cryptography

MDPI

Review

Variations of QKD Protocols Based on Conventional System Measurements: A Literature Review

Abdulbast A. Abushgra

Cybersecurity Department, Utica University, 1600 Burrstone Road, Utica, NY 13502, USA; aaabushg@utica.edu; Tel.: +1-315-223-2303

Abstract: Cryptography is an unexpected revolution in information security in the recent decades, where remarkable improvements have been created to provide confidentiality and integrity. Quantum cryptography is one such improvement that has grown rapidly since the first announced protocol. Quantum cryptography contains substantial elements that must be addressed to ensure secure communication between legitimate parties. Quantum key distribution (QKD), a technique for creating a secret key, is one of the most interesting areas in quantum cryptography. This paper reviews some well-known quantum key distribution techniques that have been demonstrated in the past three decades. Furthermore, this paper discusses the process of creating a secret key using quantum mechanics and cryptography methods. Moreover, it explains the relationships between many basic aspects of QKD protocols and suggests some improvements in the cryptosystem. An accurate quantitative comparison between the QKD protocols is presented, especially the runtime execution for each QKD protocol. In addition, the paper will demonstrate a general model of each considered QKD protocol based on security principles.

Keywords: quantum key distribution protocol; district variable; continues variable; superposition state; quantum bit (qubit); entanglement state

Citation: Abushgra, A.A. Variations of QKD Protocols Based on Conventional System Measurements: A Literature Review. *Cryptography* **2022**, *6*, 12. https://doi.org/10.3390/cryptography6010012

Academic Editor: Josef Pieprzyk

Received: 21 January 2022
Accepted: 2 March 2022
Published: 4 March 2022

1. Introduction

For several years, cryptographers have experimented with encryption/decryption techniques to create effective and robust methods to secure communications between two or more legitimate entities. The first known method is conventional (classical) cryptography, which involves sharing a message between two legitimate parties through insecure channels (for example, the Internet). The shared message (plaintext) should be sent from Alice (the sender) to Bob (the receiver) without the intervention by Eve (an eavesdropper). To prevent impersonation and listening, both Alice and Bob are supposed to send an X encrypted plaintext. Alice encrypts the plaintext X to become ciphertext C. On the other hand, Bob decrypts the ciphertext C to reveal the original plaintext X. The two legitimate participants use a shared code (secret key) to produce a ciphertext C [1]. For a long time, various algorithms have been used in the cryptosystem, and Caesar Cipher is one of them. Later, several cryptographic algorithms were introduced based on the large amount of data used by the Internet. RSA [2] and Diffie-Hellman [3] are the two most popular classical algorithms used in today's cryptographic systems.

Due to the huge amount of data that is shared over the Internet, classical cryptography may not provide sufficient protection. Hence, scientists are seeking a more reliable system based on the law of physics. A quantum system is the next solution for cryptography, where quantum cryptography is empowered by quantum mechanics. The power of quantum cryptography was initially introduced by Wiesner and Bennett in 1979 [4,5]. Quantum cryptography applies theories of physics to produce a secret key that can be shared by communicating parties. Moreover, sharing a secret key usually contains a random string of qubits (quantum bits) between the two entities. The encryption/decryption of plaintext X

are processed using a quantum system [6]. This quantum system operates based on the principles of physics and generates a secret key that is shared between communicating parties. These parties can communicate over insecure (public) channels to confirm the uncovered qubits (sifting phase). Therefore, Quantum Key Distribution (QKD) provides a secret key generator that can be guaranteed by the law of physics. Generally, QKD protocol consists of two approaches to deal with a string of qubits: the first approach is a Discrete Variable (DV), which is coded in the quantum state of a single photon, and the binary data should be measured using a single photon detector, while the second approach contains a Continuous Variable (CV), which is encoded on coherent states of weak pulses of light. Continuous data values are measured using homodyne detection methods [7], where CV systems should offer advantages over traditional DV systems. These advantages could be reflected in a higher secret key exchange rate for short distances, lower cost, or compatibility with telecommunication technologies.

Although quantum computers are light years away from today's technology, current cryptologists are exploring the impacts of this future technology, especially by breaking the current cryptosystem, which is based on prime factors of large numbers, such as ECC and RSA. Moreover, these impacts are a threat to cryptography elements such as confidentiality, data integrity, and authentication (digital signature). Consequently, breaking public-key cryptography requires a quantum computer that is supposed to be at least ($\approx$ 2000 qubits). Therefore, this quantum computer would break a prime factor of a great number in milliseconds. Usually, breaking an RSA encryption using an existing classical system (even power computers) would take hundreds of years [8].

This paper highlights the mechanisms and processes that are used to produce a secret key in quantum system using QKD protocols such as the BB84 protocol, SARG04 protocol, B92 protocol, Coherent-One-Way (COW) protocol, KMB09 protocol, EPR protocol, S09 protocol, Differential-Phase-Shift (DPS) protocol, and S13 protocol. Although many interesting QKD protocols have been announced in the quantum cryptography world, these QKD protocols determine the most applicable and well-known protocols.

2. Literature Review

In 1984, Bennett and Brassard invented the first quantum key distribution protocol, which became the first step toward quantum cryptography [9,10]. Many recent QKD protocols have been designed based on Bennett and Brassard's algorithms, particularly the quantum channel. A quantum key distribution protocol has been adopted to generate a shared key using quantum mechanics. This shared key is supposed to be confirmed in sifting and correcting error phases, where both phases should be applied during the classical communication channel. Each QKD protocol has a certain design that makes the generation of a secret key either secure or weak. These designs will be explained later based on the original protocols.

Section 2.1 presents the history of classical cryptography, and Section 2.2 explains quantum cryptography as well as the known QKD protocols. Finally, the paper presents outcomes of the QKD protocol comparisons and experiments that mainly focus on runtime execution during secret key exchange.

2.1. Classical Cryptography

Classical (conventional) cryptography relies on the complexity and difficulty of computing mathematical equations. This complexity will help Alice and Bob hamper Eve by exposing submitted messages or taping some of the contents. However, the reality of classical cryptography depends on the ability to stop threats caused by two types of cyber-attacks. One attack is called an active attack, and the other is called a passive attack. Active attacks can involve changes in the submitted data, unlike passive attacks that tap data without any changes. Historically, classical cryptography was explored in 1900 B.C., after the discovery of some messages that had been written in an ancient Egyptian tomb [11]. Subsequently,

several cryptographic algorithms [1] were introduced, and all of these algorithms have the same goal, which is inferred by creating a secret shared key as follows:

$$C = E\{K, X\}, \text{ and then } X = D\{K, C\}.$$

In addition, many encryption/decryption algorithms have been developed in classical cryptography, such as the RSA algorithm. The RSA algorithm [2] uses a cryptography method that is based on the complexity of computing prime factors of large integers. Moreover, there are many cryptographic algorithms [12,13] that possess stability and applicability in today's classical system. However, these algorithms will most likely fail once a quantum computer exists. The main reason behind the failure of the classical system is due to the power of the quantum system, especially the processing speed of the quantum machine. Because of the high degree of parallelism in quantum machines introduced by superposition states, the time required for factoring a large prime number is small compared to a classical machine.

Generally, classical cryptography is considered secure, as long as the quantum computer is not publicly available. The weakness of classical cryptography is related to the time needed to encrypt/decrypt algorithms, which can be a cryptanalysis solution for any cryptographic algorithm. However, the mechanism of encrypting/decrypting any kind of plaintexts in the classical system is applicable and simple operations, but it contains difficult mathematical equations that are impenetrable for any attack as shown in Figure 1. The main purpose of a cryptographic system is to send a message from Alice, who converts the plaintext X to ciphertext C using one of the available algorithms (e.g., DES, 3DES, AES, Diffie-Hellman, RSA, etc.), and then the ciphertext is sent through one of the classical communication channels (e.g., the Internet) to Bob. Hence, Bob converts the ciphertext C to plaintext X by decrypting the ciphertext using only a shared key (secret key) in symmetric cryptography methods. In asymmetric cryptography methods, Alice and Bob use the same previous scenario except that two shared keys (public and private keys) are required between the communicating parties [14].

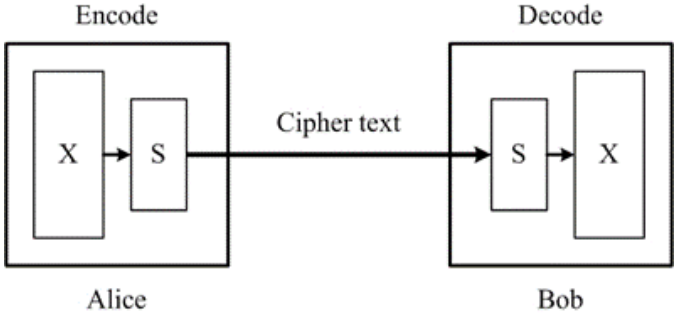

Figure 1. A simplified model of a symmetric encryption in the classical system, where X is a plaintext, S is an algorithm, and Ciphertext is the encoded message.

2.2. Quantum Cryptography

Presently, quantum cryptography is a sparkling topic in the field of communications and information technology. There is no doubt that the main focus of the cryptosystem is to prevent any entity from accessing the shared data, except for legal correspondents. Confidential communications between the sender and the receiver message must be ensured, and the principles of quantum communication ensure that data are transmitted over a secure channel with notification in the event of a data interruption.

Quantum cryptography uses symmetric key encryption, which is very common in classical cryptography methods. Moreover, symmetric key encryption provides straightforwardness between communicating parties in a quantum system, such as the One-Time Pad (OTP) encryption technique [15]. Furthermore, the Shared Secret Key (SSK) is used in the OTP mechanism by converting the entire plaintext X into a long string of n-bits.

On the other hand, the SSK should be created to match the same length (number of bits) of the original plaintext X, where the plaintext X and secret key SSK will be XORed ($\otimes$) to produce a ciphertext C. This process can be used in quantum cryptography when both legitimate parties obtain an SSK with the known plaintext X. An interesting point in quantum cryptography is the inability to make a copy of the original plaintext X or even listen to the message content by an eavesdropper, due to the rules of quantum mechanics (non-cloning theory) [16]. For instance, if two connected terminals are interrupted during a quantum communication, then the quantum system will be altered. This alteration is recognized by destroying the content of the message. Therefore, communicating parties can detect the information attacks, while the eavesdropper cannot take advantage of a system interruption.

Furthermore, several requirements are available to achieve secure communication using quantum systems. The first condition is a quantum channel, where the submitted data (set of quantum bits) is transmitted. The quantum channel should be either free space or fiber optics [17]. The data submitted in the quantum channel includes information about the shared secret key (SSK), which is carried by elementary particles. The second condition consists of a classical channel that should be established to recognize whether the shared key was detected by an eavesdropper or altered by the environment [18]. The classical channel uses a sifting process to correct errors that occur during the transmission of quantum channel. In addition, it is used to terminate the communication initiated between the parties if the detected error rate is high.

Public key cryptography provides only a certain amount of protection. Therefore, quantum mechanics will provide a complete solution for the next generation of secure communication networks. Quantum cryptography is based on quantum mechanics, in which some of theories of physics are applied. Moreover, there are many physical quantities or observables, such as photon polarization, momentum, and mass that can be used in the field of cryptography [19]. Based on the law of physics, the process of exchanging information within a quantum system is naturally protected from passive attacks, but it is still a challenge for active attacks. Here, the information used is initiated as a string of bits that are converted by quantum devices into quantum bits (qubits). The qubits are directed by light filters to different polarized states $|\varphi_i\rangle$. Therefore, a single photon can be initiated and measured based on multiple states. The explanations for used symbols are shown in Table 1.

Table 1. Units for quantum cryptography.

Symbol	Description
Φ	Quantum superposition of n states.
Ψ	Quantum superposition of n states.
$\otimes$	Exclusive OR (digital logical gate).
A	Alice, and usually is the sender.
B	Bob, and usually is the receiver.
$\uparrow$	A state with a definite value of spin operator.
X	The original message that should be shared between Alice and Bob.
$\|v\rangle$	Ket-notation, where it is a vector v.
$\langle v\|$	Bra-notation, where it is a linear form.
+	Non-orthogonal States.
$\times$	Orthogonal States.
OTP	One-Time Pad.
Qubit	Quantum Bit.
QBER	Quantum Bit Error Rate.

Then, this paper introduces some well-known QKD protocols and explains each QKD protocol based on the mechanism of the adopted algorithm. In addition, the QKD protocols were tested and analyzed to demonstrate the differences between all selected QKD protocols.

2.2.1. The BBB84 Protocol

In 1984, the BB84 protocol was introduced by Bennett and Brassard [20]. The concept of the BB84 protocol depends on the exchange of a secret key between Alice and Bob through a secure quantum channel. The process is described as a tossing-coin, whereby two communication channels (quantum and classical channels) are initiated between Alice and Bob. The quantum channel is technically the emission of a photon in either a free space or fiber-optic cable. The classical channel is an ordinary, traditional bit-shift channel, in which communications during the classical channel do not need to be secure. Both the sender and receiver should have a random number generator and four polarizing filters to generate qubits. These requirements should be available to fulfill the quality of photon submissions [21,22].

Moreover, generating a secret key using the BB84 protocol requires each of the communicating parties (the sender and receiver) to have a random number generator that should be placed in an appropriate position. The generator can be set in the middle between the legitimate parties. Primarily, the sender (Alice) starts preparing plaintext X, which is converted to a string of bits. Simultaneously, Alice initiates a random set of bases (recliner or diagonal) that matches the length of plaintext X. These bases include four states ($|+\rangle$, $|-\rangle$, $|0\rangle$, and $|1\rangle$). Each state on a different basis reflects the probability of (0 or 1). Furthermore, the entire prepared states $|\varphi_i\rangle$ will be submitted to a quantum channel with the same polarization of the prepared state as long as there is no interruption [23].

The probability of X encoded during Alice's setup represents the randomness of an encryption algorithm, where encoding the same information of X many times produces various ciphertexts [24]. Although many schemes have been published illustrating the inefficiency of the BB84 protocol as well as the weaknesses in the encryption mechanism, the BB84 protocol remains a solid background for many modern QKD protocols. The BB84 protocol is also considered to demonstrate the relationship between simplicity and durability, as shown in Figure 2.

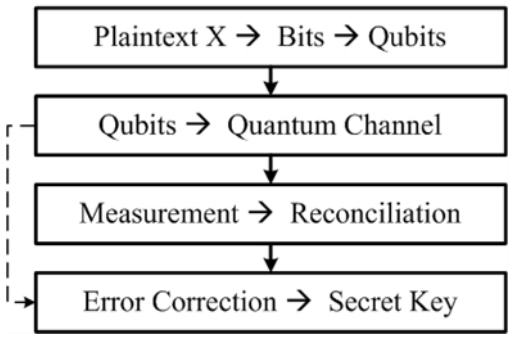

Figure 2. The main outline of secret key's process in quantum key distribution protocols, where the sender initiates the communication from up with converting the bits to quantum bit. Then processing the submissions between the sender and receiver through quantum channels, and finally the confirmation and error-correction will be using a classical system.

In addition, the BB84 protocol relies on the non-cloning theorem and the Heisenberg uncertainty principle to secure the submitted qubits. The non-cloning theorem is derived from the superposition principles of quantum mechanics [25]. Moreover, the non-cloning feature makes the BB84 protocol more stable by detecting any attack, although attackers never stop attempting to crack any cryptographic protocol. The Heisenberg uncertainty principle is described as the impossibility to prepare or measure states simultaneously in a specific environment based on position and momentum with quantum conditions.

In general, quantum key distribution protocols can be categorized by two disciplines of the photon behavior: the first one is based on superposition states (orthogonal/non-orthogonal) and the second one is based on the entangled states, where the BB84 protocol uses polarized orthogonal states [26]. In superposition states, Alice sends a state that should

be generated on bases of $(\times)$ or $(+)$ as above, where in this case, Bob should work on one of these bases randomly. Furthermore, if Alice uses the $(\times)$ basis to submit a $|1\rangle$ state, she will send a $|\nwarrow\rangle$ state. Following the same, if she wants to send a $|\uparrow\rangle$ state, and Bob already measured the $|\uparrow\rangle$ state in the $(+)$ basis, he will record a $|1\rangle$ state. Additionally, if Alice sends a photon as $|\nearrow\rangle$ or $|\nwarrow\rangle$ state and Bob just measures the photon in the basis $(+)$, the measurement will be in the polarized states in Equation (1) as follows:

$$\begin{aligned} |\nearrow\rangle &= \tfrac{1}{\sqrt{2}}(|\uparrow\rangle + |\rightarrow\rangle), \\ |\nwarrow\rangle &= \tfrac{1}{\sqrt{2}}(|\uparrow\rangle + |\rightarrow\rangle). \end{aligned} \tag{1}$$

Therefore, there is a 50% chance of recording $|0\rangle$ or $|1\rangle$ state by Bob as well as four possibilities [9] in Equation (2) as follows:

$$\begin{aligned} |\nwarrow\rangle \; with \; (+) &= \tfrac{1}{\sqrt{2}}(|\uparrow\rangle - |\rightarrow\rangle), \\ |\nearrow\rangle \; with \; (+) &= \tfrac{1}{\sqrt{2}}(|\uparrow\rangle + |\rightarrow\rangle), \\ |\uparrow\rangle \; with \; (\times) &= \tfrac{1}{\sqrt{2}}(|\nearrow\rangle + |\nwarrow\rangle), \\ |\rightarrow\rangle \; with \; (\times) &= \tfrac{1}{\sqrt{2}}(|\nearrow\rangle - |\nwarrow\rangle). \end{aligned} \tag{2}$$

These possibilities are shown on the Bloch sphere to display the measure of each polarization state that can be displayed in a three-dimensional space (x, y, and z) as shown in Figure 3.

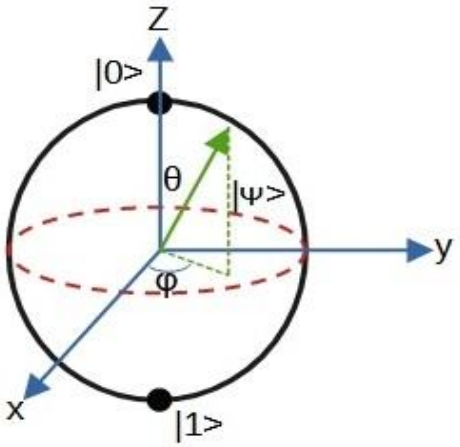

Figure 3. The Bloch Sphere.

In fact, there are many sequential steps for both parties to create a successful Shared Secret Key (SSK) using the BB84 protocol, as shown in Figure 2. These steps are described as follows:

Step 1: The length of the plaintext X should be set up by Alice to become a string of n-bits, and then the n-bits are applied to a randomly prepared basis ($\times$ or$+$).

Step 2: Each random basis will produce a random state either $|0\rangle$ or $|1\rangle$ if the basis is $|\times\rangle$, or $|0\rangle$ or $|1\rangle$ if the basis is $|+\rangle$ as shown in Table 2.

Table 2. The Alice sends n random bits in random bases.

Bit Number	0	1	2	3	4	5
Alice's random bits	0	1	1	0	1	1
Alice's random bases	+	+	×	+	+	+
Alice sends	→	↑	↖	→	↑	↑

Step 3: When the string of n-qubits is submitted by Alice, Bob measures the upcoming n-qubits based on random bases, as shown in Table 3. Next, Bob obtains a string of states that reflect n-bits. If Bob cannot measure all the submitted qubits, both parties will release additional qubits by sharing the used bases through a public channel.

Table 3. Bob receives *n* random bits in random measurements.

Bit Number	0	1	2	3	4	5
Bob's random bases	×	+	×	×	+	×
Bob observes	↗	↑	↖	↖	↑	↗
Bob's bits	0	1	1	1	1	0

Step 4: Both Alice and Bob start estimating the errors that could be caused by Eve, where many error-correction methods [27] are used in the BB84 protocol. The raw secret key is processed, when Alice and Bob compare the matching bits, where the uncorrelated bits are discarded, as shown in Table 4. This is called a sifting procedure, which enhances any attempt by Eve to obtain information and detect any error [28].

Table 4. Alice and bob publicly compare used bases.

Bit Number	0	1	2	3	4	5
Alice's random bases	+	+	×	+	+	+
Bob's random bases	×	+	×	×	+	×
Agreement		✓	✓		✓	
Shared secret key		1	1		1	

Step 5: After matching the sent and received qubits, the communication moves to the reconciliation phase only if the error rate is low. On the other hand, Alice and Bob end up the current communication if the error rate is too high.

Step 6: If the error rate is low, Alice and Bob share the raw key. Moreover, the raw key contains the matched qubits of both parties. Unmatched qubits are supposed to be removed from the shared key SSK.

Step 7: Next, Alice and Bob start correcting the erroneous qubits again in a separate phase, as shown in Table 5, as they endeavor to reduce the number of exposed qubits.

Table 5. Alice and Bob publicly compare half of the remaining bits.

Bit Number	0	1	2	3	4	5
Shared secret keys		1	1		1	
Randomly chosen			✓		✓	
Shared secret key		1	1		1	
Agreements			✓		✓	
Unrevealed secret keys		1				

Step 8: After checking for errors, Alice and Bob share an SSK that has the same length of plaintext X [29,30]. In other words, Alice could cheat in this position by sending a different basis (rectilinear and diagonal basis or neither rectilinear nor diagonal photon), so that she is not in a position to agree with any of Bob's table records in step (3). In contrast, Bob's table records the result of probabilistic behavior that is not under the control of the matching raw key [31].

Hence, it is very important to realize that if Alice tries to cheat in step (1), for instance, by sending a mixture of rectilinear and diagonal states, Alice will lose the ability to agree with Bob's records table after step (1).

Finally, the BB84 protocol is assigned a secure protocol as mentioned in [32], and it is a simple protocol compared with current QKD protocols. This simplicity is based on the law of physics that occur during key generation.

2.2.2. The SARG04 Protocol

In 2004, SARG04 was introduced by Scarain, Acin, Ribordy, and Gisin [33]. This protocol was then extracted using the previous protocol BB84. The SARG04 protocol uses the same bases and states as the BB84 protocol, where two bases ($\times$ or $+$) and four states ($|+\rangle$, $|-\rangle$, $|0\rangle$, $|1\rangle$) are used to initiate quantum submissions between the communicating parties. The SARG04 protocol is designed to be a robust protocol against Photon-Number-Splitting (PNS) attacks, especially when weakened laser pulses are emitted instead of a single photon source. Furthermore, SARG04 and BB84 are essentially equivalent to each other in the quantum communication phase, but the variation occurs by encoding and decoding the exchanged information into the classical channel [34].

The SARG04 protocol has a certain number of instructive differences, of which Bob must always choose the bases with a probability of $\frac{1}{2}$, even when Alice uses the same bases [35,36], [37] (p. 4). Although the SARG04 protocol is considered a new quantum mechanism for creating a secure shared key, the BB84 protocol is still seen in the instructions of the SARG04 protocol. In other words, when Alice matches the initiated qubits with the equivalent qubits from Bob, the Quantum Bit Error Rate (QBER) increases based on the presence of the error (unlike BB84, which is satisfied by the sifting phase).

To abstract the sequential steps of the SARG04 protocol between the two legitimate parties Alice and Bob, one-way communication was applied as follows:

Step 1: Alice creates n photons that start randomly with each of the four states ($|\varphi_i\rangle$, $i = 0, 1, 2, 3.$); Bob should receive one of the four states.

Step 2: When the photon is sent to Bob, it is measured randomly into quantum detectors using two bases ($\times$or $+$). If this measurement does not match or cannot be measured, Bob informs Alice to ignore this photon.

Step 3: Alice informs Bob about the states of photons $|\varphi_i\rangle$ that were chosen during the initiation period. Bob then matches outcomes using only two states. If the result was proven to be an orthogonal state to one of the set of states, the other states will already be proven. However, if the measured photons are not orthogonal, Bob should know that the measurements are not incisive. He then asks Alice to provide more specific details in the reconciliation phase.

Step 4: In the reconciliation phase, some qubits are chosen randomly to be tested and corrected by Alice, where Bob calculates the QBER. If the measurement of QBER was very high, Alice and Bob would agree to cancel the protocol and start another communication.

Step 5: In accordance with the previous step, both Alice and Bob retain only the conclusively matched qubits, which are used in a raw key. Unmatched qubits are treated during the qubit error-correction and privacy amplification phases [38–40].

SARG04 protocol can withstand PNS attacks. Although SARG04 appears as the BB84 protocol for all manipulations at the quantum level, it differs in the error-correction phase (sifting phase), where both parties communicate using a classical channel by encoding and decoding the shared information.

2.2.3. The B92 Protocol

B92 was proposed by Bennett in 1992 [41]. The protocol contains only two particle states, rather than four states in the BB84 protocol. The two states should be nonorthogonal, as illustrated in Figure 4. The process of the B92 protocol is involved in the quantum phase as follows:

Step 1: Alice sends a random string of qubits (A) to Bob; where $A \in \{0,1\}^n$, $n > N$ (which N is the length of final key), so if Alice sent the $|0\rangle$ state that means $A_i = 0$, and $A_i = 1$ if she sent $|+\rangle$ state, for all $i \in \{0, 1, \ldots n\}$.

Step 2: On the other hand, Bob creates a vector of bits (B) where $B \in \{0,1\}^n$, $n > N$, which if $(B_i = 0)$; then Bob will choose the basis $(+)$, and if $(B_i = 1)$ he will choose the basis $(\times)$ for all $i \in \{0,1 \ldots n\}$.

Step 3: When Bob starts measuring the upcoming qubits, each qubit is measured on a selected $(+)$ or $(\times)$ basis.

Step 4: After measuring the vector of states, Bob starts completing the following rules: if the measurement of the qubit produces $|0\rangle$ or $|+\rangle$ then $T_i = 0$, and if it produces $|1\rangle$ or $|-\rangle$, $T_i = 1$ for all $i \in \{0,1 \ldots n\}$ [42].

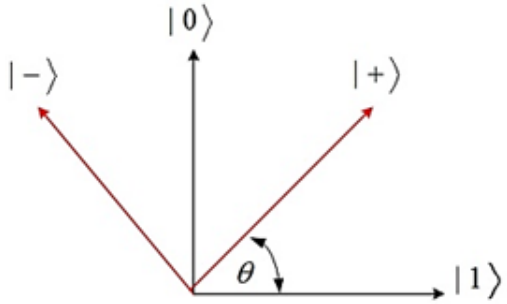

Figure 4. The non-orthogonal polarized states in the B92 protocol that represent the four states $|-\rangle, |+\rangle, |0\rangle$, and $|1\rangle$.

In general, the B92 protocol uses a non-orthogonal state to transmit information to a quantum channel. The protocol has a robust scheme with optical imperfection and detector noise, unlike the BB84 protocol. Naturally, the noise at the end of the communication can be as high as 1.6% [43]. Moreover, the B92 protocol technically has less usage of quantum memory (if any) and quantum channel capacity.

2.2.4. The Coherent One-Way Protocol

The Coherent One-Way (COW) is a simple protocol [44,45], which depends on decoding the information into time slots. Alice sends coherent pulses in logic states as [35] or decoy states. Each logical bit is encoded to either $(\mu - 0)$ for logical (0) or $(0 - \mu)$ for logical (1) by a sequence of two pulses. Furthermore, to improve the security of this protocol, Alice adds decoy sequences of $(\mu - \mu)$ while submitting the other logical states. If the pulses submitted to the interferometer are well aligned on Bob's side, then the received pulses will be perfectly detected on DM1 (interferometer) and there will be no detection on DM2 (detector). Therefore, the loss of coherence will be displayed on the detector when the eavesdropper tries to listen [46].

$$\begin{aligned} &\text{logic } 1: \ |0\rangle + |\mu\rangle \\ &\text{logic } 0: |\mu\rangle + |0\rangle \\ &\text{Decoy}: |\mu\rangle + |\mu\rangle, \end{aligned} \tag{3}$$

where μ is the mean photon number per pulse.

In this protocol, the transmission and reception of data depends on the time of arrival of the signal and does not depend on the polarization of the optical signals. The COW protocol works briefly as follows:

Step 1: Alice transmits a sequence of binary bits using time slots to Bob and generates both logical states of $|1\rangle$ or $|0\rangle$ (which has the same probability unless decoy states are added). Obtaining a probability of $\frac{1}{2}$ for each of $|1\rangle$ or $|0\rangle$ states and adding the decoy states are calculated by $(1 - f)/2$ (where f is the probability of decoy state generation).

Step 2: Bob exploits the time detection to generate a raw key, where all previous processes are performed by different detectors to improve the security rate in Equation (4).

$$V = \frac{p(D_{M1}) - p(D_{M2})}{p(D_{M1}) + p(D_{M2})} \tag{4}$$

where $p(D_{Mj})$ is the probability of the (D_{Mj}) clicks at the time when (D_{M1}) should click, as shown in Figure 5.

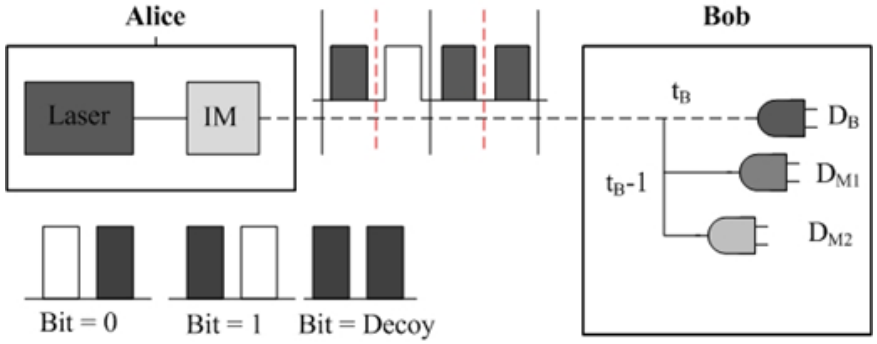

Figure 5. The Coherent-One-Way (COW) scheme between two legitimate parties (Alice and Bob), where Alice and Bob should have a particular equipment to process the submission and measure the split time. The submitted qubits include three categories of data, for instance the value of 0, the value of 1, and value of decoy states.

Step 3: Bob declares the number of bits by simultaneous procedures between the data detector and time detection on the side.

Step 4: On monitoring the detectors, Alice ensures that the sequence of decoy states and bit sequences still exists. If not, Eve has tapped the communication. In this case, Alice breaks the coherence into two pulses to detect an interrupted state.

Step 5: Alice informs Bob about the bits that have been removed from the raw key because those bits belong to the decoy state sequence.

Step 6: The secret key is extracted after dropping the decoy sequences from the raw key using a classical process, and the shared key is obtained by error-correction and privacy amplification [47].

This protocol, as reported in [48], is designed to be a robust quantum protocol against reduced interference visibility and PNS attacks. The COW protocol also has simple transmissions into data lines, low losses at the measurement side, and a small QBER detection.

2.2.5. The KMB09 Protocol

This protocol was presented in 2009 [49] by Khan, Murphy, and Beige, and is designed to be robust against PNS attacks. Khan et al. describe the protocol as being between two parties (Alice and Bob) and an eavesdropper (Eve). Both parties must use two bases e and f, where both parties should use different indices i whenever they use the same basis [50]. Moreover, the i index is publicly declared between two legitimate parties, which can be pointed to Alice's prepared indices as i. and Bob's measured indices as j.

In KMB09, the authors attempted to create a protocol that could withstand PNS attacks. In addition, KMB09 was created when other protocols were used for a few kilometers, where the system error rate could exceed the eavesdropper's presence. The protocol was optimized by using an Index Transmission Error Rate (ITER) instead of QBER during the reconciliation phase. The next steps briefly explain the KMB09 protocol as follows:

Step 1: Alice randomly generates a sequence of classical bits, and then randomly specifies an index $i = 1, 2 \ldots N$.

Step 2: Alice sends the prepared bits in a single photon into either $|e_i\rangle$ or $|f_i\rangle$ basis to Bob.

Step 3: Each incoming state is measured by Bob to be randomly switched between bases e and f.

Step 4: Alice announces in public communication to Bob about the random sequential indices i to obtain the secret key.

Step 5: Bob translates the measurement outcomes.

Step 6: Bob communicates with Alice publicly to share that the photon measurements were successfully received and obtained the secret key.

Step 7: Alice and Bob can determine whether Eve is eavesdropping on the communication as Equation (5) [51].

$$P_{ITER} = 1 - \frac{1}{2N} \sum_{i=1}^{N} \sum_{k=1}^{N} \left[|\langle g_k|e_i\rangle^4 + g_k|f_i\rangle^4| \right] \quad (5)$$

where e, f and g are bases, and the state of $|g_k\rangle$ is Eve's possible measurement outcomes, and it is forwarded to Bob without alteration.

The polarization of a single photon is initiated in multi-dimensional states, as shown in Figure 6, which is based on orthogonal or non-orthogonal bases [52].

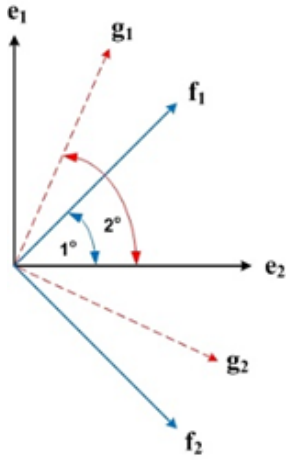

Figure 6. Two bases vector used by Alice, Bob, and Eve in the N = 2 protocol.

The KMB09 protocol is designed to be used under ideal conditions, where it is impossible for Alice and Bob to have different indices while using the same basis. This protocol is more robust against any eavesdropper who tries to hide his/her presence. In addition, the strong correlation between QBER and ITER makes the eavesdropper produce a distinct signature that is easy to detect.

2.2.6. The EPR Protocol

EPR Pair Paradox was inspired by Einstein, Podolsky, and Rosen, who presented a dialectical paper in 1935 [53]. The presented theory has led to an argument about quantum mechanics, which is not a completely physical theory. The main concept uses three states of polarization considering $|\theta\rangle$, where the polarization state of the photon is linearly polarized at angle θ. More precisely, the EPR is a pair of particles that can be separated even over a great distance, so that both photons show in a paradox "action at a distance" [54].

To explain the nature of the EPR pair paradox clearly, when one photon is measured on the right side, the outcome may be a vertical linear polarization $|0\rangle$ state. On the other hand, the measurement will be a $|1\rangle$ state on the left side, where the measured photons will be horizontally in a linear polarization state $|\pi/2\rangle$ (and vice versa). Therefore, the EPR is one of the four Bell states as Equation (6).

The EPR protocol was presented by Artur K. Ekert in 1991 [55], which is completely based on the use of an entanglement state between two remote parties. Moreover, few

modifications have been made since the first EPR protocol has become popular. Hwang et al. [56] explained some of these modifications to the EPR protocol. The EPR process is shown in steps that demonstrate the original protocol [9]:

Step 1: Alice creates a sequence of EPR photons (entangled qubits) n, where one photon is stored in a quantum memory and sends the other to Bob.

$$\begin{aligned} |\psi_1\rangle &= \tfrac{1}{\sqrt{2}}(|00\rangle+|11\rangle) \\ |\psi_2\rangle &= \tfrac{1}{\sqrt{2}}(|00\rangle-|11\rangle) \\ |\psi_3\rangle &= \tfrac{1}{\sqrt{2}}(|10\rangle+|01\rangle) \\ |\psi_4\rangle &= \tfrac{1}{\sqrt{2}}(|10\rangle-|01\rangle) \end{aligned} \tag{6}$$

Step 2: Both communicators randomly choose a sequence of bases ($\times$or $+$); these bases are used to measure the particles at each side of the communication, as shown in Table 6.

Table 6. The measurements in the EPR protocol Alice and bob measure in each of their random bases.

Bit Number	1	2	3	4	5	6
Alice's random bases	$\times$	$\times$	$+$	$+$	$\times$	$+$
Alice's observations	$\nearrow$	$\nwarrow$	$\rightarrow$	$\uparrow$	$\nearrow$	$\rightarrow$
Bob's random bases	$\times$	$+$	$+$	$\times$	$\times$	$+$
Bob's observations	$\nearrow$	$\rightarrow$	$\rightarrow$	$\nearrow$	$\nearrow$	$\rightarrow$

Step 3: In public, Alice and Bob match the outcomes of their measurements and keep only the qubits that were measured on the same basis, as in Table 7.

Table 7. The measurements in the EPR protocol Alice and bob publicly compare their bases.

Bit Number	1	2	3	4	5	6
Alice's random bases	$\times$	$\times$	$+$	$+$	$\times$	$+$
Public channel	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$	$\Updownarrow$
Bob's random bases	$\times$	$+$	$+$	$\times$	$\times$	$+$
Agree	✓		✓		✓	✓

The remaining of EPR protocol includes decisions made by communicating parties. The public channel will be the next choice to ignore any errors while exchanging qubits through the quantum channel. Therefore, classical communication is analogous to the reconciliation phase of the BB84 protocol.

2.2.7. The S09 Protocol

S09 protocol was presented by Esteban and Serna in 2012; this protocol has a different technique compared to the previous protocols. S09 relies on public-private key cryptography, and the main process of the S09 is based on exchanging a qubit multiple times to build a secret key between Alice and Bob. However, the S09 protocol transfers the qubit into any arbitrary state that is agreed on between Alice and Bob only through the quantum channel. The sequences of the S09 protocol are briefly explained as follows.

Step 1: Generate a bit i by Alice that would be in element of a secret base (B_k) to create a qubit $|\Psi, k\rangle$, which in turn is sent to Bob with a quantum channel.

Step 2: Bob applies (U_j) to qubit $|\Psi, k\rangle$ on the other side, which is only recognized by Bob. Thus, he can send the outcome of the qubit to Alice.

Step 3: When Alice receives the qubit, it is measured in the base (B_k) and includes bit j, where the qubit must be in a pure state (ρ) by the operator density [50]:

$$\rho = |\Psi,\ k\rangle\langle\Psi,\ k|, \tag{7}$$

where the interaction of the qubits (ρ) with the environment produces:

$$\rho' = \sum_{j} E_j \rho E_j^{\dagger}, \tag{8}$$

where E_j is an operator that acts in the space of a qubit. Subsequently, these operators convey the state of qubit $|\Psi, k\rangle$ in the overlap.

$$|\Psi,\ k\rangle \ \rightarrow\ E_i\,|\Psi,\ k\rangle \tag{9}$$

Step 4: After a complex operation, parity bits are appended by the operators ($\|$ or &&).

Step 5: The previous step is attached to the distribution of the sent addresses or hashed values.

In addition, with the approach of this protocol, Eve can obtain nothing from her eavesdropping, since B_k and U_j transformations can be changed as frequently as needed. On the other hand, the S09 protocol has a complex exchange process that makes operating the protocol inefficient.

2.2.8. The S13 Protocol

S13 is a quantum key distribution protocol developed by Serna in 2013 [57]. This protocol corresponds to the BB84 protocol in quantum procedures but differs in the classical channel. S13 was designed for implementation in current system devices, without the need for modifications.

Furthermore, S13 has the same quantum communication phase as BB84; however, this will be overlooked in this section because it is already explained in Section 2.2.1. The second phase of the S13 protocol is explained as follows:

- Quantum part
  - Raw key exchange: (as shown in the BB84 protocol).
  - Random seed: one of the communicating parties creates a random binary string $(x_1\ x_2\ \ldots\ x_N)$.
  - Missing key exchange:
    1. Alice makes a summation of the random binary string with the binary basis from the first part and obtains a binary basis $(t_1\ t_2\ \ldots\ t_N)$. Alice then randomly generates another string of binary $(j_1\ j_2\ \ldots\ j_N)$, where this is an exchanged key with Bob.
    2. Bob sums each of the sequences sent to him by Alice with the created binary string $(1 \oplus m_k) \oplus x_k$, where $(k = 1, 2 \ldots N)$. Thus, the sum becomes a binary string basis $(n_1\ n_2\ \ldots\ n_N)$. Next, Bob measures the received state $|\Psi_{t_k}\ j_k\rangle$, with the correspondence of the basis (B_{nk}) to generate $(b_1\ b_2\ \ldots\ b_N)$.
- Classical part

  Alice and Bob apply function (f) to different binary exchanges in a set of binary strings:

$$f(z,x,y) := \begin{cases} x, & z = 0 \\ y, & z = 1 \end{cases} \tag{10}$$

  1. Asymmetric cryptography:

     Step 1: Alice sums the binary string created by her in quantum part *i* with a random string of binary values that were created by missing the key exchange *j*.

$$i_k \oplus\ j_k, (\ k = 1, 2 \ldots N), \tag{11}$$

     where $(y_1\ y_2\ \ldots\ y_N)$ will be sent to Bob.

Step 2: To obtain the public key, Bob encrypts:

$$\begin{aligned} u_k &= n_k \oplus f(m_k,\ a_k, b_k \oplus y_k), \\ v_k &= n_k \oplus f(m_k, b_k,\ a_k \oplus y_k). \end{aligned} \tag{12}$$

Step 3: Alice makes a summation to obtain the private string of m_k, which is:

$$t_k \oplus f(s_k,\ (1 \oplus i_k) \oplus u_k,\ j_k \oplus v_k), \tag{13}$$

and then decrypts the string $(m_1\ m_2\ \ldots\ m_N)$.

2. Private Reconciliation:
Step 4: Bob receives the binary sequence $(l_1\ l_2\ \ldots\ l_N)$ after completing the comparison between $(s_1\ s_2\ \ldots\ s_N)$ and $(m_1\ m_2\ \ldots\ m_N)$ by Alice.
Step 5: Bob sums the sequence of bases m_k with l_k, where $(m_k \oplus l_k)$, $k = 1,\ 2\ \ldots\ N$. This is to obtain the private string s_k.

$$\begin{aligned} f(l_k,\ a_k, b_k \oplus\ y_k) &\equiv i_k \\ f(l_k, a_k \oplus y_k, b_k) &\equiv\ j_k, \\ (k = 1,\ 2\ \ldots\ N). & \end{aligned} \tag{14}$$

Bob then obtains the private string from Alice $(i_1\ i_2\ \ldots\ i_N)$.

Finally, the S13 protocol is designed to be functional with existing devices, especially in the exchange phase after a qubit transmission. Several exchanges in the public channel will lead to a waste of time, as well as a chance for an eavesdropper to tap data. Furthermore, S13 is an improvement of the S09 protocol, which was ranked as a complex QKD protocol.

2.2.9. The Differential-Phase-Shift Protocol

The Differential-Phase-Shift (DPS) protocol was developed in 2002 by Inoue et al. [48]. The DPS protocol is based on four fully non-orthogonal states, in which Alice's photon splits into three pulses and it is randomly modulated. On the other hand, Bob measures the incoming photons from Alice with a differential phase measurement. As mentioned in [58], the DPS protocol is more suitable for fiber-optic transmission and provides a higher effective shared key than the BB84 protocol. Additionally, the DPS protocol has specific advantageous features that are included in a simple configuration, accurate time usage, and robustness again PNS attacks [59].

Technically, the DPS is used to create a secret key between two parties, and it starts at Alice's side when the single photon is divided into three paths (a, b, and c) and then recombined them using a beam splitter (BS) or optical switcher (SW), as shown in Figure 7. Moreover, the time delay (between a, b and b, c) is equal, so that the recombined photon is converted to each of $(0 \,||\, \pi)$. The incoming photons from Alice to Bob are divided into two paths and recombined using (50:50) beam splitters. The entire expected scenario of the DPS protocol is performed in the following sequential steps.

Step 1: At Alice's side, a photon is sent from (a) to the short path on Bob's side.

Step 2: Another photon is pushed through (a) to the long path on Bob's side and through (b) to the short path.

Step 3: A photon is pushed through (b) to the long path on Bob's side, and (c) to the short path.

Step 4: Another photon is pushed through (c) to the long path on Bob's side.

In the first part of processing DPS, two probabilities overlap in steps (2) and (3), where the phase difference is $(0\ or\ \pm\pi)$ which depends on Alice's modulation. Moreover, each detector clicks on (0) and the other clicks on $(\pm\pi)$ phase difference. Finally, when Bob's detectors click, Bob records the time and knows which detector clicks. During the classical two-way communication, Alice knows which one clicks at Bob's detector [48,58].

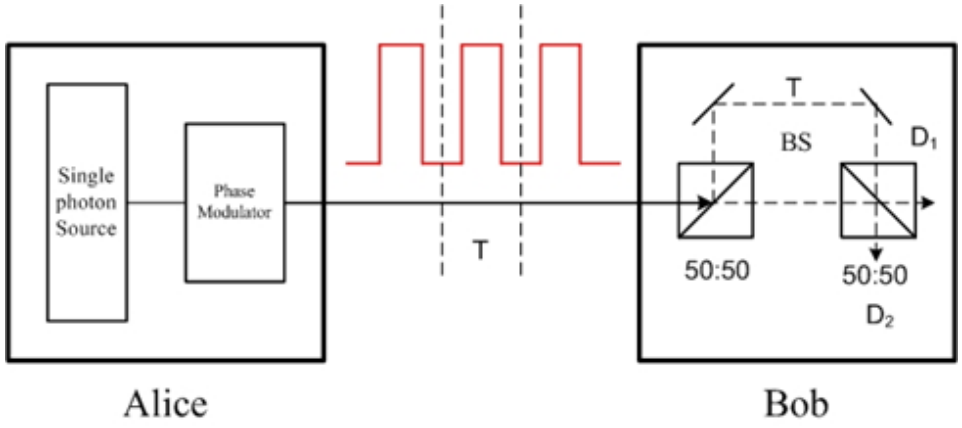

Figure 7. The DPS scheme between two parties (Alice and Bob) where $|\varphi\rangle$ is the qubits that are transmitted in certain time (Differential-Phase). T is the slot time considered to be measured by detectors on Bob's side, also Bob uses the double beam splitters with accuracy of 50:50.

3. The QKD Protocols Features Based on Quantum Computing

As stated in the previous schemes, each quantum key distribution protocol is addressed in two critical aspects. The first aspect is the movement of particles, which is related to quantum mechanics and theories of physics. Moreover, this aspect shows the physical motion of specific photons using the required observables (polarization, momentum, mass, etc.) at initiation and measurement conditions. The second aspect represents the comfort of using a classical system (or our current computer) with quantum bits (qubits). The ability to convert qubits to bits in existing platforms using quantum computers is still unavailable.

Furthermore, the conclusion of the previous aspects has been collected on the cryptographic point of view as shown in Table 8 [60], where the main security classifications of the quantum cryptography for some well-known QKD protocols (review of previous literature) are presented. These classifications focused on the instructions for the QKD protocols that used either the law of physics or the fundamentals of classical cryptography. However, many cryptographic features are still not available for approval today, such as professional quantum apparatus and quantum hardware. To improve the presence of these features, the entire classical system used, and the upcoming quantum system should overlap. For example, security attackers attempt to break any information system by discovering vulnerabilities and weaknesses into those platforms.

Table 8. The mechanism and features of well-known GKD protocols.

Cases	Quantum Key Distribution Protocols								
	BB84	**B92**	**SARG04**	**COW**	**KMB09**	**EPR**	**S09**	**S13**	**DPS**
Properties	Heisenberg	Heisenberg	Heisenberg	Arbitrary	Heisenberg	Entanglement	K_p, K_s	Heisenberg	Arbitrary
Number of States	4 states	2 States	4 States	Time slots	2 states	2 EPR	Arbitrary states	4 states	4 States
Detection of presence	QBER	QBER	QBER	Break of coherence	ITER	Bell's inequality	appending parity bits	Random Seed	Timeslot
Polarization	Orthogonal	Non orthogonal	Orthogonal	Arbitrary	Arbitrary	Orthogonal	Bit-Flip Phase-Flip	2 orthogonal	DPS
State Probability	Various	50%	50%	Calculated	50%	Equal	Various	Various	Equal
Qubit String	Discrete	Discrete	Discrete	Discrete	Discrete	Discrete	No	Discrete	Discrete
Classical channels	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes
Decoy States	No	No	No	Yes	No	No	Yes	No	No
Sifting phase	Revealing Bases	Alice = 1 − Bob	Revealing non-orth. state	Revealing times 2k + 1	Revealing Indices	Bell's Inequality	No	Revealing Bases	Timeslot
Bell's inequality	No	No	No	No	No	Yes	No	No	No

Table 8. *Cont.*

Cases	Quantum Key Distribution Protocols								
	BB84	**B92**	**SARG04**	**COW**	**KMB09**	**EPR**	**S09**	**S13**	**DPS**
PNS attack	Vulnerable	Vulnerable	better than BB84	Robust	Robust	N/A	N/A	N/A	Robust
IRUD attack	Vulnerable	Vulnerable	Vulnerable	Under Test	Under Test	Vulnerable	N/A	N/A	N/A
BS attack	Vulnerable	Vulnerable	Robust	Robust	Robust	Vulnerable	N/A	N/A	Robust
DoS attack	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	Vulnerable	N/A	N/A	Robust
MAM attack	Vulnerable	Robust	Robust	Robust	Robust	Robust	Robust	N/A	Robust
IRA attack	Vulnerable	Vulnerable	Robust	Robust	Robust	Bell's inequality	Robust	N/A	Robust
Authentication	No	No	No	No	No	No	No	Yes [classic]	No

4. Runtime Analysis of QKD Protocols

Since the processing time of each QKD protocol is a critical term for secret key generation, the runtime execution has been experimented with in a specific ecological scheme. The same number of n-qubits has been implemented, and all QKD algorithms are formatted based on the original protocol, with many improvements recently updated. Moreover, the experiments present runtime simulations of all QKD protocols in a classical system by applying quantum libraries [61,62] in MATLAB. All these simulations reflected a high percentage of reality as far as they are used in a quantum system. These simulations were implemented by applying 500 qubits to create a secret key (SSK), where each protocol has an independent scheme. In terms of the number of qubits that can be sent from Alice to Bob, each QKD protocol expects an error rate (covered qubits). Based on the recorded error rate, both participants can decide to either extend the protocol's operations or cancel the communication entirely.

Figures 8–16 [60] show only one of the runtime execution measurements for each QKD protocol, although several implementations were performed to determine the behavior of each QKD protocol. The measurements show variations during the process of each QKD protocol, as well as the real-time of each protocol, especially, when the QKD protocol applies more than 500 qubits between two legitimate parties. The gaps between each runtime execution depend on the properties of the states used (arbitrary, superposition, or entangled) and the number of bases used (orthogonal or non-orthogonal) that are employed for encrypting/decrypting the plaintext X. In addition, the differences between the studied QKD are related to the type of communication channels that are initiated between two legitimate parties (e.g., quantum or public channels). Each communication channel should have a certain mechanism, either to generate a specific state of a photon or to measure the received qubits. Therefore, runtime execution measurements were applied in the absence of Eve since Eve can cause relative and unstable errors. These errors can produce large variations based on different methods [63]. However, noise was applied in these experiments, which are usually generated by the environment.

After simulating each QKD algorithm by applying the runtime execution T(n) function, the results show the relative complexity in each QKD protocol with obvious variations. The T(n) function is linear and will lead to an increase in the life of the key generation, as long as the communication is active. For instance, the SARG04 protocol is similar to the BB84 protocol, except that SARG04 has a higher complexity than the BB84 protocol. Furthermore, SARG04 takes ($\approx$ 0.815 ms) to generate a secret key more than BB84 ($\approx$ 0.364 ms). Table IX shows the runtime execution of the QKD protocol limited to 500 qubits (ms).

In addition, the runtime execution measurement for the DPS protocol shows differentiation because initiating the DPS requires applying a photon in multiple states (arbitrary states). The differentiation of running each QKD protocol at certain functions and operations is based on the nature of each operation complexity such as converting a bit to qubit. In addition, the looping of each function produces sequential procedures before creating the SSK.

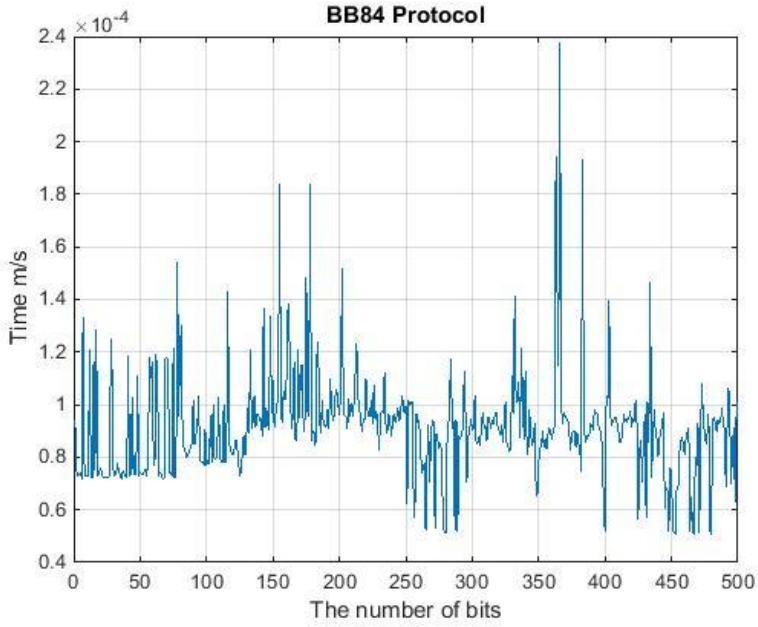

Figure 8. The Runtime Execution is measured in the BB84 by applying 500 qubits of transferred data (qubits) into quantum channel as well as reconciliation phase into classical channel. This QKD protocol was experimented with in a classical system using quantum libraries, where these libraries designed in Python as well as MATLAB. The accuracy of these experiments would be high enough to run the QKD protocols and measure the time needed to create a secret key.

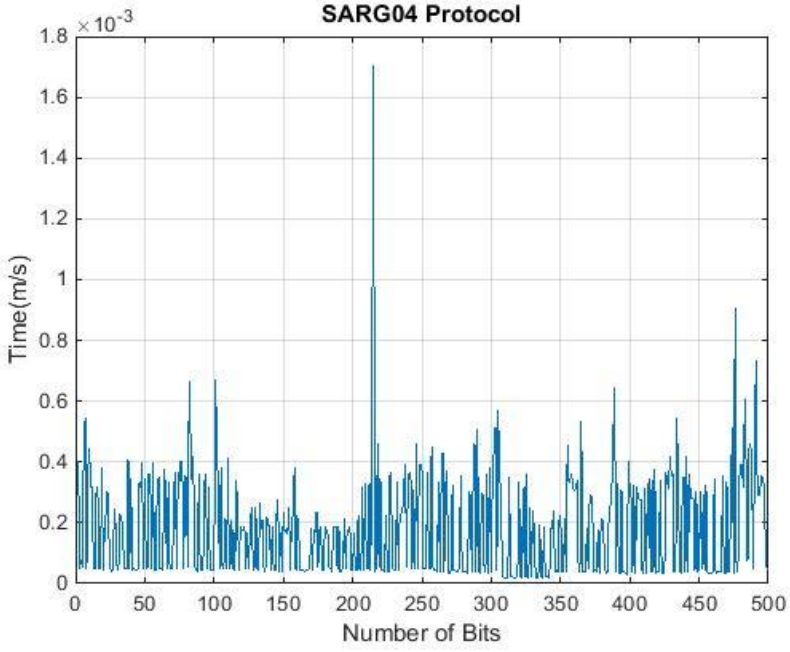

Figure 9. The Runtime Execution in the SARG04 after applying 500 qubits of transferred data (qubits) into quantum channel as well as reconciliation phase into classical channel. The SARG04 protocol was experimented with in a classical system using quantum libraries, where these libraries designed in Python as well as MATLAB. The accuracy of these experiments would be high enough to run the QKD protocols and measure the time needed to create a secret key.

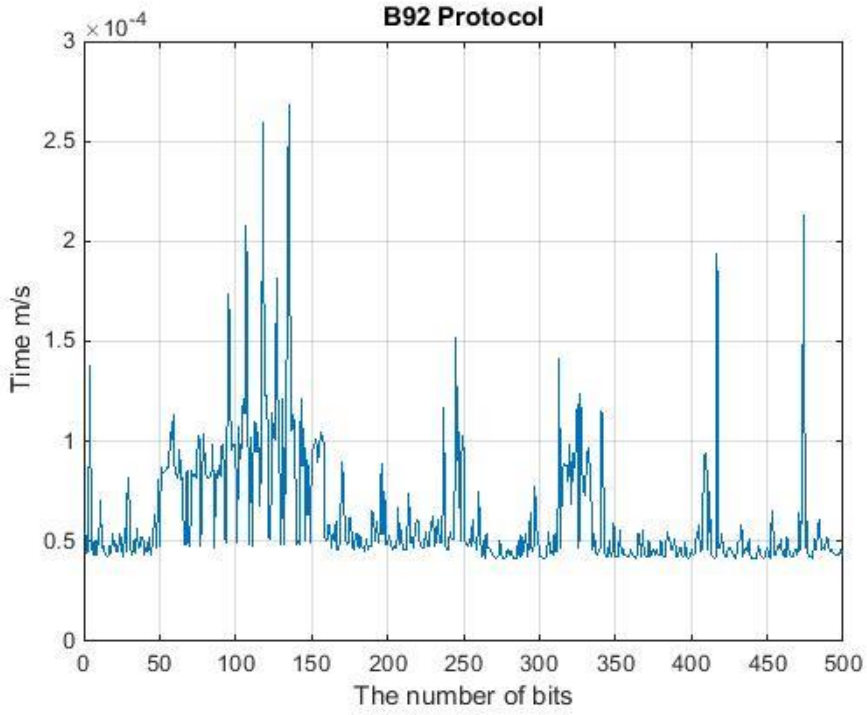

Figure 10. The Runtime Execution in the B92 after applying 500 qubits of transferred data (qubits) into quantum channel as well as reconciliation phase into classical channel. The B92 protocol was experimented with in a classical system using quantum libraries, where these libraries designed in Python as well as MATLAB. The accuracy of these experiments would be high enough to run the QKD protocols and measure the time needed to create a secret key.

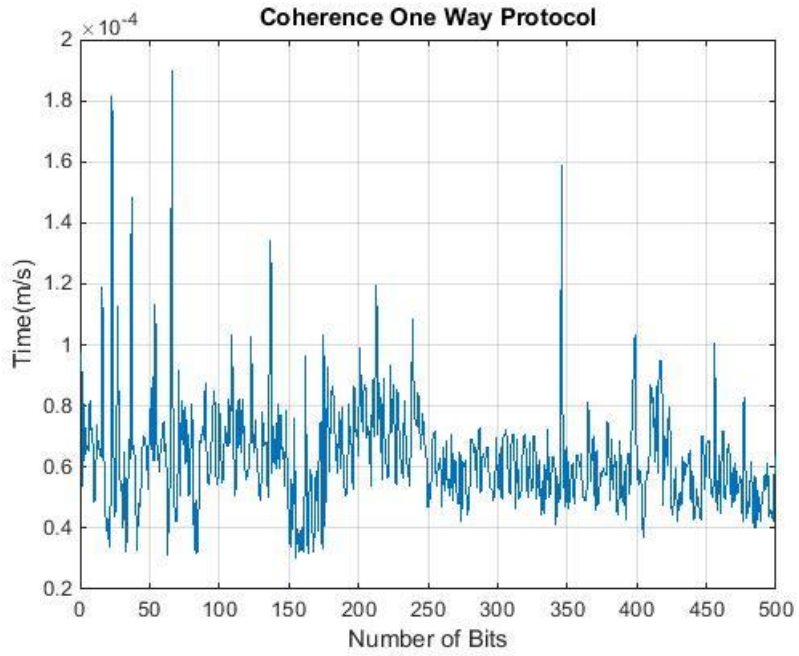

Figure 11. The Runtime Execution in COW after applying 500 qubits of transferred data (qubits) into quantum channel as well as reconciliation phase into classical channel. The COW protocol was experimented with in a classical system using quantum libraries, where these libraries designed in Python as well as MATLAB. The accuracy of these experiments would be high enough to run the QKD protocols and measure the time needed to create a secret key.

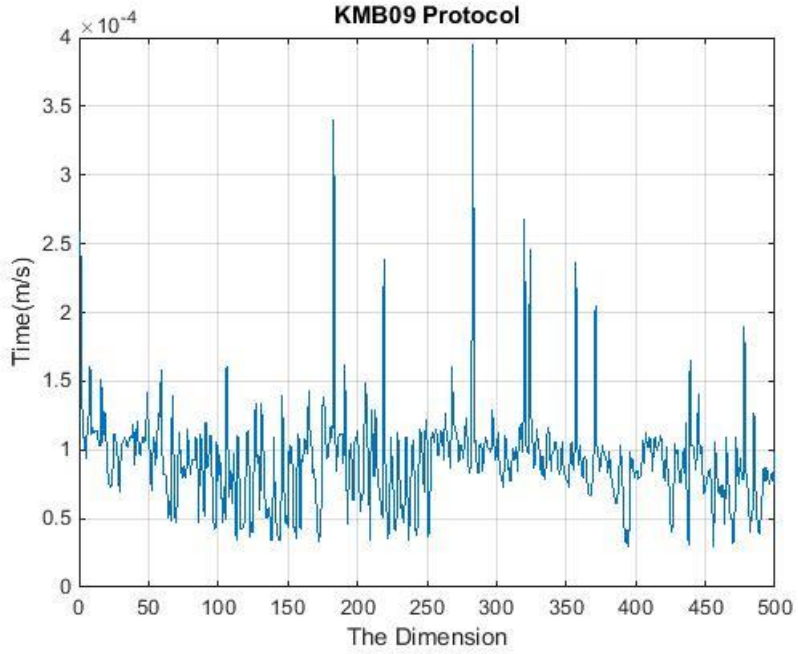

Figure 12. The Runtime Execution in the KMB09 after assigning multidimensional qubits of transferred data (qubits) into quantum channel as well as reconciliation phase into classical channel. The KMB09 protocol was experimented with in a classical system using quantum libraries, where these libraries designed in Python as well as MATLAB. The accuracy of these experiments would be high enough to run the QKD protocols and measure the time needed to create a secret key.

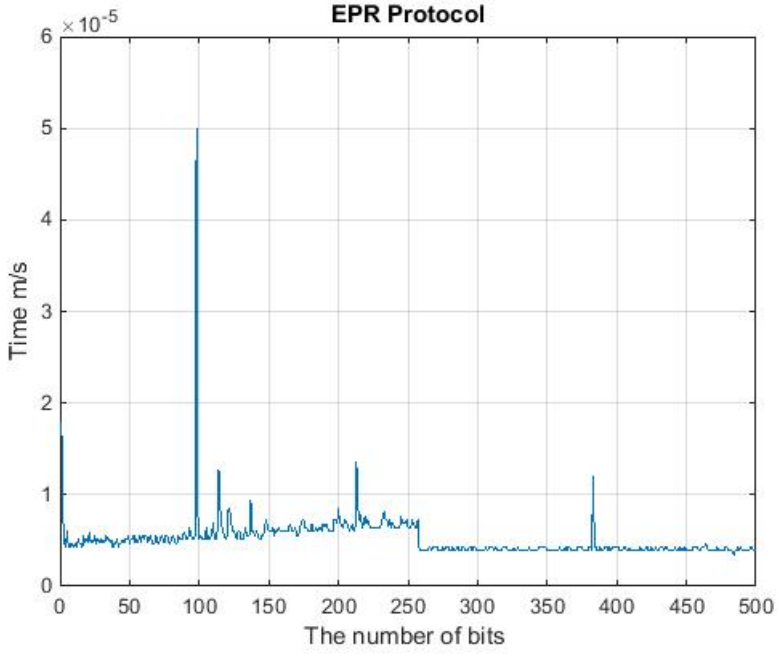

Figure 13. The Runtime Execution in the EPR after applying 500 qubits of transferred data (qubits) into quantum channel as well as reconciliation phase into classical channel. The EPR protocol was experimented with in a classical system using quantum libraries, where these libraries designed in Python as well as MATLAB. The accuracy of these experiments would be high enough to run the QKD protocols and measure the time needed to create a secret key.

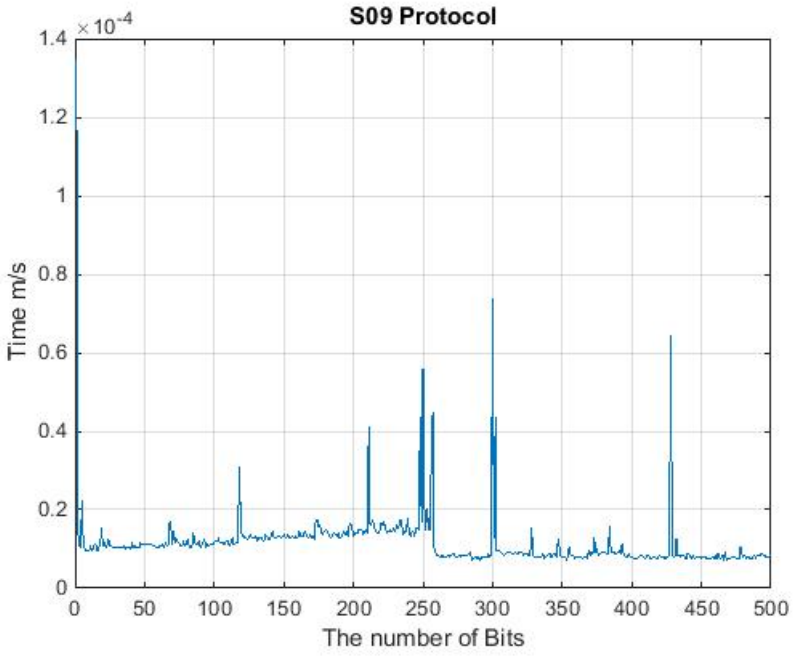

Figure 14. The Runtime Execution in the S09 after applying 500 qubits of transferred data (qubits) into quantum channel as well as reconciliation phase into classical channel. The S09 protocol was experimented with in a classical system using quantum libraries, where these libraries designed in Python as well as MATLAB. The accuracy of these experiments would be high enough to run the QKD protocols and measure the time needed to create a secret key.

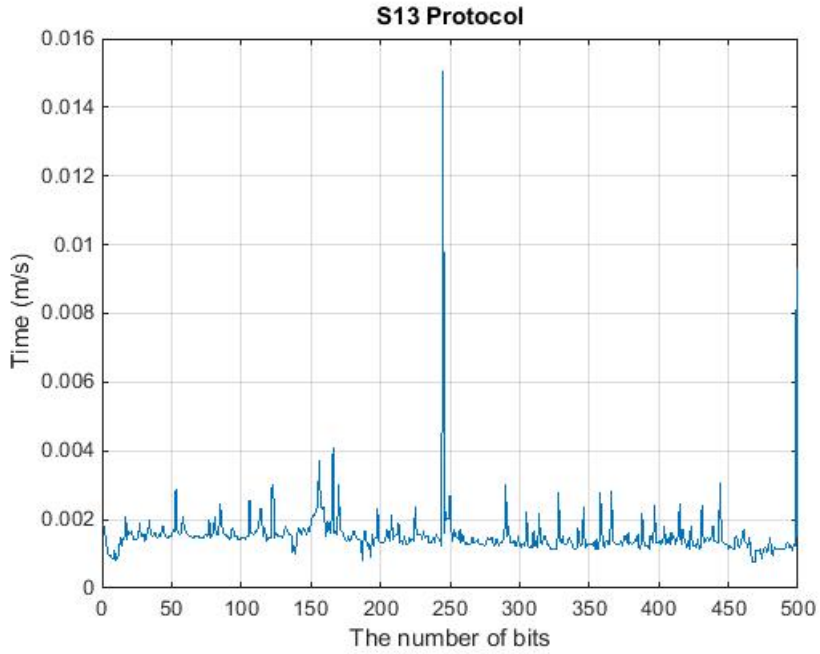

Figure 15. The Runtime Execution in the S13 after applying 500 qubits of transferred data (qubits) into quantum channel as well as reconciliation phase into classical channel. The S13 protocol was experimented with in a classical system using quantum libraries, where these libraries designed in Python as well as MATLAB. The accuracy of these experiments would be high enough to run the QKD protocols and measure the time needed to create a secret key.

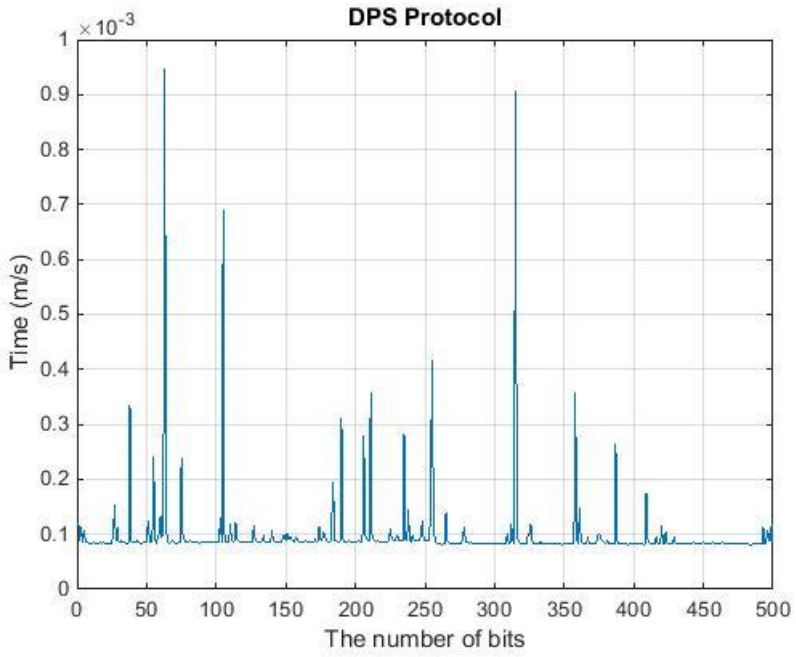

Figure 16. The Runtime Execution in the DPS after applying 500 qubits of transferred data (qubits) into quantum channel as well as reconciliation phase into classical channel. The DPS protocol was experimented with in a classical system using quantum libraries, where these libraries designed in Python as well as MATLAB. The accuracy of these experiments would be high enough to run the QKD protocols and measure the time needed to create a secret key.

For instance, matching the measured qubits with the expected raw key depends on a separate reconciliation algorithm. Hence, the operations of running each reconciliation algorithm are calculated with the entire execution process, which starts from initiating a stream of bits until the secret key SSK is created before correcting any errors. Therefore, the following formula is applied to measure the runtime execution for each QKD protocol:

$$T(n) = \sum_{i=1}(p.t), \tag{15}$$

where p is a single loop of each QKD protocol process, t is the total time taken by each loop, and n is the length of plaintext X. Furthermore, each QKD protocol should have multiple operations during the quantum communication and reconciliation (usually classical communications) phases to generate a secret key. Afterwards, the QKD protocols are similar in three sequential phases: the first phase is initiation and preparation, the second phase is a submission, and the third phase is a reconciliation phase. These phases are shown in the previous QKD protocol, Algorithm 1.

Algorithm 1: QKD Protocol

1. Initiate n Qubits	*// prepare a plaintext*
2. ***for***: each $n \rightarrow (+ \,\|\|\, \times)$	*// Initiation loop*
3. ***if*** $(n_i == +)$:	
4. ***then*** n_i $(0 \,\|\|\, 90)$	// Loop (1)
5. ***else*** n_i $(45 \,\|\|\, 135)$	
6. ***end***; ***end***	*// ending the loop*
7. Reconciliation phase:	
8. ***for***: $1 \rightarrow n$	*// reconciliation loop*
9. ***if*** : $(i \neq j)$	*// loop (2)*
10. *use different mechanisms*	
11. *to correct error,*	
12. ***else*** : *accept*	
13. ***end***; ***end***;	*//ending the loop*

More precisely, many execution loops occurred in the S09 and S13 protocols, especially during the reconciliation phase, unlike the BB84 and B92 protocols. The KMB09 protocol uses unique reconciliation procedures by exchanging indices instead of bases, which makes the correction phase more efficient. Therefore, runtime execution reflects the simplicity of using each QKD protocol. Thus, the BB84 protocol is classified as a simple QKD protocol based on previous measurement.

5. Comparison between QKD Protocols

Subsequently, several cryptographic approaches were extracted from the well-known QKD protocol (BB84, B92, SARG04, EPR, COW, DPS, KMB09, S09, and S13), which clarified the variations between these QKD protocols. These variations assist in realizing the weaknesses and strengths of each QKD process during communication between two users. As shown in the previous sections, some technical details are presented in the definition of each protocol, especially when a certain protocol has a unique design, such as the Coherent-One-Way (COW) protocol. The COW protocol depends on the insertion of the decoy states (μ_i) in pulse transmission. Using decoy states means more protection against PNS attacks, while extra time is required during either the submission or reconciliation phases.

Moreover, the QKD protocols vary in terms of the techniques used to determine the reliability of each QKD protocol against attack challenges. Security, simplicity, and efficiency are factors that are typically applied to measure QKD protocols. The previous QKD protocols were tested using one of these factors, where the simplicity factor was applied to test the runtime execution at a limited number of qubits. On the other hand, the major issue in most QKD protocols is the verification of the identity of the communicated parties. As demonstrated in Table 9, the authentication is not approved in the previous QKD protocols except for the S13 protocol, which requires verification of user identity in the classical channel. Moreover, verification requires additional procedures that reduce the lifetime of the protocol. Thus, spending a long term could expose communication, especially when the public channel is used to confirm the transferred data during the quantum channel. In other words, the multiple processes that occur in the classical channel provide a high rate of information gain by eavesdroppers.

Table 9. The Runtime Execution of QKD protocols limited up to 500 qubits (ms).

QKDP	Input (Qubit)	Output (Qubit)	Time (ms)
BB84	500	142	0.164
B92	500	119	0.177
SARG04	500	247	0.815
KMB09	16	362	0.012
EPR	500	119	0.860
DPS	500	N/A	Constant
S09	500	N/A	0.927
S13	500	N/A	0.639
COW	500	126	0.686

In addition, there are three categories that cryptosystem designers try to achieve. First, designing a cryptographic algorithm assists the submitted data in being heavily mixed and complex. Second, creating encryption/decryption keys locks and unlocks the submitted plaintext X. Finally, the secure keys should be distributed between trusted communicating entities. In other words, there are still several hard attempts to break the QKD protocol or at least show weak points. None of these attempts can be functionally considered, even when Eve tries to intercept and resend the submitted particles and then generate new particles. On the other hand, the eavesdropper wishes to read at least 25% of the originally submitted message, where the rest of the message remains for guessing as null.

Furthermore, Intercepted/Resend attacks (IRA) are a well-known strategy against QKD protocols. IRA is the most popular attack, which is based on replacing some of

the submitted qubits by applying a random basis ($\times$ or $+$) when Alice sends qubits to Bob. Meanwhile, Eve leaves the rest of these qubits without changes because Eve wishes that the qubits should be received by Bob without changes. Next, when Alice and Bob start to compare the matching qubits, Eve constructs other qubits to be incompatible measurements. More precisely, if Eve attempts to listen to the transmitted qubits between Alice and Bob, Eve and Bob will use identical bases sent by Alice. Bob obtains identical bases to the eavesdropper, but no one can detect Eve. In other words, if Eve used a different measurement to intercept the bases sent by Alice, Eve will would experience uncertain changes in the state's polarization.

6. Conclusions

This paper presents a set of QKD protocols over a period of time, as QKD is a new generation of cryptography in the information theory world. Moreover, several issues in quantum cryptography are theoretically solved using the QKD protocol; in particular, QKD is powered by quantum mechanics. The reason behind the strength of quantum mechanics is the non-cloning theory, which produces an alteration of any permeation. On the other hand, the explanation of ambiguous ideas was clearly discovered in this study to show the mechanism of each QKD protocol. One of the most important insights is to implement the runtime of each QKD protocol, as the use of multi-communications in a classical channel requires extending the life of the QKD protocol process as well as increasing the rate of information attacks. The classical channel is used heavily during the implementation of each QKD protocol, where a significant amount of processing time is spent during the reconciliation phase. In contrast, a classical channel is needed, as long as there is no way to reconcile the submitted information during the quantum channel. In other words, the classical channel should not completely affect the implementation of the QKD protocol, as this will lead to an infinite experience. Finally, the QKD protocol provides a Secure Shared Key (SSK) between legitimate parties through the secure communication. The secret key should be robust against any type of information attack with a 0.0% exposure to the SSK. QKD is expected to be the next generation of secret key in various information exchange systems.

Funding: This research received no external funding.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Schneier, B. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
2. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
3. Diffie, W.; Hellman, M. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
4. Wiesner, S. Conjugate coding. *ACM Sigact News* **1983**, *15*, 78–88. [CrossRef]
5. Brassard, G. Brief history of quantum cryptography: A personal perspective. In *IEEE Information Theory Workshop on Theory and Practice in Information-Theoretic Security*; IEEE: Piscataway, NJ, USA, 2005; Volume 2005, pp. 19–23.
6. Walk, N.; Ralph, T.C.; Symul, T.; Lam, P.K. Security of post-selection based continuous variable quantum key distribution against arbitrary attacks. In *CLEO: Applications and Technology*; Optical Society of America: Washingdon, DC, USA, 2011; p. JTuC4.
7. Oesterling, L.; Hayford, D.; Friend, G. Comparison of commercial and next generation quantum key distribution: Technologies for secure communication of information. In Proceedings of the 2012 IEEE Conference on Technologies for Homeland Security (HST), Waltham, MA, USA, 13–15 November 2012; pp. 156–161.
8. Possignolo, R.T.; Margi, C.B. A quantum-classical hybrid architecture for security algorithms acceleration. In Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, Liverpool, UK, 25–27 June 2012; pp. 1032–1037.
9. Yanofsky, N.S.; Mannucci, M.A. *Quantum Computing for Computer Scientists*; Cambridge University Press: Cambridge, UK, 2008.
10. 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. *Adv. Opt. Photon.* **2020**, *12*, 1012. [CrossRef]
11. Sharbaf, M.S. Quantum cryptography: An emerging technology in network security. In Proceedings of the 2011 IEEE International Conference on Technologies for Homeland Security (HST), Waltham, MA, USA, 15–17 November 2011; pp. 13–19.

12. Barker, W.C.; Barker, E.B. *SP 800-67 Rev. 1. Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*; Citeseer: Princeton, NJ, USA, 2012.
13. Schneier, B. Description of a new variable-length key, 64-bit block cipher (Blowfish). In *International Workshop on Fast Software Encryption*; Springer: Berlin/Heidelberg, Germany, 1993; pp. 191–204.
14. Patarin, J.; Goubin, L. Asymmetric cryptography with S-Boxes Is it easier than expected to design efficient asymmetric cryptosystems? In Proceedings of the International Conference on Information and Communications Security, Beijing, China, 11–14 November 1997; pp. 369–380.
15. Lo, H.-K.; Ma, X.; Chen, K. Decoy state quantum key distribution. *Phys. Rev. Lett.* **2005**, *94*, 230504. [CrossRef] [PubMed]
16. Bužek, V.; Hillery, M. Quantum copying: Beyond the no-cloning theorem. *Phys. Rev. A* **1996**, *54*, 1844. [CrossRef]
17. Steege, M. *Free-Space Optics: A Viable, Secure Last Mile Solution?* Sans Institute: Bethesda, MD, USA, 2002.
18. Niemiec, M.; Pach, A.R. Management of security in quantum cryptography. *IEEE Commun. Mag.* **2013**, *51*, 36–41. [CrossRef]
19. Cabello, A.; Feito, Á.; Lamas-Linares, A. Bell's inequalities with realistic noise for polarization-entangled photons. *Phys. Rev. A* **2005**, *72*, 052112. [CrossRef]
20. Bennett, C.H.; Brassard, G. Quantum cryptography: Public key distribution and coin tossing. *Theor. Comput. Sci.* **2014**, *560*, 7–11. [CrossRef]
21. Russell, J. Application of quantum key distribution. In Proceedings of the MILCOM 2008—2008 IEEE Military Communications Conference, San Diego, CA, USA, 16–19 November 2008; pp. 1–6.
22. Gottesman, D.; Lo, H.-K. From quantum cheating to quantum security. *arXiv* **2001**, arXiv:quant-ph/0111100. [CrossRef]
23. Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **1999**, *41*, 303–332. [CrossRef]
24. Cao, Z.; Liu, L. Improvement of one quantum encryption scheme. *Int. J. Quantum Inf.* **2012**, *10*, 1250076. [CrossRef]
25. Zhao, S.-M.; Li, F.; Zheng, B.-Y. A proof of security of quantum key distribution in probabilistic clone scheme. In Proceedings of the International Conference on Communication Technology Proceedings, ICCT 2003, Beijing, China, 9–11 April 2003; Voume 2, pp. 1507–1509.
26. Sharma, R.D.; De, A. A new secure model for quantum key distribution protocol. In Proceedings of the 2011 6th International Conference on Industrial and Information Systems, Kandy, Sri Lanka, 16–19 August 2011; pp. 462–466.
27. Jouguet, P.; Kunz-Jacques, S. High performance error correction for quantum key distribution using polar codes. *arXiv* **2012**, arXiv:1204.5882. [CrossRef]
28. Cerf, N.J.; Bourennane, M.; Karlsson, A.; Gisin, N. Security of quantum key distribution using d-level systems. *Phys. Rev. Lett.* **2002**, *88*, 127902. [CrossRef]
29. Kartheek, D.N.; Amarnath, G.; Reddy, P.V. Security in quantum computing using quantum key distribution protocols. In Proceedings of the 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), Kottayam, India, 22–23 March 2013; pp. 19–25.
30. Zeng, G.; Wang, X. Quantum key distribution with authentication. *arXiv* **1998**, arXiv:quant-ph/9812022.
31. Sharma, A.; Ojha, V.; Lenka, S.K. Security of entanglement based version of BB84 protocol for Quantum Cryptography. In Proceedings of the 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China, 9–11 July 2010; Volume 9, pp. 615–619.
32. Shor, P.W.; Preskill, J. Simple proof of security of the BB84 quantum key distribution protocol. *Phys. Rev. Lett.* **2000**, *85*, 441. [CrossRef]
33. Scarani, V.; Acin, A.; Ribordy, G.; Gisin, N. Quantum cryptography protocols robust against photon number splitting attacks for weak laser pulse implementations. *Phys. Rev. Lett.* **2004**, *92*, 057901. [CrossRef]
34. ElGamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory* **1985**, *31*, 469–472. [CrossRef]
35. Stipčević, M. How secure is quantum cryptography? In Proceedings of the 2012 35th International Convention MIPRO, Opatija, Croatia, 21–25 May 2012; pp. 1529–1533.
36. Ghazali, L.I.A.; Abas, A.F.; Adnan, W.A.W.; Mokhtar, M.; Mahdi, M.A.; Saripan, M.I. Security proof of Improved-SARG04 protocol using the same four qubit states. In Proceedings of the International Conference on Photonics 2010, Langkawi, Malaysia, 5–7 July 2010.
37. Abushgra, A.A. SARG04 and AK15 Protocols Based on the Run-Time Execution and QBER. In Proceedings of the 2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP), Zhuhai, China, 8–10 January 2021; pp. 176–180.
38. Rass, S.; Schartner, P.; Greiler, M. Quantum coin-flipping-based authentication. In Proceedings of the 2009 IEEE International Conference on Communications, Dresden, Germany, 14–18 June 2009; pp. 1–5.
39. Zhou, Y.; Zhou, X.; Gao, J. Scarani-acin-ribordy-gisin decoy-state protocols in quantum key distribution with a heralded single photon source. In Proceedings of the 2009 9th International Conference on Electronic Measurement & Instruments, Beijing, China, 16–19 August 2009; pp. 4–751.
40. Scarani, V.; Bechmann-Pasquinucci, H.; Cerf, N.J.; Dušek, M.; Lütkenhaus, N.; Peev, M. The security of practical quantum key distribution. *Rev. Mod. Phys.* **2009**, *81*, 1301. [CrossRef]
41. Bennett, C.H. Quantum cryptography using any two nonorthogonal states. *Phys. Rev. Lett.* **1992**, *68*, 3121. [CrossRef] [PubMed]
42. Elboukhari, M.; Azizi, M.; Azizi, A. Quantum Key Distribution Protocols: A Survey. *Int. J. Univers. Comput. Sci.* **2010**, *1*, 59–67.

43. Jobez, P.; Timoney, N.; Laplane, C.; Etesse, J.; Ferrier, A.; Goldner, P.; Gisin, N.; Afzelius, M. Towards highly multimode optical quantum memory for quantum repeaters. *Phys. Rev. A* **2016**, *93*, 032327. [CrossRef]
44. Gisin, N.; Ribordy, G.; Zbinden, H.; Stucki, D.; Brunner, N.; Scarani, V. Towards practical and fast quantum cryptography. *arXiv* **2004**, arXiv:quant-ph/0411022.
45. Stucki, D.; Brunner, N.; Gisin, N.; Scarani, V.; Zbinden, H. Fast and simple one-way quantum key distribution. *Appl. Phys. Lett.* **2005**, *87*, 194108. [CrossRef]
46. Gottesman, D.; Lo, H.-K.; Lutkenhaus, N.; Preskill, J. Security of quantum key distribution with imperfect devices. In Proceedings of the International Symposium onInformation Theory, Chicago, IL, USA, 27 June–2 July 200; p. 136.
47. Singh, H.; Gupta, D.L.; Singh, A.K. Quantum key distribution protocols: A review. *J. Comput. Eng.* **2014**, *16*, 1–9. [CrossRef]
48. Inoue, K.; Waks, E.; Yamamoto, Y. Differential phase shift quantum key distribution. *Phys. Rev. Lett.* **2002**, *89*, 037902. [CrossRef]
49. Khan, M.M.; Murphy, M.; Beige, A. High error-rate quantum key distribution for long-distance communication. *New J. Phys.* **2009**, *11*, 063043. [CrossRef]
50. Serna, E.H. Quantum key distribution protocol with private-public key. *arXiv* **2009**, arXiv:0908.2146.
51. Han, Z.-F.; Li, H. Security of practical quantum key distribution system. In Proceedings of the 2011 International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), Chiang Mai, Tailand, 7–9 December 2011; pp. 1–3.
52. Khan, M.M.; Xu, J.; Beige, A. Improved Eavesdropping Detection in Quantum Key Distribution. *arXiv* **2011**, arXiv:1112.1110.
53. Einstein, A.; Podolsky, B.; Rosen, N. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.* **1935**, *47*, 777. [CrossRef]
54. Abushgra, A.; Elleithy, K. Initiated decoy states in quantum key distribution protocol by 3 ways channel. In Proceedings of the 2015 Long Island Systems, Applications and Technology, Farmingdale, NY, USA, 1 May 2015; pp. 1–5. [CrossRef]
55. Ekert, A.K. Quantum cryptography based on Bell's theorem. *Phys. Rev. Lett.* **1991**, *67*, 661. [CrossRef] [PubMed]
56. Hwang, T.; Lee, K.-C. EPR quantum key distribution protocols with potential 100% qubit efficiency. *IET Inf. Secur.* **2007**, *1*, 43–45. [CrossRef]
57. Serna, E.H. Quantum Key Distribution from a random seed. *arXiv* **2013**, arXiv:1311.1582.
58. Inoue, K.; Waks, E.; Yamamoto, Y. Differential-phase-shift quantum key distribution using coherent light. *Phys. Rev. A* **2003**, *68*, 022317. [CrossRef]
59. Honjo, T.; Uchida, A.; Amano, K.; Hirano, K.; Someya, H.; Okumura, H.; Yoshimura, K.; Davis, P.; Tokura, Y. Differential-phase-shift quantum key distribution experiment using fast physical random bit generator with chaotic semiconductor lasers. *Opt. Express* **2009**, *17*, 9053. [CrossRef] [PubMed]
60. Abushgra, A.; Elleithy, K. Differentiations of QKDPs in Run-Time Execution. p. 12. Available online: Khaledelleithy.org/Conferences/5-%20Differentiations%20of%20QKDPs%20in%20Run-Time%20Execution.pdf (accessed on 10 January 2022).
61. Rohde, P.P. *Quack! A Quantum Computer Simulator for Matlab*; Centre for Quantum Computer Technology, Department of Physics, University of Queensland: Brisbane, Australia, 2005.
62. Tan, S.M. A quantum optics toolbox for Matlab 5. *J. Opt. B Quantum Semiclass. Opt* **1999**, *1*, 161. [CrossRef]
63. Bruen, A.A.; Forcinito, M.A. *Cryptography, Information Theory, and Error-Correction: A Handbook for the 21st Century*; John Wiley & Sons: Hoboken, NJ, USA, 2011; Volume 68.