات المتصورة لأجهزة الكمبيوتر الكمومية للبنية التحتية التقليدية لأمن المعلومات القائمة على تشفير المفتاح العام. ويقع هذا البحث تحت المظلة الواسعة لما يسمى بالتشفير ما بعد الكمي. لكن بالنسبة لأي بروتوكول لتوزيع المفتاح الكمي، يعد التقدير الدقيق لمعدل الخطأ الكمي (QBER) قبل مرحلة تسوية المعلومات أمرًا بالغ الأهمية لكل من منظوري الكفاءة والأمن. وفي هذا البحث، نستخدم تطويراً لمحاكي EnQuad لدراسة تأثير تقدير QBER على كفاءة وأمان بروتوكول BB84 QKD تحت ظروف إزالة استقطاب القنوات الكمومية وهجمات الاعتراض/إعادة الإرسال الفردية.. لقد تم فحص العوامل المختلفة التي تؤثر على عملية تقدير الخطأ الكمي بعناية. وأظهر تقييمنا الأهمية الكبيرة لخطوة تقدير الخطأ الكمي هذه. كما تم أيضا استخلاص صيغة تجريبية لحساب الحجم المطلوب للمجموعة الجزئية من المفتاح المنقى اللازمة لتقدير قيمة الخطأ الكمي، حيث يعد ذلك اسهاما مهما لهذا العمل.

الكلمات المفتاحية: تسوية المفاتيح، الخطأ الكمي، التشفير الكمي، توزيع المفتاح الكمي، الأمان غير المشروط.