Hybrid Quantum-Classical Key Distribution

Alharith A. Abdullah, Suadad S. Mahdi

Abstract: ***Quantum Key Distribution (QKD) has been developed over the last decade; QKD addresses the challenge of a securely exchanging cryptographic key between two parties over an insecure channel where there are two parties that simultaneously generate and share a secret key using the polarization of quantum states of light by applying the phenomena of quantum physics. The integration of QKD protocol with public key cryptography for securely exchanging the encryption/decryption keys is proposed and simulated, the simulation results evaluate the work of the existing and proposed protocol taking into account different measures. Finally, a short security analysis is given to show the difference between the proposed protocol and its counterparts.***

Keywords : ***Quantum Computing, Quantum Key Distribution (QKD), BB84 protocol, public key cryptography, Diffie-Hellman Algorithm.***

I. INTRODUCTION

Recently, the world has become a global marketplace through the rapid introduction of the Internet and the increasing number of people's use of the Internet in e-commerce, but this development in technology has led to the need for safety measures to protect important data from the unauthorized persons like an eavesdropper. In the process of protecting data from unauthorized access, many countermeasures have emerged such as: encryption, hiding information, firewalls, access lists, proxy application portals, and security protocols like SSL, TLS, etc. [1]. Cryptographically speaking, encryption is known as the procedure of sending (encoding) messages that carry information in a way only that can be accessed only by authorized parties, whereas the unauthorized cannot. There are two types of encryption algorithms, symmetric (private-keys) and asymmetric (public-keys) encryption. If the message's encryption and decryption has the same key, this is known as a symmetric algorithm, whereas if various keys are utilized for encrypting and decrypting, this is known as an asymmetric algorithm [2]. However, all encryption techniques will not be effective if the main distribution mechanism is weak because if the key is detected by the eavesdropper, then the ciphertext can be discovered easily [3].

In 1976, Whitfield Diffie and Martin Hellman studied this problem of key distribution and they developed an algorithm that was widely known as “Diffie-Hellman” algorithm, which is utilized for securely exchanging a key over an unprotected channel [4].

The Diffie-Hellman algorithm was the first public key algorithm that employed mathematics for creating a shared key for the sender and receiver by the use of a channel to communicate them. This is done by modular arithmetic and discrete logarithm. The cryptography of a public key, TLS, SSH, PGP, and other PKI systems use the Diffie-Hellman algorithm. [5].

The quantum key distribution (QKD) is regarded as one of the most interesting domains in quantum information to exploit the quantum physics laws to allow random keys exchange between two parties and these random keys are represented as qubits that can be used as keys for encryption and decryption [6]. The idea is to use techniques of quantum to assure the modification of any measurement to the status quantum bits (qubit) transmitted, and the sender as well as receiver of the quantum bits can detect this modification if there is a third party eavesdropping (Eve) [7].

Quantum key distribution protocol was proposed in 1984 by Charles H. Bennet and Gilles Brassard, it was later called BB84 [8]. It Utilized the uncertainty concept and no-cloning theorem to guarantee that the transmission of the key has not been eavesdropped or altered [9] [10]. This protocol uses four non-orthogonal polarized states ($0^{\circ}, 90^{\circ}, 45^{\circ}, 135^{\circ}$). Both Shor and Preskill proved that to be definitely safe, expediting safe communication between parties with no any pre-shared secret information.

In this work, we propose to the integration of the Diffie–Hellman protocol along with QKD-BB84 key agreement process to provide a security at a higher level for the key by compelling an attacker to cut off two totally unlike cryptosystems to access the key that is used in encryption/decryption algorithm. This proposed solution offers double security through methods of merging quantum (physical layer security) and classical (computationally hard to resolve) to provide high-level security and also provide inherits existing certifications from the conventional security scheme [11] [12] while increasing the security with the quantum-based cryptosystems.

The paper goes as the following plan: In Section 1 the conventional and quantum key distribution is highlighted in brief. Section 2 reviews a background of the papers that interested in this field. Section 3 presents the proposed integrated protocol with the simulation results in Section 4 separately.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author

Dr. Alharith A. Abdullah*, Information Technology, College of Information Technology, University of Babylon, Babil, Iraq alharith.khafaji@yahoo.com

Dr. Suadad S. Mahdi, College of Information Technology, University of Babylon, Babil, Iraq suadadsafaa@gmail.com

Lastly, the paper ends with the analysis of the security and the conclusions respectively in both Sections; 5 and 6.

II. RELATED WORKS

A new technology called QKD appeared for keeping important data during the process of transmission in a new communication medium. Therefore, several researches proposed to modify the BB84 protocol.

K. Wen and G.L. Long [13] checked a modification to the BB84 protocol equipped with a two-way conventional communication parallel texture refining protocol. This protocol of a modified key distribution is definitely safe with a 20%. as has a higher acceptable error rate. The main purposes from modification are increasing the length of the quantum key.

The Hybrid key was proposed by Amrin M. et al. [14] as a solution to the attack of MITM in BB84. It utilizes the algorithms of computational security and takes another algorithm for creating key from image algorithm. By utilizing these two keys, this proposed system does not offer an offline key establishment and development.

The integration of the mechanics of absolutely secure authentication from conventional cryptography with QKD was presented by Sufyan T. and Omer K. [15] in order to reduce the authentication cost. But this solution leads to lessen the competence of the proposed system.

Marcin N. and Andrzej R. [16] depicted a novel concept of the measures of security in QKD and they suggested a new concept of incompetence of security in QKD and a unique measure of security is defined. Two different levels of security were presented by the authors; one is basic and the other is advanced. This differential security division assists to choose the suitable security level for needs and requirements of specific end-users. However, the strength of the enhanced system against intrusion attacks was not tested by the authors.

Omer K. et al. [17] set the QKD-BB84 protocol as a real simulator and they showed the outcomes with different level configuration that shows the immediate effects on the time needed for key generation and length of the key obtained. Furthermore, it carries out this simulator on two different modes, with attack influences and without attacker.

In their study paper, Rupesh K. et al. [18] developed the simulation algorithm of BB84 on the classical computer by using C programming language as well as disordered sequence generator instead of random number generator in order to obtain a strong key.

III. THE PROPOSED WORK

Despite the difference between the protocols, each protocol needs to exchange many messages between two parties, and there are the key exchange protocols between two parties, whether traditional methods or quantum key distribution (QKD) system.

Diffie-Hellman is very common used protocol for key exchange. This protocol demands exchanging several messages between two endpoints to take part inevitable parameters to create a shared secret key via public channel.

The QKD-BB84 protocol operates in the same principle as Diffie-Hellman, where the two ends need to send key ID ($\otimes,\oplus$) to each other over an open public channel. This process requires several messages for the synchronization of keys on both ends.

Therefore, because of these similarities, the process of integration BB84 protocol and Diffie–Hellman protocol together can be easily done. By expanding Diffie–Hellman messages and including new parameters, like key IDs, to secure the key more. The workflow is as follows:

1. Alice can choose any number (N) that is input to detect the number of bits used and pass it to Random function to generate a number of the random bits (0 or 1) and equal to the input number. Also, Alice determine threshold that using later at the final step.

2. Passing N number to another function to generated random bases (rectilinear + or diagonal x) and keeping the bases generated in AliceBasesArray to create a polarization by using these generated bases. Then, these polarizations are sent to Bob, polarizations are (→=0°, ↑=90°, ↗=45°, ↖=135°). The condition of generated polarizations is as the following:

Depending on the random bit value generated by random function when the value is 1, 0 and it depends on the bases +, x.

When the base is +, that state depends on the random bit value as follows:

When bit value = 0, polarization state will be →.

When bit value = 1, polarization state will be ↑.

And when the basis is x, which state depends on the random bit value as follows:

When bit value = 0 polarization state will be ↗.

When bit value = 1, polarization state will be ↖.

3. When these 1, 2 steps done, we can send polarizations to Bob over quantum connection channel to measure it on Bob side and get the key.

4. Once polarization received, Bob begin measurement operation by generating a random. Also, keeping the generated bases in BobBasesArray. The measures the polarization do as the following:

When bases is + and polarization → return 0.

When bases is + and polarization ↑ return 1.

When bases is x and polarization ↗ return 0.

When bases is x and polarization ↖ return 1.

This operation repeats equally to the number of received polarizations. Then, each value that returned is kept in the RawKeyArray.

5. Each of Alice and Bob encoding random bases to (0,1) depending on the prior agreement between the two parties, as a following:

$$\oplus \Rightarrow 0$$

$$\otimes \Rightarrow 1$$

At the same time, Alice and Bob agree on the constants p where it represents a prime number and g is the generator over a public channel from a trusted third party.

6. Alice chooses her private key “a”. Then, calculates the public key A:

$$A = g^a mod\ p$$

7. Bob chooses his private key "b". Then, calculates the public key B:

$$B = g^b mod\, p$$

8. Over classical channel, Alice and Bob exchange the coding bases as key ID and public key with authentication (by using the one-way function sha256) with each other.

9. After decoding the bases at Alice and Bob. Then, each of them verifies the validity of the authentication is valid or nor. If authentication is valid. Then, Alice and Bob compared bases between them and each corresponding bases is considered a correct case and keeps the bit that matches it in array location. And if authentication is not valid the connection closed.

10. Each of Alice and Bob measure the key length percentage to compare it with the threshold required. If the key length percentage is equal or bigger than the threshold the process is success and if the key length percentage is less than the wanted percentage the whole process will repeat until we get the wanted percentage.

Finally, Alice and Bob are calculating classical key (based on the laws of algebra which are the same on both ends) and quantum key. Then occurs the combination of both keys through XOR (addition module 2) in order to use them together in symmetric encryption algorithms. The processes of the proposed system is shown in Figure 1.

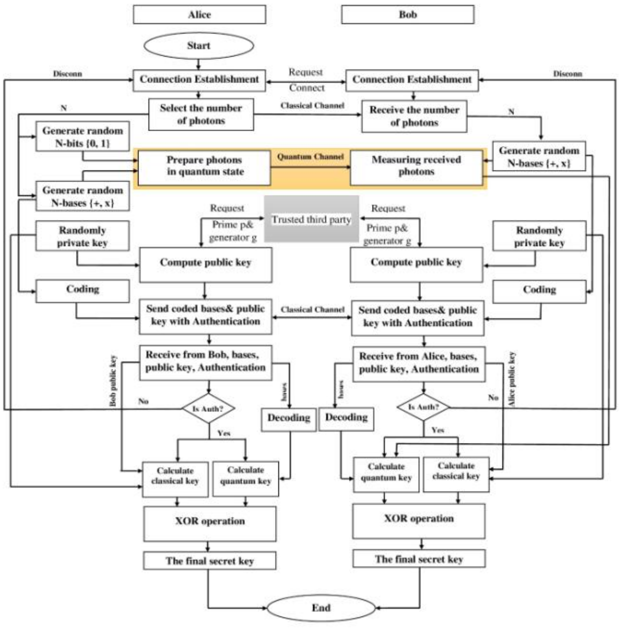

Fig. 1 Flowchart for the proposed protocol

IV. SIMULATION AND RESULTS

To prove the strength of the proposed work, first it is required to simulate the protocol BB84 and then simulate the proposed system, and then compare the results of simulation for both of them. In this work, the simulation environment has been developed by using python programming language and IDE is PyCharm based on Windows 10 as running system. Core i5, processor (2.40 GHz) associated with 4GB RAM as a Hardware is used for the performance of the simulation.

Figure (2) demonstrates the experimental results of the simulator BB84 according to the hardware specified, with pumping initial qubits at different length. The figure below shows the simulator starts from 128 (an initial qubit) up to 1024 qubits and 45% bit error rate allowed. The authentication cost is commonly known as the variance of the results obtained as well as and the lost qubits, as the figure shows.

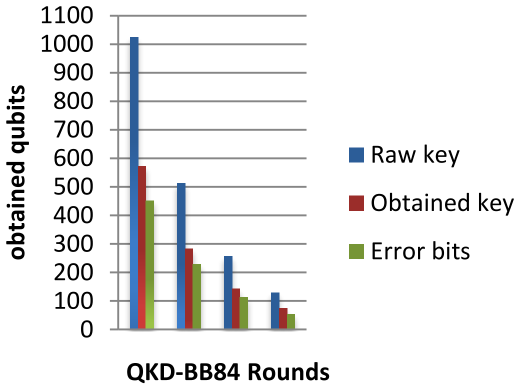

Figure (2): The obtained key with a 0.45-bit error rate

While Figure (3) shows the results in experiment of the simulator proposed system depending on the same configuration levels and hardware specification with implementing Diffie-Hellman protocol using a python library called pyDH 0.1.2 and 2048-bit MODP group [19].

This group is given an id 14:

The prime is: 2^2048 - 2^1984 - 1 + 2^64 * { [2^1918 pi] + 124476}

Its hexadecimal value is:

FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1

29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD

EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245

E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED

EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE45B3D

C2007CB8 A163BF05 98DA4836 1C55D39A 69163FA8 FD24CF5F

83655D23 DCA3AD96 1C62F356 208552BB 9ED52907 7096966D

670C354E 4ABC9804 F1746C08 CA18217C 32905E46 2E36CE3B

E39E772C 180E8603 9B2783A2 EC07A28F B5C55DF0 6F4C52C9

DE2BCBF6 95581718 3995497C EA956AE5 15D22618 98FA0510

15728E5A 8AACAA68 FFFFFFFF FFFFFFFF

The generator is: 2.

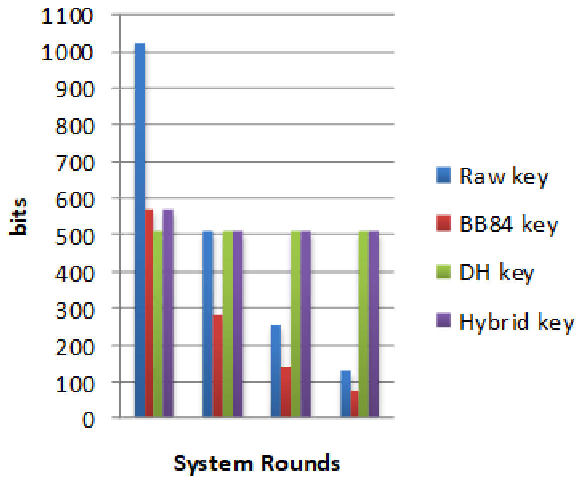

Figure (3): Rounds for generating the hybrid key

Figure (4) clarifies that the efficiency of our proposed protocol will be more efficient compared with the BB84 protocol. It is clear that the length of the key will be increased when the BB84 protocol generates key less than 512 bits, in addition to increasing randomness rates for the final key.

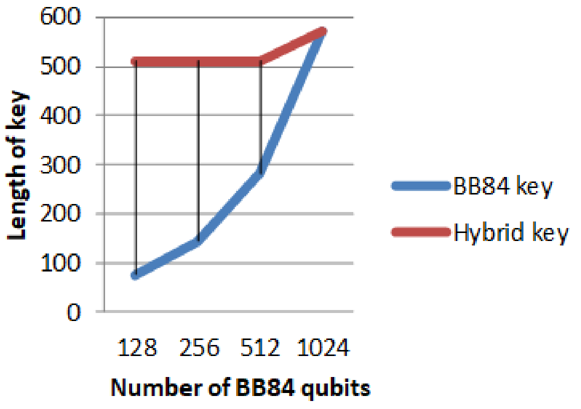

Figure (4): Efficiency for BB84 key and Hybrid key

In addition, for examining and evaluating the randomness rate for final key based on the p-values, NIST suites have been implemented. The NIST suite test is used by calculating a (p-value). When a p-value is equal to (1), this means the sequence has perfect randomness. When a p-value is equal to (0), this means the sequence appears to be completely non-random.

We execute the NIST by the use of a binary sequence that consists of different key bits. Table (1) shows an execution for 4 times for the BB84 protocol, it shows calculating the randomness of keys in each time by using tests Frequency (Monobit), frequency within a block, Runs, and Cumulative sums (forward),and in Table (2) we execute the same tests randomness on the proposed system. We noticed the p-value in most tests is more near to (1) in the proposed system compared with the BB84 protocol.

V. SECURITY ANALYSIS OF THE PROPOSED WORK

. This section is devoted to discuss the features of the security of the proposed protocol. Particularly, the protocol analysis is presented with respect to the features of known-key security, forward secrecy, plus replay attack resilience.

1) Known-key security: is a feature of key agreement protocols, in the proposed protocol we are interested in determining the number of the parameters around the final keys. If the opponent knows one of the public keys this does not mean he can calculate the final key because he needs to know the private key for both parties in addition to the bases IDs for both parties and the measurement of the photon state transmitted via the quantity channel. This is very difficult because of the difficulty of solving the discrete logarithm to which no competent method for computing on conventional computers is known [20]. As well as the impossibility of copying polarized photon state and measure based on quantum laws. And this makes it impossible for the final key to be reached.
2) Forward secrecy: this feature functions that a user's session key will not be compromised, for instance, if one of the endpoint's private key is compromised, then the adversary is still incapable of determining the final key used, and this feature has been achieved in the proposed protocol mainly through integrating BB84 with Diffie-Hellman protocols. Also, this feature depends on the security of the third party and methods of verification before sending basic parameters.
3) Replay attack resilience: it is a feature of (specific) secure in the replay attacks where a valid message is captured and maliciously replicated later. The proposed protocol protects against the replay attacks through authentications which are on two levels: the quantum channel level, by exploiting the physical properties of the photon, and the classical channel level through using the one-way function (SHA-256) which has proven to be safe against quantum computer attacks [21].

In addition, the trusted third party generates the prime numbers and the generator in each process key exchange in different numbers, so it will generate a key that is different in every process so the attacker does not benefit from the old parameters.

Table- I: Results of BB84 execution for 4 times the NIST test

	Used Bit Number	Key Length	Frequency (Mono-bit)	Frequency within a Block	Runs	Cumulative sums (forward)
Key 1	128-bit	512-bit	0.5958	0.6701	0.6319	0.5518
Key 2	256-bit	512-bit	0.5987	0.0119	0.6693	0.5701
Key 3	512-bit	512-bit	0.1372	0.2809	0.8003	0.2444
Key 4	1024-bit	572-bit	0.2097	0.3039	0.9885	0.2241

Table- II: Results of proposed system execution for 4 times the NIST test

	Diffie-Hellman Groups	Number Bit Used in BB84	Final Key Length	Frequency (Mono-bit)	Frequency within a Block	Runs	Cumulative sums (forward)
Key 1	2048-bit Group	128-bit	128-bit	0.8596	0.5151	0.6319	0.4302
Key 2	2048-bit Group	256-bit	256-bit	0.8005	0.0590	0.75138	0.5031
Key 3	2048-bit Group	512-bit	512-bit	0.5958	0.4578	0.8671	0.2894
Key 4	2048-bit Group	1024-bit	572-bit	0.5582	0.7529	0.6023	0.8892

VI. CONCLUSION AND DISCUSSIONS

The main problem with symmetric encryption is the process of distributing the key securely and reliably. The traditional methods used to solve this problem depends on the computational complexity and the time needed to solve it. So, it is not impossible getting the secret key by attacker. Therefore, we propose an integrated system to generate the key and exchange it safely by providing reliability by exploiting the physical properties of the photon in order to achieve this, in addition to the strength and randomness of the quantum key and providing the computational complexities and certificates of traditional systems to become an integrated hybrid key.

In this paper, simulations of the proposed system were implemented by using Python language, and the results showed the efficiency of the proposed system for key exchange as well as the easy execution of this work in practice for achieving better results.

REFERENCES

1. Alfred J. Menezes et al., Handbook of Applied Cryptography, 2001.
2. B.Dan and S.Victor , A Graduate Course in Applied Cryptography, 2015.
3. B. Chris, “Vladimir Aleksandrovich Kotelnikov: Pioneer of the sampling theorem, cryptography, optimal detection, planetary mapping,” IEEE Communications Magazine, 2009.
4. K.Brian et al., “SIDH on ARM: Faster Modular Multiplications for Faster Post-Quantum Supersingular Isogeny Key Exchange Protocol on ARM,” 2016.
5. Sheffer Y and Fluhrer S, “Additional Diffie-Hellman Tests for the Internet Key Exchange Protocol Version 2 (IKEv2),” 2013.
6. Ali Ibnun Nurhadi et al., “Quantum Key Distribution (QKD) Protocols: A Survey,” IEEE Communications Magazine, 2018.
7. C.H. Bennett and G. Brassar, “Quantum cryptography: Public key distribution and coin tossing,” Theoretical Computer Science, 2014.
8. Charles H. Bennett et al., “Experimental Quantum Cryptography,” 1991.
9. William K.Wootters and Wojciech H.Zurek, “The no-cloning theorem,” Physics Today, 2009.

10. Vasileios Mavroeidis et al., “The Impact of Quantum Computing on Present Cryptography,” International Journal of Advanced Computer Science and Applications, 2018.
11. Nino Walenta et al., “Practical aspects of security certification for commercial quantum technologies,” Electro-Optical and Infrared Systems: Technology and Applications XII; and Quantum Information Science and Technology, 2015.
12. Alejandro Aguado et al., “Hybrid Conventional and Quantum Security for Software Defined and Virtualized Networks,” Optical Society of America, 2017.
13. Mohsen Sharifi et al., “A Simulative Comparison of BB84 Protocol with its Improved Version,” JCS&T , 2007.
14. AmrinBanu M. Shaikh and Parth D. Shah, “BB84 and Identity Based Encryption (IBE) Based A Novel Symmetric Key Distribution Algorithm,” Fifth International Conference on Advances in Recent Technologies in Communication and Computing, 2013.
15. Sufyan T. Faraj Al-Janabi and Omar Kareem Jasim , “Reducing the Authentication Cost in Quantum Cryptography,” 2011.
16. Marcin Niemiec and Andrzej R. Pach, “The measure of security in quantum cryptography,” Communication and Information System Security Symposium , 2012.
17. Omer K. Jasima et al., “Quantum Key Distribution: Simulation and Characterizations,” Procedia Computer Science, 2015.
18. Rupesh Kumar Sinha et al., “Quantum Key Distribution: Simulation of BB84 Protocol in C,” International Journal of Electronics, Electrical and Computational System, 2017.
19. Kivinen and Kojo, “More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE),” 2003.
20. Taher Elgamal, “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” IEEE, 1985.
21. Lily Chen et al., “Report on Post-Quantum Cryptography,” National Institute of Standards and Technology , 2016.

AUTHORS PROFILE

Dr. Alharith A. Abdullah received his BS degree in Electrical Engineering from Military of Engineering College, Iraq, in 2000, his MS degree in Computer Engineering from University of Technology, Iraq, in 2005, and his PhD degree in Computer Engineering from Eastern Mediterranean University, Turkey, in 2015. His research interests include Security, Network Security, Cryptography, Quantum Computation and Quantum Cryptography.

Dr. Suadad Safaa, received her B.S. degree in 2016 from the College of Information Technology (IT), University of Babylon, Iraq. She's currently a master's student in the Department of Information Networks. Her main interests include software-defined networks, cryptography and steganography.

.
.