Extended BB84 Protocol Using Lucas Series and Identity Based Encryption

AmrinBanu M. Shaikh[1(✉)] and Parth D. Shah[2]

[1] U. & P.U. Patel Department of Computer Engineering,
Chandubhai S. Patel Institute of Technology,
CHARUSAT, Changa, Anand 388421, India
amrinbanushaikh.ce@charusat.ac.in
[2] Department of Information Technology,
Chandubhai S. Patel Institute of Technology,
CHARUSAT, Changa, Anand 388421, India
parthshah.ce@charusat.ac.in

Abstract. In 1984 Bennett and Brassard proposed a Quantum Key Distribution (QKD) protocol known as BB84 protocol to distribute a random and frequently changed key using quantum mechanism. A major problem in this Protocol is to prove authentication. One of a solution of this problem has already been proposed in [4]. But in presence of Hardware Fault or Interception, above protocol is not applicable. In This paper Proposed System, Key Distillation has been added to overcome Hardware Fault or Interception with minor changes were not available in [4] and was available in original algorithm. It may increase performance of the Proposed System.

Keywords: Quantum key distribution (QKD) · BB84 · Identity key · Hybrid key

1 Introduction

BB84 Protocol is having mainly three steps: Raw Key Exchange, Key Shifting and Key Distillation. In Raw key exchange, Alice chooses randomly selected bit value with randomly selected bases from rectilinear or diagonal bases which results in four quantum states which are exchanged with Bob through quantum channel. The only way for Bob to derive any information from the incoming quantum states is to measure them against a randomly selected sequence of bases of his own. In Key Shifting if Alice/Bob selects the same base, which was used for decoding/encoding, then the result is determined to be correct. When the bases are different, then the result of this measurement is in deterministic [6]. After Key Shifting, key should be free of errors but it is possible only if no interception or no Hardware Fault is there. Key Distillation is consists of two steps. The first step corrects all the errors in the key, by using a Classical Error Correction protocol to precisely estimate the actual error rate. With this error rate, it is possible to accurately calculate the amount of information the eavesdropper may have on the key. The second step is called privacy amplification which compresses the key by an appropriate factor to reduce the information of the eavesdropper. The compression factor depends on the error rate. The higher the error rate, the more information an

S.C. Satapathy and A. Joshi (eds.), *Information and Communication Technology for Intelligent Systems (ICTIS 2017) - Volume 1*, Smart Innovation, Systems and Technologies 83, DOI 10.1007/978-3-319-63673-3_10

eavesdropper might have on the key and the more it must be compressed to be secure [1]. (For more information on how BB84 protocol works please refer [2, 3].) To calculate error rate Alice will repeatedly picks up a random position and check the bit value stored in that position with Bob then both will calculate error rate. Then after checked positions and their bit values are going to be deleted. But still BB84 is not able to prove authentication so it is possible that Alice is communicating with Eve. For that, Key Distillation Step should be removed. To know how BB84 protocol works without Key distillation step please refer [4]. But it is possible that because of Hardware Fault or Interception, keys at both ends are not same. In that case [4] is not applicable. For that Key Distillation step is in need. Outcome of above written introduction is, now we have to keep original BB84 protocol by considering another aspect. Proposed system is a combination of BB84 protocol [2] and "BB84 and Identity Based Encryption (IBE) based a novel Symmetric Key Distribution Algorithm" [4] with some changes which may improve performance. And here, rather than selecting random positions for Key Distillation step Lucas series [5] is used which is as same as Fibonacci series but the starting two numbers are chosen by Alice and Bob. The Advantage of Lucas series is that instead of sharing all random position numbers, Alice or Bob have to share only starting two numbers.

2 Proposed Algorithm

Step: 1 Raw Key Exchange

Alice Encodes randomly selected bit value by randomly selecting bases (Rectilinear or Diagonal) and generate polarized photon states and send to Bob through Quantum Channel and keep base-value combination with her in digital memory.
Bob measures the polarized photons in one of two set of bases (Rectilinear or Diagonal).

- If Bob selects same base as Alice result will be correct for that bit
- Else result will be random for that bit.

[Keep record for result and bases used in digital memory.]

Step: 2 Key Shifting

Bob tells Alice which Bases he used where bases are encrypted with public key of Alice via authenticated classical channel.
Alice tells Bob Which Bases she used where bases are encrypted with public key of Bob via authenticated classical channel.
Both will compare their Bases and keep only those values where they both used the same base and discard others let N be the number of remaining bits.

[Known as raw key named k1]

Step: 3 Transfer Challenge

Alice will send (nonce encrypted with k1A (hash value of k1 (at Alice side)), known as challenge) and encrypt this message with KB (public key of Bob) and send it to Bob.
Bob will decrypt the message with his private key and decrypt the challenge with k1B (hash value of k1 (at Bob side)) and send message back to Alice by encrypting it with KA (public key of Alice).
Alice will verify whether they both are having same k1 by verifying the received challenge with sent challenge. If yes then they will execute step:7 otherwise they will continue by executing step: 4.

Step: 4 Error Estimation

Alice and Bob will confidentially choose two numbers, and that will be starting numbers of Lucas Series.
Lucas series values are behaving like position numbers.
Alice will calculate the MAC (Message Authentication Code) by inserting position number, bit value and a secret key [Assume Secret key is confidentially chosen] and transfer it to Bob until Lucas series, position number exceed then the length of k1 and Bob will verify whether error is there or not and communicate where it didn't match and discard all Lucas Series Positioned bits.
Alice and Bob both are going to calculate error rate e. If error rate is higher than threshold they have to abort otherwise continue.

[Assume error rate e = 0.2 because it could be possible that it didn't get the eroded position because proposed system is using Lucas series.]

Step: 5 Reconciliation

It is a parity based protocol. To estimate error it generates a block and checks parity for that block.
Initial Block size K0 = $\left(\frac{1}{e}\right) + \left(\frac{1}{(4+e)}\right)$
Alice and Bob both are going to Check parity for that block and if it didn't match discard the whole block otherwise keep it as it is.
Increase Block size by K i + 1 = 2*Ki
Repeat until Block Size K i + 1 exceeds $\frac{1}{4}$ of all bits (length of key at step 1).

[After this step Alice and Bob both will have one key k2. If length (k2) < length (message) then abort.] [3]

Step: 6 Transfer Challenge

Alice will send (nonce encrypted with k2A (hash value of k2 (at Alice side)), known as challenge) and encrypt this message with KB (public key of Bob) and send it to Bob.
Bob will decrypt the message with his private key and decrypt the challenge with k2B (hash value of k2 (at Bob side)) and send message back to Alice by encrypting it with KA (public key of Alice).
Alice will verify whether they both are having same k2 by verifying the received challenge with sent challenge. If yes then they will continue to communicate by executing step: 7 otherwise they will abort [4].

Step: 7 Generate Identity Key

Assume Bob has already exchange his image (photo) to Alice by entity Authentication. Here, Bob can share his image after or before Quantum key established.
Alice and Bob both will generate Image key (Ik) from image [4].

[This step could be performed first]

Step: 8 Generate Hybrid key

Both will repeat Ik until its length becomes same as k1(or k2) length and find final key k by XOR operation.

Step: 9 Encryption

Alice will Encrypt message using k and send it to Bob. Here, Alice encrypt message using public key of Bob.

Step: 10 Decryption

Bob knows his private key, can only decrypt the message then after Alice and Bob both will Exit.

Assumptions
Private keys of Alice and Bob are not known to Eve.
Identity (Image) is confidentially shared.
Lucas Series starting two numbers are confidentially shared.
Single polarized photons are transferred.
Length of Raw Key Exchange = 8 Message length in bits.*

3 Proposed Algorithm Flow

Below Figure shows flow of Proposed Algorithm (Fig. 1)

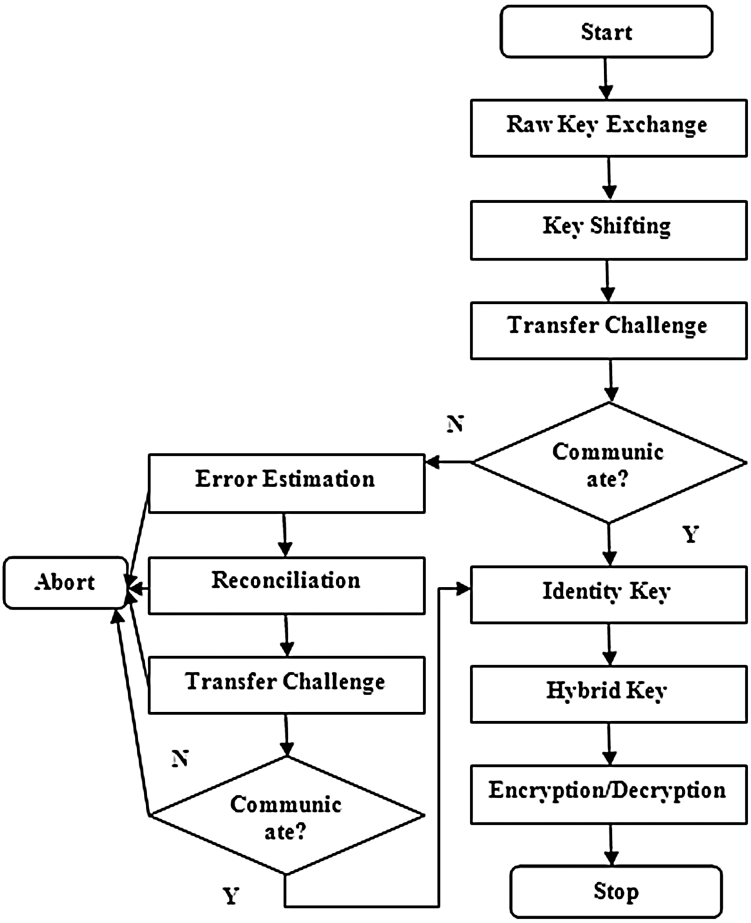

Fig. 1. Proposed algorithm flow

4 Conclusion

In this Paper the Proposed System has used Lucas Series to remove the number of assumptions and disadvantage of this series is that if starting two numbers are known then whole series is known. If Eve Knows Lucas series then interception becomes easy. So, rather than communicating with bit value Alice and Bob will communicate through MAC (Message Authentication Code). At the end, Alice or Bob will check whether keys are same or not by proving the given challenge and communicate if challenge has been proved. After adding Key Distillation Step, it improves the performance in

presence of Hardware Fault or Interception. It Transfers challenge twice to overcomes cycle problem.

Acknowledgement. We Express our Auspicious thank to ARTCom2013 conference who has motivated us to think extra on our previous proposed system by selecting our paper as an extended paper. And we have never worked with QKD hardware so whether this combination is going to work or not that we can't answer.

References

1. http://swissquantum.idquantique.com/?Key-Distillation
2. Cobourne, S.: Quantum key distribution protocols and applications (2011). http://www.ma.rhul.ac.uk/static/techrep/2011/RHUL-MA-2011-05.pdf
3. Implementation of the BB84 QKD Protocol. http://www.cki.au.dk/experiment/qrypto/doc/QuCrypt/bb84prot.html
4. Shaikh, A.M., Shah, P.D.: BB84 and identity based encryption based a novel symmetric key distribution algorithm. ARTCom2013 (2013)
5. http://www.maths.surrey.ac.uk/hosted-sites/R.Knott/Fibonacci/lucasNbs.html
6. Scharitzer, G.: Basic quantum cryptography. Version 0.9, Vienna University of Technology Institute of Automation (2003)