Develop a quantum key distribution application based on the BB84 protocol combined with a classical channel

Tat-Thang Nguyen[1], Thanh-Toan Dao[1], Nhu-Quynh Luc[2]
[1]Faculty of Electrical and Electronic Engineering, University of Transport and Communications, Hanoi, Vietnam
[2]Academy of Cryptography Techniques, Hanoi, Vietnam

Article Info

Article history:

Received Jul 23, 2024
Revised Oct 26, 2024
Accepted Nov 19, 2024

Keywords:

BB84 quantum key distribution
Network communication
Quantum basis
Quantum gates
Quantum inspire platform
Quantum key distribution

ABSTRACT

Amid the escalating concerns over internet security, quantum cryptography stands out as a highly promising solution for significantly enhancing the security of networking systems, emerging among them is the quantum key distribution (QKD) with the function of creating secret session keys a breeze when leveraging the intriguing properties of quantum mechanics. This study is rooted in the BB84 QKD method, where in the distribution process in the quantum realm is simulated to derive a shared key via a public channel connecting two clients with the assistance of a server, utilizing the quantum inspire (QI) platform to generate qubits within the BB84 protocol. The results, the findings regarding the performance of BB84 reveal that when the server is set up, and the key size increases to 4000 bits, the process of sending module takes 16.215 sec, the transfer module takes approximately 5.2 hours, the receive module takes 1.257 sec to finish the process for the final session key share. This indicates a noteworthy enhancement in the execution speed of QKD employing the BB84 protocol, which now holds the potential for reinforcing network security using quantum computing systems.

Corresponding Author:

Nhu-Quynh Luc
Academy of Cryptography Techniques
141 Chien Thang Road, Tan Trieu, Thanh Tri, Hanoi, Vietnam
Email: quynhln@actvn.edu.vn

1. INTRODUCTION

With the development of the Internet and the increasing eavesdropping on social media, system security become one of the biggest problems in using wired and wireless networks [1]-[4]. Within wireless communication networks, cryptography plays a pivotal role in ensuring data security by employing encryption and decryption processes [1], [5], [6]. In recent years, addressing the challenge of securing data exchange between two parties has led to the emergence of numerous access control mechanisms [3], [4], [6]. This intricate procedure involves addressing the security facets of communication, which include authentication, the secure exchange of data, and the appropriate closure of communication channels [3], [4].

In contrast to classical cryptography, quantum cryptography offers a secure method for establishing keys over a private channel, leveraging the principles of quantum theory. In wireless networks, quantum computation is harnessed to ensure the security of information transmission, quantum computing plays an important role in cryptography at present [7]. The emergence of quantum computing has brought opportunities in the evolution of cryptography. Consequently, researchers have learned more about and doing a lot of experiments on quantum cryptography harnesses the principles of quantum mechanics to construct robust communication systems that are impervious to attacks from both classical and quantum adversaries [7]. Quantum key distribution (QKD) is a cryptographic technique that uses quantum computing and enables two parties to securely exchange encryption keys over a public channel. In contrast to classical cryptographic techniques, which rely on the computational difficulty of solving certain mathematical problems [7]. QKD protocols have been the main topic of research and applicability to communication networks. In QKD, there exists a critical component known as the "Quantum Basis," which aligns with binary values and generates qubits [8], instrumental in establishing the secret sharing key within the communication environment. The core concept of this paper revolves around generating a quantum key through the employment of the BB84 protocol, ensuring that both the sender and recipient possess an identical secret key.

BB84 is a method for securely transmitting private keys to users by harnessing the principles of quantum mechanics. By harnessing Heisenberg's uncertainty principle and the no-cloning theorem, this method systematically eliminates the risk of tampering, ensuring a robust and secure approach [8]-[10]. BB84 protocol uses quantum gates such as Hadamard, and Pauli-X to generate random qubits into superposition states [11]. An illustrative practical application of the BB84 protocol is its use in facilitating secure private key sharing between two parties during online one-time password (OTP) generation [12].

In this study, the author's primary focus centers on the investigation, analysis, and assessment of the BB84 QKD protocol within a network context [13]. The study utilizes quantum inspire (QI) to create qubits, facilitating calculations and comparisons between qubits through measurement to obtain the session key. Subsequently, the author uses Python to measure the time taken for each process, enabling a thorough comparison, the author conducts simulations of the BB84 QKD process and assesses the effectiveness of key distribution within an internet-based environment. These aspects are comprehensively discussed within the various sections of this study.

2. RELATED WORKS

2.1. Quantum key distribution protocol

In the realm of quantum cryptography, researchers have put forth numerous advancements and theories related to the BB84 protocol. In their work, Basu and Sengupta [14] utilize the concept of a charge-coupled device, this method optimizes the transfer of qubits from sender to receiver by utilizing a quantum channel. This technology offers a distinct advantage over polarizers, as it solely requires the qubit transmission circuit, thereby streamlining the entire process.

One crucial and distinctive feature of QKD is the capacity to communicate with users to identify any third parties attempting to access key information during the distribution process. Any endeavor by a third party to eavesdrop on the key through measurement introduces detectable irregularities, thereby exposing their presence or intrusion. By harnessing the principles of quantum superposition [15], quantum entanglement [16], and conveying information in a quantum state, a communication system can ascertain, during the key distribution process, whether a third party is attempting eavesdropping. The security inherent in employing QKD, which is founded on the principles of quantum mechanics, differs from that of public key cryptography, which depends on computational complexity. As described by authors Bennett and Brassard, the BB84 protocol [17] is structured around two primary stages: i) step 1, involving the communication process conducted over the quantum channel and ii) step 2, which encompasses the communication process through the public channel and comprises two distinct phases [12].

Quantum cryptography entails encoding information using qubits. Quantum computing and quantum cryptography are captivating fields that harness the extraordinary characteristics of quantum bits, or qubits, and distinguish themselves from classical bits by transcending the binary limitations of '0' or '1'. Their exceptional trait lies in the capacity to exist simultaneously in a superposition of both '0' and '1'. Notably, it is impossible to replicate arbitrary qubits, as any such attempt can be readily detected, rendering them ideal for ensuring secure communication. Moreover, the phenomenon of entanglement allows two qubits to form an inherently private connection that cannot be intercepted or shared with any external entity, a distinct advantage over classical bits in conventional communication. In quantum cryptography, photons play a crucial role by assuming different polarization states for encoding and decoding information. To ascertain these polarization states, polarimeter systems come in two main types line polarization systems and cross-polarization systems, serving as pivotal tools in QKD, and secure communication.

Most current QKD systems operate via point-to-point fiber links [18]. This necessitates the presence of unused dark fibers, and for extended distances, the requirement to regenerate the key within trusted nodes, which can negatively impact both cost and security. Additionally, practical QKD systems exclusively employ one-time pad encryption for the secret key and rely on a parallel, classically authenticated communication channel.

Numerous protocols have successfully pushed the boundaries of secure communication over long distances, utilizing both satellite-based systems [19], and fiber-based systems [20]. Nonetheless, significant challenges persist in the quest for achieving extended links without resorting to quantum repeaters. A key obstacle in this endeavor arises from imperfections in detection systems, notably stemming from inherent dark counts. As communication distances grow, the signal strength diminishes, and the inherent noise within the detectors becomes increasingly dominant, impeding the reliable extraction of a confidential cryptographic key. The BB84 protocol recommends the use of advanced, high-performance components strategically selected to reduce the quantum bit error rate (QBER) in the detection system. This approach has yielded remarkable results, achieving an impressive signal transmission of 55 dB, equivalent to an astonishing 275 kilometers in ultra-low-loss fiber [21].

The crucial step towards deploying QKD systems [22], [23] within telecom operator infrastructures involves establishing a wide-area optical network where classical data channels and quantum key channels coexist. The initial challenge lies in addressing the difference in optical power transmission between classical and quantum channels in Figure 1, as quantum channels employ single or few-photon transmitters. In systems where classical and quantum channels are wavelength-multiplexed, this discrepancy can lead to significant crosstalk or non-linear effects penalties. Another challenge arises from the limited achievable link distance in QKD, typically spanning only a few hundred kilometers, due to channel losses and noise.

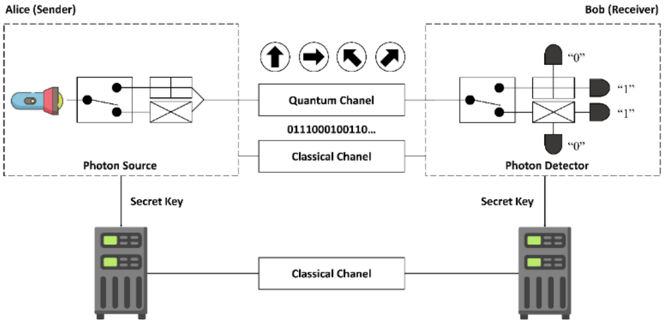


Figure 1. The communication model of the QKD service is based on the BB84 protocol

A quantum network [2] facilitates the transfer of qubits and in a broader context the generation of entanglement between various nodes interconnected in the network. These nodes can be as simple as photonic devices designed for single-qubit measurements or can involve more intricate and advanced equipment. In contrast to quantum computing, where tangible real-world value is only attainable once a quantum computer surpasses classical (super-)computers in performance, the journey to delivering benefits in the field of quantum networking is more gradual. Simple photonic devices are already capable of supporting practical applications, including quantum-secure communication within urban areas. Currently, quantum communication services are available in metropolitan regions, allowing for a security network. These services are primarily used for fundamental applications enabled by QKD.

2.2. Qiskit implementation

According to this research, the author used IPv4 as a communication protocol, disassembled and encapsulated to encode IPv4's DATA [24] frame when delivered to the channel and decoded "DATA" in IPv4 when received. The application's encapsulated data for transmission is a TCP/IP protocol that employs a client/server communication paradigm. In which a user or device (client) sends data by another computer (server) through the network.

This paper uses the QI platform and Qiskit [25] for the generation of qubits and superposition. The QI is an open-source software which is developed by Qutech, this enables quantum computing till the machine-level code of quantum assembly level language (QASM). Transitioning from bits and classical circuits to qubits and quantum circuits marks a significant shift. Much like a classical computer's bit, a qubit assumes the fundamental role of an information unit in a quantum computer. QI provides an algorithm on one of the available backends that will show the backend that has been selected for the experiment, the number of qubits used in the experiment, the name of the current experiment, and the number of shots of this experiment. In this paper, the author uses QX-26 which means using 26 qubits for calculating, quantum computing is still in its infancy. Quantum computing has great potential, but its unique challenges hinder mainstream adoption, these mainly revolve around the inherent properties of quantum mechanics and the practical difficulties of translating them into a computational context. One of the challenges is quantum decoherence, which refers to the loss of quantum behavior when a system interacts with its environment. This causes a quantum state to transition into a classical state significant obstacle because the time before decoherence occurs limits coherence time, or how long quantum information can be processed and stored.

The main contradiction between an ordinary bit and a qubit is the values that they accept, we have already seen bits take either zero or one. A qubit using the properties of quantum mechanics can take the superposition values of zero and one thus resulting in a wider range of possibilities and difficulty in deciphering. The Hadamard gate is involved in a superposition of the qubit, single qubit operations involve operating a qubit. It takes any input that is the position of the qubit and it transforms into $|1\rangle$ or $|0\rangle$ randomly, by using the probability of occurrence of $|1\rangle$ or $|0\rangle$. This is completely random thus this cannot be predicted by any eavesdropper, hence this handles the entanglement of the qubits. Even though having the circuits that need to measure each qubit individually to run the machine or algorithm, it becomes more accurate if it has given a high number of shots. These shots are the number of times the circuit is executed.

Once a measurement is performed on a particular basis, it becomes impossible to reverse the measurement and determine the potential result on a different basis. The original state is irreversibly altered, leaving only the collapsed state, this fact is of paramount importance in various QKD protocols. Table 1 shows the basis, math transforms of bit $|1\rangle$ or $|0\rangle$, $|0\rangle_x$ or $|1\rangle_x$ corresponding with the basis, and superposition using BB84 for creating session key share.

Table 1. Representation of the basic quantum state used in BB84

Present Z basis and X basis								
	Z basis	Math transform		X basis			Math transform	
Bit 0	$|0\rangle$	$\begin{bmatrix}1\\0\end{bmatrix}$		$|+\rangle$			$\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$	
Bit 1	$|1\rangle$	$\begin{bmatrix}0\\1\end{bmatrix}$		$|-\rangle$			$\frac{1}{\sqrt{2}}(|0\rangle-|1\rangle)$	
Photons with random values								
Client 1's bit	0	1	1	1	1	1	0	
Client 1's basis	+	×	×	+	×	×	×	
Client 1's polarization	→			↑				
Measurement between clients								
Client 2's basis	+	×	×	×	+	×	+	
Client 2's measurement	0	0	1	0	0	1	0	
Key match								
Basis match	+		×			×		
Agreed key	0		1			1		
a) Sender's bit								
Sender's bit				1	0	0	0	1
Sender's basis				x	x	x	z	x
Sender's qubit through quantum gate				$|1\rangle_x$	$|0\rangle$	$|0\rangle_x$	$|0\rangle$	$|1\rangle_x$
b) Receiver's bit								
Receiver's basis				x	z	x	x	z
Receiver's bit				1	0	0	0	0
Receiver's qubit through a quantum gate				$|1\rangle_x$	$|0\rangle$	$|0\rangle_x$	$|0\rangle_x$	$|0\rangle$
c) Matching position				x	x	x		
d) Shared session key				1	0	0		

3. THE STRUCTURE OF THE PROPOSED QUANTUM KEY DISTRIBUTION SERVICE IS BASED ON THE BB84 PROTOCOL

Firstly, the authors choose a platform for establishing basic quantum gates using the Qiskit based on Python language and called them to form an online account to create qubits and measurements for its superposition. This paper uses two quantum gates: Pauli-X gates ($|0\rangle \rightarrow |1\rangle$; $|1\rangle \rightarrow |0\rangle$) and Hadamard gates ($|0\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$; $|1\rangle \rightarrow \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$). After calling quantum gates [26] on Qiskit, the platform will start to generate qubits, and measurement is stated to know how qubit polarity cross polarization or straight polarization. From there, compare the qubits of the bases to create a shared session key.

In this work, the polarization technique is employed in conjunction with the BB84 protocol as outlined in Table 1. this method enables both parties to establish a shared secret key seamlessly, with no loss of information. In the realm of the BB84 protocol, the sender (client 1) takes on the task of randomly encoding an n-bit key into photons. These photons then embark on a journey over a quantum channel to reach the awaiting receiver (client 2). The receiver then utilizes matching polarization settings to derive the key, ensuring that both the sender and receiver agree on the polarization basis employed.

The BB84 protocol is dependent on the no-cloning theorem, a principle stating that quantum states cannot be replicated [27], to achieve precise encryption of the secret key on a bit-by-bit basis, the protocol converts the bits into the polarization state of photons. Since measuring the polarization state of a photon requires its destruction, any effort by an eavesdropper to measure it is likely to result in an incorrect polarization state, thus leading to immediate detection.

As the transmitted photons traverse an insecure channel, the potential for eavesdropping or tampering arises, to detect eavesdropping during the key exchange phase, client 2 opts for a subset of key bits around one-third, and shares these with client 1. Subsequently, client 1 examines if her corresponding values align with client 2, the identification of any disparities in this comparison signals a potential instance of eavesdropping or tampering, necessitating a restart of the process. Once all the test bits undergo successful verification, the remaining bits are then established as the secure key.

Eavesdroppers can only gain useful information from the photon, when the eavesdropper selects the correct basis, the interference remains undetected by client 1 and client 2. Nonetheless, in situations where the attacker selects an incorrect basis, which occurs roughly half the time, client 1 and client 2 will record conflicting bit values. This mismatch serves as a clear indicator of potential eavesdropping, this fundamental aspect of QKD safeguards the confidentiality and integrity of the shared secret key. If there is any difference between the two random subsets, the cause is certainly the attacker is spying. Then client 1 and client 2 return from beginning to start again. If there is no difference, the probability that the attacker escapes the above check is $P_{false} = \left(1 - \frac{\lambda}{4}\right)^{m}$.

Furthermore, even if the measurement basis becomes publicly known, it doesn't jeopardize the key's security. That's because, in the grand scheme of things, the knowledge of the basis becomes irrelevant to the attacker once all the photons have been measured. However, it's worth noting that security in QKD can be vulnerable to technological imperfections and limitations that might undermine the protocol's effectiveness. Quantum encryption systems, use quantum principles such as the Heisenberg uncertainty principle, the principle of unclonability, and the principle of quantum entanglement to protect and distribute encryption keys. Quantum encryption systems allow two people (or two organizations), who have not previously shared secret information, to communicate over public channels securely.

4. RESULTS AND DISCUSSION

4.1. Develop a program for a quantum key distribution service system based on the BB84 protocol using classical channels

The author has developed a module for simulating the key distribution process of the BB84 protocol to assess its effectiveness in a network environment. This module includes several components: the server module, the client module, an interface module, and the utilization of quantum generation via the Qiskit and QI platform present in Figure 2, the BB84 protocol scheme via the server is depicted. Step 1 involves initiating an interface for client usage within the scheme. The pivotal step is step 2, this step poses a significant challenge due to its involvement in generating a qubit through the server, particularly challenging because of implementing the BB84 protocol. The difficulty arises from the requirement to transmit the qubit through the quantum interface to reach the server. In addition to building a module for creating quantum bits using the BB84 protocol, the authors used an account on QI platform for building qubits using quantum gates like Hadamard gate and Pauli-X gate. Then resend the qubit from the account for the client to start to compare the session key to ensure the qubit transmission between two clients. The transmission of the BB84 protocol includes two states qubit and bases status transmissions, compare bases, and superposition. When two clients are set up for the protocol, a quantum account on QI is established to check the quantum gate used for creating quantum bits and superposition for the process. The qubit will be generated through QI and Qiskit and sent to the server for the comparison process.

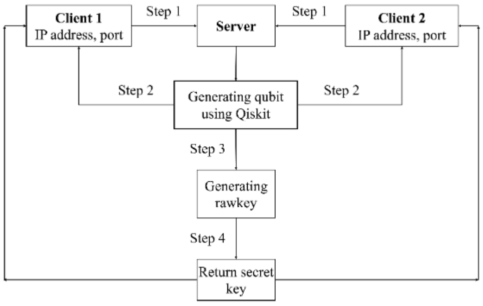


Figure 2. The proposed QKD scheme is based on the BB84 protocol

In Figure 3, the author shows the module for building the BB84 key protocol, the process begins once both the server and client are operational. Initially, qubit generation occurs, followed by awaiting the shared key to become available. The author opted for the Qiskit program to simulate quantum gates using Python, executed via an account on QI.

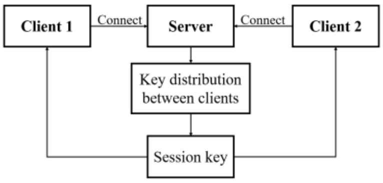


Figure 3. The operational flow of the proposed QKD service is based on the BB84 protocol combined with a classical channel

Server module: the server mission includes receiving packets from the client, distributing data flow, and sending them to the correct address.
Client module: the author designed and built the client module including the main components: IP address, and ID for verifying client and sending qubit.

After connecting successfully to the server, the client will proceed to create a qubit by sending the number of qubits on the client interface. Once the input has been generated, the server starts to call QI to generate the qubit and return the key after having measured and compared. Figure 4 has three modules for understanding the process of creating the final session key. First, is the client module for sending and entering number of qubits, this is the input from the client sent to the server, also has the receiving module for receiving the session key after decoding from the server and being encapsulated by the IPv4 protocol. Secondly, the transfer module on the server has the function of receiving input from clients and sending it to the Qiskit platform to generate qubits. Finally, Qiskit module used IPv4 to encode the qubit sent to the server module and back to the client module.

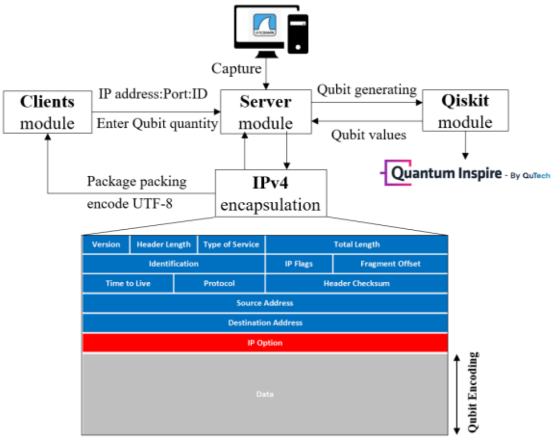


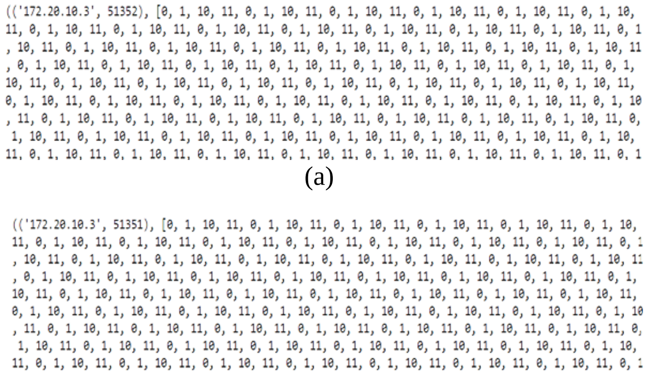
Figure 4. The module function for the proposed QKD service, extraction and encapsulation of DATA in IPv4

According to this study, the author used IPv4 as a communication protocol, in which the author sent the qubit data then packaged it to UTF-8 format and transferred it into the DATA frame in IPv4 packet when transmitted over the Internet. The public channel then separates the qubit in DATA frame in IPv4 packet upon receipt and then decodes that data. Figure 4 is a detail of the author's process of encapsulating data to encode into the DATA frame when sent to the channel and decode data in DATA frame in IPv4 when received. Once the server obtains the IP address, clients send data, and the server checks to validate if the address is correct. Subsequently, clients request a key value linked to their respective client IDs from the server. This key value is combined with the final value key, which is then sent back from the Qiskit. The data is decoded in UTF-8 format, and upon reaching a sufficient size, the key begins comparison processes, transforming into items and establishing a connection to the server, the Qiskit initiates measuring, and comparing the key. When measurements are finished, the final value key is returned to the clients, the data is now linked to the clients' IDs, accompanied by the final secret key. Consequently, the key generation session concludes, marking the termination of the process.

4.2. Evaluating the efficiency of the BB84 key distribution program

For this study, the authors have created software based on Python language to execute a simulation of BB84 QKD. Performing a test run of the BB84 key distribution software, the author uses a laptop computer with the configuration: Ryzen 7 3750 H-2.3 GHz, 8 Gb RAM. After connecting to the server, two clients start to connect and generate a qubit for the session, the BB84 used here is for the session key shared between two clients through a server with IPv4 setting. Using the server as an intermediary for qubit information transmission slows down the process, prolonging client wait times for the final shared key. Qubit transfer through the server, comparison of data, and subsequent return of values to clients heighten the complexity of the procedure. After creating input, the server calls the QI to start to generate qubits through quantum gates, number of shots the author uses is 512 shots to have the accuracy and best measurement of qubits.

After the process is finished, the key generated has a single state of superposition state due to the quantum gates being used in the process. Figures 5(a) and (b) present the final results of the session, displaying the IP addresses of both clients, the IDs of client 1 and client 2, and the session's secret key. Client 1 sends to client 2 including IP address and client 1's ID to verify that the destination is correctly verified. Once this verification is complete, the final session key is sent from the server to both clients. When using Wireshark capturing packet, Figures 6(a) and (b) result in encoding data when the server transmits to clients. This indicates the importance of IPv4 packaging when encoding the data from the server.



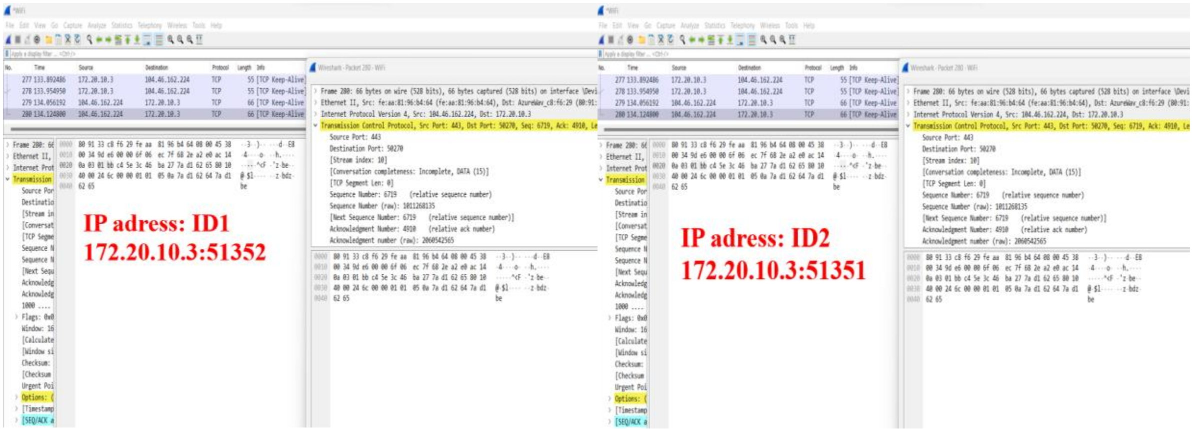
```
(('172.20.10.3', 51352), [0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10,
11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1
, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11
, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1,
10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11,
0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10
, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0,
 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10,
11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1
```

(a)

```
(('172.20.10.3', 51351), [0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10,
11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1
, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11
, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1,
10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11,
0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10
, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0,
 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10,
11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1, 10, 11, 0, 1
```

(b)

Figure 5. The result of the software: (a) when return session key qubit from the server to client 1 and (b) when return session key qubit from the server to client 2



*WiFi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Wireshark · Packet 280 · WiFi
IP adress: ID1
172.20.10.3:51352
IP adress: ID2
172.20.10.3:51351

```
No.  Time         Source          Destination     Protocol  Length  Info
277  133.892486   172.20.10.3     104.46.162.224  TCP       55 [TCP Keep-Alive]
278  133.954950   172.20.10.3     104.46.162.224  TCP       55 [TCP Keep-Alive]
279  134.056192   104.46.162.224  172.20.10.3     TCP       66 [TCP Keep-Alive
280  134.124800   104.46.162.224  172.20.10.3     TCP       66 [TCP Keep-Alive
```

```
> Frame 280: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Devi
> Ethernet II, Src: fe:aa:81:96:b4:64 (fe:aa:81:96:b4:64), Dst: AzureWav_c8:f6:29 (80:91:
> Internet Protocol Version 4, Src: 104.46.162.224, Dst: 172.20.10.3
v Transmission Control Protocol, Src Port: 443, Dst Port: 50270, Seq: 6719, Ack: 4910, Le
    Source Port: 443
    Destination Port: 50270
    [Stream index: 10]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 0]
    Sequence Number: 6719    (relative sequence number)
    Sequence Number (raw): 1011268135
    [Next Sequence Number: 6719    (relative sequence number)]
    Acknowledgment Number: 4910    (relative ack number)
    Acknowledgment number (raw): 2060542565
    1000 ....
  > Flags: 0x0
    Window: 16
    [Calculate
    [Window si
    Checksum:
    [Checksum
    Urgent Poi
  > Options: (
  > [Timestamp
  > [SEQ/ACK a
```

```
0000  80 91 33 c8 f6 29 fe aa  81 96 b4 64 08 00 45 38   ··3··)·· ···d··E8
0010  00 34 9d e6 00 00 6f 06  ec 7f 68 2e a2 e0 ac 14   ·4····o· ··h.····
0020  0a 03 01 bb c4 5e 3c 46  ba 27 7a d1 62 65 80 10   ·····^<F ·'z·be··
0030  40 00 24 6c 00 00 01 01  05 0a 7a d1 62 64 7a d1   @·$l···· ··z·bdz·
0040  62 65                                              be
```

(a) (b)

Figure 6. Showing the result of using wireshark to capture packets: (a) the content of the packet captured on wireshark is captured when server sends it to client 1 and (b) the content of the packet captured on wireshark is captured when server sends it to client 2

Table 2 shows the module processing time from start to finish when generating the session key for both clients. The results obtained indicate that, upon establishing a connection with the server, two clients commenced qubit generation via the interface. For 50 qubit transmission cycles, the module sending process between the two clients on the server takes approximately 0.766 sec to send the input to the server. However, when the size of the input module is up to 4000 bits, the computation time extends to approximately 16.215 sec. This highlights the ability to execute QKD quickly and securely over the internet, coupled with the rapidity of quantum computing facilitated by QI platform, the quantum computer system effortlessly simulates quantum gates and measurement procedures, all by the BB84 protocol. These results could be applied in the future as bit size increases and transmission time decreases, thanks to advancements in technology, this can be used in online transactions, and to enhance security in commercial trading.

Table 2. Results showing the runtime of all module

Module processing (qubit)	Sending (seconds)	Transfer (hours)	Receiving (seconds)
2000	7.363	≈2.4	0.542
3000	11.284	≈3.8	0.784
4000	16.215	≈5.2	1.257
Calculating the processing time for shared session key generation in the BB84 protocol			
Works	Quantity of trials	Size of quantum session key (bits)	Sending (seconds)
This works	1	50	0.766
	2	100	1.231
	3	500	2.052
	4	1000	4.058
	5	2000	7.363
	6	3000	11.284
	7	4000	16.215
[28]	1	170	5012.350
	2	550	4955.661
	3	670	5087.994
	4	720	5178.945
	5	1300	5011.237
	6	1423	4952.667
	7	1570	5012.557

This research executed 7 times transmission for the BB84 protocol with different sizes of key share, when changing the qubit size, the time needed to generate session key increases, the reason for the time increases is due to the number of shots set on the platform and the internet speed. Table 2 [28] demonstrates the speed of the key-shared session, the author uses this table to compare times with this research different key-share. The difference in the time shown is due to the quantum gate the author used based on the QI tool, Qiskit version 1.2.4, and code Python so the time for generating is more efficient compared with Table 2 [28] using java code and has a computer with better processing to assist and is more flexible in using quantum gates. In this study, the author performs BB84 under perfect conditions so QBER was not considered. Due to current technological limitations, these results can primarily be used for academic purposes in universities or colleges for learning and research. However, with advancements in technology, the results can be improved to offer greater benefits.

The author observes that as the number of qubits exchanged between two clients increases, the time required for all the modules likewise increases, and the growth rate is directly correlated. The computation time for generating the shared session key depends on the internet speed and the performance of the Qiskit platform on QI. This study is different from [12] in that it not only uses stimulation but also incorporates Qiskit and IPv4 for data transmission, providing a distinct advantage. This suggests that the program's functionality can be applied globally and can be expedited under improved conditions, rendering it applicable to various currently deployed applications.

5. CONCLUSION

In this study, the author conducted an extensive examination of the data transmission process between two clients to establish session keys using the BB84 QKD protocol using IPv4. With the encoding and decoding of IPv4 using UTF-8, the data transfer from the server to the client can be sent and received effectively. Furthermore, the author delved into the utilization of quantum gates, particularly Hadamard and Pauli-X gates, for the manipulation of input qubits within the protocol. The QI platform was employed for the generation of qubits and the measurement of their states during the transmission. The findings regarding the performance of BB84 reveal that when the server is set up, and the key size increases to 4000 bits, the process of sending module takes 16.215 sec, the transfer module takes approximately 5.2 hours, the receive module takes 1.257 sec to finish the process for the final session key share. The results is improved speed, significant enhancements, and increased security achieved when implementing BB84 QKD on a quantum computer system. This approach has demonstrated its efficacy and practicality in bolstering system security through QKD protocols. This study demonstrates the potential of integrating QKD solutions to enhance the security of the wired and wireless network, providing a valuable resource for researchers and scientists to develop in the future.

ACKNOWLEDGEMENTS

The authors thank Academy of Cryptography Techniques for supporting this work. This work was supported by the Minister of Education and Training (MOET), grant no. B2025-GHA-02.

REFERENCES

[1] Y. Xiao, X. S. Shen, and D.-Z. Du, Eds., *Wireless Network Security*. in Signals and Communication Technology. Boston, MA: Springer US, 2007, doi: 10.1007/978-0-387-33112-6.

[2] S. Wehner, "Quantum network technology," *Open Access Government*, vol. 39, no. 1, pp. 276–277, 2023, doi: 10.56367/oag-039-10952.

[3] D. Conner, "*Cryptographic techniques*," Edn, vol. 41, no. 2, pp. 57–68, 1996, doi: 10.1007/978-3-030-74524-0_1.

[4] K. G. Paterson and A. K. L. Yau, "Cryptography in Theory and Practice: The Case of Encryption in IPsec," *Advances in Cryptology-Eurocrypt 2006*, vol. 4004, pp. 12–29, 2006, doi: 10.1007/11761679_2.

[5] V.-H. Le, N.-Q. Luc, T. T. Dao, and Q.-T. Do, "Building an Application that reads Secure Information Stored on the Chip of the Citizen Identity Card in Vietnam," *Engineering, Technology & Applied Science Research*, vol. 13, no. 1, pp. 10100–10107, February 2023, doi: 10.48084/etasr.5531.

[6] N. Q. Luc, T. T. Nguyen, D. H. Quach, T. T. Dao, and N. T. Pham, "Building Applications and Developing Digital Signature Devices based on the Falcon Post-Quantum Digital Signature Scheme," *Engineering, Technology & Applied Science Research*, vol. 13, no. 2, pp. 10401–10406, Apr. 2023, doi: 10.48084/etasr.5674.

[7] L. Gyongyosi, L. Bacsardi, and S. Imre, "A Survey on Quantum Key Distribution," *Infocommunications Journal*, vol.11, no. 2, pp. 14–21, 2019, doi: 10.36244/ICJ.2019.2.2.

[8] L. Cohen, "Unitary Transformation," in *The Weyl Operator and its Generalization*, Basel: Springer Basel, pp. 85–90, 2013, doi: 10.1007/978-3-0348-0294-9_7.

[9] S. Axler, "Hilbert Spaces," in *Graduate Texts in Mathematics: Measure, Integration & Real Analysis*, San Francisco, CA, USA, pp. 211–254, 2020, doi: 10.1007/978-3-030-33143-6_8.

[10] B. B. Poojary, "Origin of Heisenberg's Uncertainty Principle," *American Journal of Modern Physics*, vol. 4, no. 4, pp. 203-211, July 2015, doi: 10.11648/j.ajmp.20150404.17.

[11] M. Sarvaghad-Moghaddam and M. Zomorodi, "A general protocol for distributed quantum gates," *Quantum Information Processing*, vol. 20, no. 8, p. 265, August 2021, doi: 10.1007/s11128-021-03191-0.

[12] T.-T. Nguyen, T.-L. V. Khac, and N.-Q. Luc, "Simulation of the BB84 Quantum Key Exchange Protocol," in *2023 15th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 1–4, October 2023, doi: 10.1109/KSE59128.2023.10299471.

[13] D. R. I. M. Setiadi and M. Akrom, "Hybrid Quantum Key Distribution Protocol with Chaotic System for Securing Data Transmission," *Journal of Computing Theories and Applications*, vol. 1, no. 2, pp. 188–200, December 2023, doi: 10.33633/jcta.v1i2.9547.

[14] S. Basu and S. Sengupta, "Modified BB84 Protocol Using CCD Technology," *Journal of Quantum Information Science*, vol. 6, no. 1, pp. 31–38, March 2016, doi: 10.4236/jqis.2016.61004.

[15] G. Aubrun, L. Lami, C. Palazuelos, and M. Plávala, "Entanglement and Superposition Are Equivalent Concepts in Any Physical Theory," *Physical Review Letters*, vol. 128, no. 16, p. 160402, April 2022, doi: 10.1103/PhysRevLett.128.160402.

[16] "Demonstrations of quantum entanglement earn the 2022 Nobel Prize in Physics," *Physics Today*, vol. 2022, no. 10, October 2022, doi: 10.1063/pt.6.1.20221004a.

[17] S. Reddy M., S. Mandal, and C. Mohan B., "Comprehensive Study of BB84, A Quantum Key Distribution Protocol," *International Research Journal of Engineering and Technology (IRJET)*, vol. 10, no. 3, pp. 1023–1034, March 2023.

[18] L. Ma, A. Mink, and X. Tang, "High Speed Quantum Key Distribution Over Optical Fiber Network System," *Journal of Research of the National Institute of Standards and Technology*, vol. 114, no. 3, pp. 149-177, June 2009, doi: 10.6028/jres.114.010.

[19] S.-K. Liao *et al.*, "Satellite-Relayed Intercontinental Quantum Network," *Physical Review Letters*, vol. 120, no. 3, p. 030501, January 2018, doi: 10.1103/PhysRevLett.120.030501.

[20] J. Yin *et al.*, "Entanglement-based secure quantum cryptography over 1,120 kilometres," *Nature*, vol. 582, no. 7813, pp. 501–505, June 2020, doi: 10.1038/s41586-020-2401-y.

[21] G. Guarda *et al.*, "BB84 decoy-state QKD protocol over long-distance optical fiber," in *2023 23rd International Conference on Transparent Optical Networks (ICTON)*, pp. 1–4, July 2023, doi: 10.1109/ICTON59386.2023.10207397.

[22] M. Pivk, "Quantum key distribution," in *Lecture Notes in Physics*, vol. 797, pp. 23–47, 2010, doi: 10.1007/978-3-642-04831-9_3.

[23] A. Aji, K. Jain, and P. Krishnan, "A Survey of Quantum Key Distribution (QKD) Network Simulation Platforms," in *2021 2nd Global Conference for Advancement in Technology (GCAT)*, pp. 1–8, October 2021, doi: 10.1109/GCAT52182.2021.9587708.

[24] M. Bakni and S. Hanbo, "A Survey on Internet Protocol version 4 (IPv4)," *WikiJournal of Science*, vol. 5, no. 1, pp. 1-29, January 2022, doi: 10.15347/WJS/2022.002.

[25] R. Wille, R. Van Meter, and Y. Naveh, "IBM's Qiskit Tool Chain: Working with and Developing for Real Quantum Computers," *Proceedings of the 2019 Design, Automation and Test in Europe Conference and Exhibition, DATE 2019*, pp. 1234–1240, March 2019, doi: 10.23919/DATE.2019.8715261.

[26] P. I. Hagouel and I. G. Karafyllidis, "Quantum computers: Registers, gates and algorithms," in *2012 28th International Conference on Microelectronics Proceedings*, pp. 15–21, May 2012, doi: 10.1109/MIEL.2012.6222789.

[27] M. S. Zubairy, "No-cloning Theorem and Quantum Copying," in *Quantum Mechanics for Beginners: With Applications to*

Quantum Communication and Quantum Computing, pp. 172–181, June 2020, doi: 10.1093/oso/9780198854227.003.0011.

[28] A. D. V and K. V, "Enhanced BB84 quantum cryptography protocol for secure communication in wireless body sensor networks for medical applications," *Personal and Ubiquitous Computing*, vol. 27, no. 3, pp. 875–885, June 2023, doi: 10.1007/s00779-021-01546-z.

BIOGRAPHIES OF AUTHORS

Tat-Thang Nguyen works at the National Agency of Cryptography and Information Security, no. 23 Nguy Nhu Kon Tum Street, Nhan Chinh Ward, Thanh Xuan District, Hanoi, Vietnam. He focused on the mathematical aspects of elliptic curves and their applications in cryptography. He can be contacted at email: thangnt@bcy.gov.vn.

Thanh-Toan Dao is an Associate Professor at the University of Transport and Communications, no. 3 Cau Giay Street, Lang Thuong Ward, Dong Da District, Hanoi, Vietnam. His group is focusing on the topics of the design and manufacture of electronic systems for flexible organic IoT, and hardware-based encryption. He can be contacted at email: daotoan@utc.edu.vn.

Nhu-Quynh Luc was born in Nam Dinh, Vietnam in 1983. He received his Bachelor's in Mathematics at Vietnam University of Science (VNU) in 2006, and the Master of Crypto-graphic Engineering at Academy of Cryptography Techniques, Vietnam. He has a Ph.D. in electronic materials science. He focused on the mathematical aspects of elliptic curves and their applications in cryptography. Currently, his research interests are organic materials and their use in micro-electronic fabrication. He can be contacted at email: quynhln@actvn.edu.vn.