Multiparty Quantum Cryptography with Block Cipher RC6 (B-MQKD)

Iqtidar Zohair
Faculty of Mathematics and Computer Science
University of Kufa
Kufa, Iraq
Iqtidarz.alshammary@student.uokufa.edu.iq

Salah Albermany
Faculty of Mathematics and Computer Science
University of Kufa
Kufa, Iraq
Salah.albermany@uokufa.edu.iq

Received Oct. 21, 2018. Accepted for publication Mar. 26, 2019

DOI : http://dx.doi.org/10.31642/JoKMC/2018/060201

Abstract

The proposed B-MQKD method uses a quantum key distribution to generate a key and provide an authentication amongst many parties. It uses concepts of block cipher by using RC6 algorithm for encryption. The mix concept in B-MQKD (quantum and block cipher) gives the algorithm more authentication, randomness, and security which make it difficult to find the original message by attackers. In this paper using BB84 among three parties make it possiple to give the key to many users in a secret way.

Keyword- Multiparty QKD; RC6 block cipher

I.INTRODUCTION

The design of new proposed B-MQKD (Multiparty Quantum Key Distribution with block cipher) as encryption system used the principle of protocol BB84 with multi parties to generate a secret shared key, provide an authentication among these parties that communicate together and randomness in generating a key. See figure (1.1) for the general diagram of the proposer.

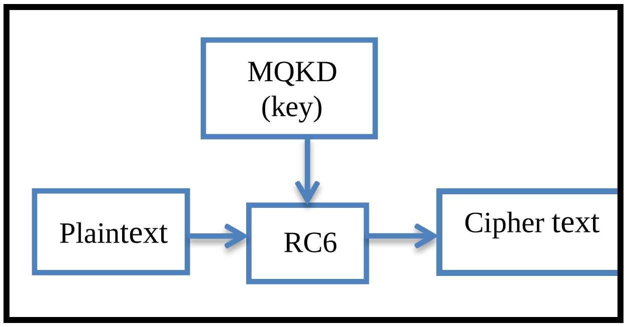

Figure (1.1) General simple diagram

II. BB84 Protocol

The first protocol for quantum cryptography was proposed in 1984 by H. Bennet and Gilles Brassard. They started from Stephen Wiesner‟s work "Conjugate Coding" and then developed this work to key distribution protocol using photon‟s polarization [1]. The **polarization** states represent both orthogonal bases for linear polarization (+) and diagonal bases for diagonal polarization (×). They used the following notation in which (—) denoted photon in a vertically polarized state or degree, (|) denoted a photon in horizontal polarized state or , (/) denoted a photon in 45 degree polarized state, and

(\ ) denoted a photon in 135 degree polarized state

The

and degree was measured as bit „ " and and

degree was measured as bit „ " [2][3], see Figure (2.1).

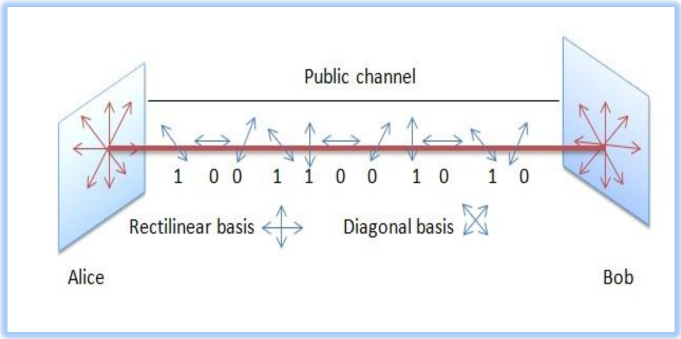

Figure (2.1) Rectilinear and diagonal Bases

In this protocol, the two participants, traditionally called Alice and Bob, wish to agree on a secret key in which no eavesdropper, traditionally called Eve, can obtain significant information. Alice randomly generates a sequence of bits and sends it with different polarization bases (measure as bit) to Bob via a quantum channel [4]. When Bob receives this sequence, also called "raw key", he chooses the basis of each qubit, his measure is either like Alice basis or different from Alice basis and sometimes he does not measure anything. Bob announces his bases through a public channel then only the bit that is similar to Alice bases is used as a secret key and the rest that is not similar bit or not measured by Bob is discarded. This short bit is called "sifted key" [5], see figure (2.1).

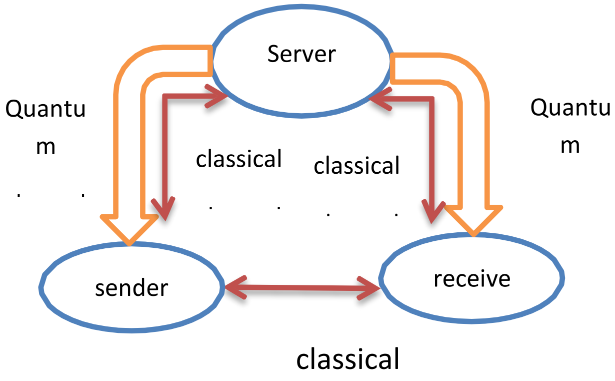

Figure (2.2) three users communicate

III. GENERATE A KEY USING BB84 FOR MULTIPARTY

The protocol BB84 can be used among three parties: server, sender, and receiver to generate a key 128-bit, and provide an authentication amongst these three parties. The three parties communicate together for sending and receiving the sequence of bases and polarization via a quantum channel (e.g. Free space or fiber optic), and the comparison of the bases amongst the parties via a public or classical channel (e.g. Internet), see figure (2.1).

The QKD has three sequences. They can be generated: sequence of bits (...............................), sequence of bases also called filters (× + + × × + + …), and sequence of polarize (

. Server randomly generates a sequence of bits with length n and a sequence of basis for each bits, then determines the polarize for each basis. When server sends his bases to sender, the latter determines his polarize too.

The server and the sender agree on the threshold, supposed TH =20% (where TH is a threshold used by sender and server), When they compare their polarize, they compute the probability for correct polarize and if it is more than or equal TH then they convert to binary bits and it must be with length 128 to represent a secret shared key.

The receiver agrees with the sender on ID, which sends it to server when he/she needs to get the correct polarize from server to convert it to a secret key with 128-bit.

```
Algorithm (1): Algorithm for generating a key ( by using Multi-party Quantum Key Distribution (MQKD)

Input: TH, sequence of bits S

Output:          //secret shared key with 128 bit

Phase one

  1. server generate a random of bits sequence S.
  2. server generate a sequence of bases (sr_ bases) and sequence of polarize(sr_ polarize) for each bit in S
  3. server send sr_ bases to sender.
  4. sender receives the sequence of bases with length L and determine the sequence of polarize (sn_ polarize) randomly.

Phase two

     for j← to length (S)
  6. if sn_ polarize = sr_ polarize
        i← i+
  7. end for j
  8.        –
  9. if >= Th
       return(
  10.if  <  Th Go
       to step 3

Phase three

       Id ← sn_ polarize(  :8)
  12. sender send Id to reciever
  13. receiver ask a key from server and send Id
  14. if Id is true
  15. server send sn_ polarize to receiver
  16.  receiver convert sn_ polarize to a key
```

IV. ENCRYPTION PROCESS IN B-MQKD

This section will explain the algorithm of B-MQKD method (3.1) for encryption process. Input of encryption algorithm is a message M and the output is the encryption message C. To encrypt message will use RC6 encryption algorithm and use the key that results from key expansion algorithm in RC6 [6], where the user key is generating by using BB84 protocol for multiparties. We will divide the message into many blocks each with 128- bit block if the block not equal 128 it can be padding by usin character „X‟

Algorithm(2): Algorithm for B-MQKD method for encryption process

```
Input: M, , λ, n (number of state), k, 𝛼, β
Output: C //cipher text
        1. To Get key ( ) use algorithm (1) to generate and make expansion by using RC6 key expansion algorithm
        2. for j → to |M|
        3. = block(β -bit) // where β = 28
    4. if length( ) != β
            = ( "X")     //convert X to binary and add to m
    5. end
                    β|| )
     7.   C =
     8. return (C)
```

V. DECRYPTION PROCESS IN B-MQKD

The decryption process begins when the cipher text is received by other two parties. Use RC6 decryption algorithm to decrypt the cipher blocks and the key that results from key expansion algorithm in RC6.

Algorithm (3): Algorithm for decryption process in B-MQKD method

```
Input: cipher text C ( ),
Output: plaintext M (
        1. for j → to |C|
        2.    =       ||            // using RC6 block sipher decryption algorithm
        3. M= ( … )
    4. end for j
    5. return (M)
```

VI. HAMMING DISTANCE

The Hamming distance H is defined as two strings of the same length. If we have two strings s and t, then H (s , t) is the number of places in which the two strings differ. This means that it measures the different characters. More clearly, the distance between two strings (or blocks) A and B is

∑| Ai Bi |.

For example 0101 and 0110 have a Hamming distance of two and "Butter" and "ladder" are four characters apart [7][8].

For using hamming distance to measure a randomness, there are many algorithms used in different ways. For example in [9] and [7], we can use another way to measure a randomness of B-MQKD, by using hamming distance for 100 run as mentioned before, the result of hamming distance differed, this because the randomness in B-MQKD see figure (5.1)

VII. TYPE OF ATTACK CRYPTANALYSIS

This section explains many methods of cryptanalysis, and illustrates effectively the methods against three

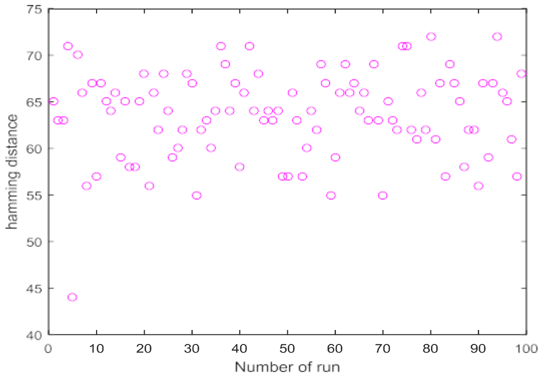

Figure 5 .1 Hamming distance result for two

attacks.

A. Exhaustive search attack (Brute force attack)

In this attack, the attackers attempt to get a key that is used in decrypting a message, they still search in all possible keys, so the possible key in exhaustive search attack is , where n is a length of key. Because this method depends on the key length, it takes a long time where the key is longer. In the new proposed algorithm B-MQKD, the length of the key is 128 bit or more thus the possible attempt to get a key is .

B. Man-In-The-Middle attack (MIM or MITM)

In this type of attack, the attackers try to eavesdrop on the communication between two users communicating with each other through a network. B-MQKD algorithm is considered secure against this type of attack because it is based on a secure channel (quantum channel) to transmit the key and existing the server that provides agreement between users.

VIII. CONCLUSION

B-MQKD provides a new encryption system depending on the randomness principle. This means that one plaintext can give many different ciphertexts. This randomness comes from the random generation of a key using BB84 protocol among many parties. Also used RC6 block cipher algorithm, which provides a simplicity by keeping cipher structure simple.

REFERENCES

[1] C. H. Bennett and G Brassard, “Quantum cryptography: Public key distribution and coin tossing,” *Theor. Comput. Sci.*, vol. 560, no. P1, pp. 7–11, 2014.

[2] G Mogos and G Radu, “QKD PROTOCOLS – SOFTWARE IMPLEMENTATION BENNET-BRASSARD vs BRUSS,” vol , no , pp 81–84, 2015.

[3] A. Goneid and A. M. Abbas, “Enhancement of Error Correction in Quantum Cryptography BB84 Protocol Enhancement of Error Correction in Quantum Cryptography BB8 Protocol,” no July, pp 0–12, 2014.

[4] B Valiron, “Quantum computation: A tutorial,” *New Gener. Comput.*, vol. 30, no. 4, pp. 271–296, 2012.

[5] I. Journal, E. Science, and I. X. I. May, “Study of Bb8 Protocol Using Qkd Simulator,” vol I, no Xi, pp –444, 2015.

[6] R. Rivest, M. J. B. Robshaw, R. Sidney, and Y L Yin, “The RC6 Block Cipher,” *First Adv. Encryption …*, 1998.

[7] M. X. He, S. V. Petoukhov, and P. E. Ricci, “Genetic code, hamming distance and stochastic matrices,” *Bull. Math. Biol.*, vol. 66, no. 5, pp. 1405–1421, 2004.

[8] N. Hakiem, M. U. Siddiqi, and A. U. Priantoro, “Randomness Test of Cryptographic One- to-many Reversible Mapping for IPv6 address generation,” Journal of Theoretical and Applied Information Technology 69(3). November, 2014.

[9] A. E. Belfedhal and K. M. Faraoun, “Building secure and fast cryptographic hash functions using programmable cellular automata,” *J. Comput. Inf. Technol.*, vol. 23, no. 4, pp. 317–328, 2015.