A Comparison of Several Implementations of B92 Quantum Key Distribution Protocol

Cătălin ANGHEL
Computer Science and Information Technology
“Dunărea de Jos” University of Galati
Galati, Romania
catalin.anghel@ugal.ro

Adrian ISTRATE
Computer Science and Information Technology
“Dunărea de Jos” University of Galati
Galati, Romania
adrian.istrate@ugal.ro

Mihai VLASE
Computer Science and Information Technology
“Dunărea de Jos” University of Galati
Galati, Romania
mihai.vlase@ugal.ro

Abstract—This paper presents the development, comparison, and analysis of several implementations of the B92 Quantum Key Distribution (QKD) protocol. To achieve this goal three prototypes which consists of traditional simulators (non-quantum) were created, one for the B92 protocol, one for the B92 protocol with an eavesdropper, and one for the B92 protocol with the Quantum Bit Travel Time (QBTT) eavesdropper detection method. The principles of quantum mechanics were studied, as a foundation of quantum cryptography, for the implementation of the simulation programs that were written in C++, focusing mainly on the B92 protocol and QBTT eavesdropper detection method. We compared the Quantum Bit Error Rate (QBER) for the implementation of the B92 protocol without an eavesdropper, the B92 protocol with an eavesdropper and the B92 protocol with the QBTT eavesdropper detection method and found that the latest one significantly reduces the QBER from the final key.

Keywords—quantum cryptography, quantum key distribution, qkd, quantum protocol, B92 protocol, QBTT eavesdropper detection method, qbit

I. INTRODUCTION

In this paper we will present the development, comparison, and analysis of several implementations of the B92 Quantum Key Distribution (QKD) protocol. We developed our own simulation programs, written in C++, to compare the Quantum Bit Error Rate (QBER) from the final key obtained by the B92 protocol without eavesdropper, the B92 protocol with an eavesdropper and the B92 protocol with Quantum Bit Travel Time (QBTT) eavesdropper detecting method. The QBTT eavesdropper detecting method that is proposed in [1] was successful in improving the Quantum Bit Error Rate for the BB84 Quantum Key Distribution protocol. We compared the Quantum Bit Error Rate (QBER) for implementation of the B92 protocol without an eavesdropper, the B92 protocol with an eavesdropper, and the B92 protocol with QBTT eavesdropper detection method. The results showed that QBER for the B92 protocol without an eavesdropper is approximately 75%, the QBER for the B92 protocol with an eavesdropper is around 89%, and the QBER for B92 protocol with QBTT eavesdropper detection method is at 75% thus we can say that the QBTT eavesdropper detection method significantly reduces the QBER from the final key.

The Quantum Key Distribution protocol B92, proposed by Charles Bennett in [2] allows two entities, the Sender and the Receiver, to establish a perfect secret, a common and unique key sequence using polarized photons – qbits, and also, due to the principles of quantum mechanics, it could detect if an Eavesdropper intercepted the quantum channel.

The unconditional security of the B92 quantum key distribution system has been demonstrated only for a mathematical model in [3] and [4]. In practice, this unconditional security cannot be achieved due to the technical imperfections of the devices involved in the exchange of quantum keys, either used for polarization or for the reading of the polarization of the photons.

II. THE B92 QKD PROTOCOL – ESSENTIALS

Quantum cryptography uses polarize photons – qbits – to represent the bits from the primary key. A photon can be polarized rectilinear (R) in one of the states (0°, 90°), diagonal (D) in one of the states (45°, 135°) and circular (left – spin L, right – spin R).

In the B92 quantum key distribution system, Sender will encode classical bits in two non-orthogonal states, figure 1 and Receiver will measure the qbits in all four states used in BB84 protocol as shown in [2] and [6], figure 2.

According to the laws of quantum mechanics, no type of measurement can distinguish between two polarized photons in non-orthogonal bases, as a result the corresponding bits cannot be identified with certainty. On the other hand, any attempt to intercept the transmitted qbits will alert the two communicating parties [6].

To implement the B92 quantum key distribution system, the Sender and the Receiver will use for the bit coding respectively for photon reading the convention set out in the table I.
TABLE I. POLARIZATION STATES FOR SENDER AND RECEIVER

|Sender| Receiver |
|---|---|
|Base| L | D |
|State| 0°| 45° |
|Qbit| → | ↗ |
|Bit| 0 | 1 |

|Sender| Receiver |
|---|---|
|Base| L | D |
|State| 0°| 45° | 90° | 135° |
|Qbit| → | ↗ | ↘ |
|Result| 0 | 0 | 1 | 1 |
|Bit| ? | ? | 1 | 0 |

**A. Steps of the B92 protocol**

1. The **Sender** creates a random bit string – primary key – noted $s$.
2. The **Sender** uses polarization states (0°, 45°) to represent bits in $s$.
3. The **Sender**, using a special equipment, creates a sequence $p$ of polarized photons – qubits, according to Table 1.
4. The **Sender** sends the qubits $p$ to **Receiver** over the quantum channel.
5. The **Receiver**, for each received qubit $p'$, randomly chooses a polarization base (linear "L" or diagonal "D"), $b'$.
6. The **Receiver**, using a special equipment, measures each received qubit with respect to the basis chosen in step 5, obtaining the raw key, string $s'$. According to Table 1 for each qubit detected as a ‘0’ announce the **Sender** and eliminate it from $s'$.
7. If the **Receiver** detects ‘0’ then $s'[i] = ?$
   - IF the **Receiver** detects ‘1’ then $s'[i] = 0$ IF $b'[i] = L$
   - or $s'[i] = 1$ IF $b'[i] = R$
8. The **Sender** eliminates from $s$ the corresponding bit where the **Receiver** detected ‘0’.

**B. Detecting eavesdropper’s presence**

For an **Eavesdropper** detection, B92 quantum key distribution system uses the QBER – Quantum Bit Error Rate method which involves calculating the percentage of errors in the final key [7], obtained at the end of the quantum transmission.

Quantum bit error rate is defined as:

$$ QBER = \frac{Q_0 - Q_1}{Q_1} \times 100 $$

Where $Q_0$ represents the number of qubits from primary key, and $Q_1$ represents the number of qubits from final key.

In the absence of the **Eavesdropper**, for the qubit in the $i$th position if we have $s'[i] \neq ?$, then $s'[i] = s[i]$.

If the **Eavesdropper** tried to read the qubit in the $i$th position, then the probability $Pr(s[i] = ?)$ increases.

In conclusion, the **Sender** and the **Receiver** will detect the **Eavesdropper** presence, due to the growth of the number of errors in the raw key and therefore the QBER value will increase.

**C. Stages of B92 protocol**

B92 quantum key distribution protocol has two communication stages [8]:

Stage 1: The quantum channel, a one-way communication through optical fiber. During this stage the primary key, encoded as qubits is transmitted from the **Sender** to the **Receiver** through the optical fiber.

Stage 2: The classical channel, a two-way communication through a public channel. During this stage **Sender** and the **Receiver** communicate over a classical channel in 3 main steps:

1. Detecting Eavesdropper presence;
2. Secret key reconciliation;
3. Privacy Amplification.

**III. IMPLEMENTATION OF B92 PROTOCOL WITHOUT EAVESDROPPER**

To implement through software simulation the B92 quantum key distribution protocol without an eavesdropper, the **Sender** and the **Receiver** will communicate through different TCP/IP ports and sockets, via a switch, that will simulate both the quantum and the classical channel.

This software application, that we developed in C++, consists of 4 objects: **Sender**, **Receiver**, **Quantum Channel** and **Classic Channel**.

At the end of the quantum transmission, the **Sender** and the **Receiver** will communicate through the classical channel and execute the steps: Detecting Eavesdropper presence, Secret key reconciliation and Privacy amplification.

**A. Hardware setup**

Block diagram of the B92 without an Eavesdropper implementation is presented in figure 3.

![Fig. 3. Implementation of B92 without Eavesdropper](image)

**B. Software setup**

For this simulation, each object (**Sender**, **Receiver**, **Quantum Channel** and **Classic Channel**) plays different role, as shown in figure 4.

![Fig. 4. Software protocol](image)
C. Pseudocode of the B92 protocol

1) Stage 1 – Quantum channel

   Sender:
   create random bits string s
   FOR each bit from s
      IF s[i] = 0 THEN polarize photon in state (0°)
      generate a qbit $p[i] = \rightarrow$
      IF s[i] = 1 THEN polarize photon in state (45°)
      generate a qbit $p[i] = \Rightarrow$
      send qbit $p[i]$ to Receiver
   ENDFOR

   Receiver:
   FOR each qbit $p'[i]$ received
      pick randomly from ("L", "D") → base $b'[i]$
      measure qbit $p'[i]$ in respect to base $b'[i] → v$
      IF $v = 0$
         THEN $s'[i] = ?$
         eliminate the corresponding bit from $s'$
         send value ‘0’ to Sender
      ELSE // $v = 1$
         IF $b'[i] = D$
            THEN $s'[i] = 0$
         ELSE // $b'[i] = L$
            THEN $s'[i] = 1$
         ENDIF
      ENDIF
   ENDFOR

2) Stage 2 – Classical channel

   a) Step 1 – Eliminate bits
   Sender:
   IF receive ‘0’
      THEN eliminate bit from $s$
   ENDF

   b) Step 2 – Detecting Eavesdropper presence:
   Sender and Receiver:
   IF Pr{$s'[i] = ?$} is higher than usual Eavesdropper was present
   ENDF

   c) Step 3 – Secret key reconciliation:
   Sender and Receiver:
   Interactive binary search for errors in $s$ and $s'$. (Strings $s$ and $s'$ are divided in small blocks of bits and their parity are compared. If the parity does not match, they are divided into smaller blocks and their parities are compared again, repeating this process until the exact location of the error is found. Once the error is found the corresponding bit from $s$ and $s'$ are discarded.)

   d) Step 4 – Privacy Amplification:
   Sender and Receiver:
   Apply random type of permutation in $s$ and $s'$. (A binary transformation – usually a random permutation – is applied to $s$ and $s'$, and a subset of bits are discarded from them).

D. Flowchart of B92 protocol

Flowchart of the B92 of quantum key distribution protocol is presented in figure 5.

E. Experimental results

After running the B92 without an eavesdropper simulation program 10 times, for a 512 bits primary key, the results from table II are obtained.

| Primary key | Raw key | Final key | QBER % |
|-------------|---------|-----------|--------|
| 512         | 132     | 132       | 74.22  |
| 512         | 120     | 120       | 76.56  |
| 512         | 115     | 115       | 77.54  |
| 512         | 138     | 138       | 73.05  |
| 512         | 121     | 121       | 75.59  |
| 512         | 142     | 142       | 72.27  |
| 512         | 140     | 140       | 72.66  |
| 512         | 132     | 132       | 74.22  |
Analyzing these data, we can see that the QBER value is approximately 75%, as shown in figure 6.

![Fig. 6. QBER – Primary key vs. Final key.](image)

**IV. IMPLEMENTATION OF THE B92 PROTOCOL WITH EAVESDROPPER**

For implementation through software simulation of the B92 quantum key distribution protocol with an eavesdropper present, the Sender, Receiver and an Eavesdropper will communicate through different TCP/IP ports and sockets, via a switch, that will simulate the quantum and the classical channel.

This software application, that we developed in C++, consists of 5 objects: Sender, Receiver, Eavesdropper, Quantum Channel and Classical Channel.

A. Eavesdropper attack method

The *Eavesdropper* use an Intercept-Resend attack [9] in which it will interrupt the quantum channel as shown in figure 7, intercept those qbits, read them and send to the Receiver other qbits according to its randomly choice of bases.

![Fig. 7. Intercept-Resend attack.](image)

The Receiver will acquire those modified qbits from the Eavesdropper through the quantum channel and read them according to its randomly choice of bases.

At the end of the quantum transmission, the Sender and the Receiver will communicate through the classical channel and execute the steps: Detecting Eavesdropper presence, Secret key reconciliation and Privacy amplification [8].

B. Hardware setup

The block diagram of the B92 protocol implementation is presented in figure 8.

![Fig. 8. Implementation of B92 with Eavesdropper.](image)

To implement the B92 protocol with an Eavesdropper we used: 3 workstations and 1 switch.

Each workstation represents the *Sender*, the *Receiver* and the *Eavesdropper*. Static IPs are used so that workstations can communicate via the switch. As described above, specific simulation software, developed by us, is installed on each workstation.

C. Software setup

For this simulation, each of object (Sender, Receiver, Eavesdropper, Quantum Channel and Classic Channel) play different role, as shown in figure 9.

![Fig. 9. B92 with eavesdropper software protocol.](image)

**TABLE III. SIMULATION RESULTS OF B92 WITH EAVESDROPPER**

| Primary key | Raw key | Final key | QBER % |
|-------------|---------|-----------|--------|
| 512         | 109     | 51        | 90.04  |
| 512         | 117     | 59        | 88.48  |
| 512         | 120     | 53        | 89.65  |
| 512         | 118     | 61        | 88.09  |
| 512         | 120     | 57        | 88.87  |
| 512         | 124     | 70        | 86.33  |
| 512         | 118     | 58        | 88.67  |
| 512         | 114     | 65        | 87.30  |
| 512         | 117     | 44        | 91.41  |
| 512         | 113     | 49        | 90.43  |

Analyzing these data, we can see that QBER for B92 with eavesdropper is approximately 89%, figure 10, with 14%
greater than QBER for B92 without eavesdropper, because an eavesdropper tapped the quantum channel.

Fig. 10. Primary key vs. Raw key vs. Final key.

V. IMPLEMENTATION OF B92 PROTOCOL WITH QBTT EAVESDROPPER DETECTION METHOD

For implementation through software simulation of the B92 protocol with QBTT eavesdropper detection method, we developed the software in C++ language. The Sender and the Receiver will communicate through the quantum channel and classical channel with or without the presence of the Eavesdropper.

A. Hardware setup

The block diagram of the B92 protocol with QBTT eavesdropper detection method implementation is presented in figure 9.

Devices used in this implementation are 3 workstations and 1 switch.

B. Software setup

For this simulation, only the appropriate function is executed on each of workstation, depends on its role: Sender, Receiver, Eavesdropper, Quantum channel or Classical channel, as shown in figure 11.

For each transmitted qbit, Sender also sends to the Receiver the timestamp T of transmission.

The receiver, for each measured qbit, calculates $\Delta T = T' - T$, where $T'$ represents the timestamp of the received qbit. If the delay time $\Delta T$ is not within normal limits, limits that were established by earlier lab communications tests [1], the Receiver will stop the transmission.

The QBTT eavesdropper detection method allows Receiver to detect the Eavesdropper’s presence at any time during quantum transmission, after each received qbit, not only at the end of it [1].

C. Pseudocode of the B92 with QBTT

1) Stage 1 – Quantum channel

Sender: SyncTime

create random bits string s
FOR each bit from s
IF $s[i]=0$ THEN polarize photon in state $(0^\circ)$
generate a qbit $p[i]$ = $\rightarrow$
ELSE 
IF $s[i]=1$ THEN polarize photon in state $(45^\circ)$
generate a qbit $p[i]$ = $\uparrow$
generate timestamp $T$
send qbit $p[i]$ to Receiver
send $T$ to Receiver
ENDFOR

Receiver: SyncTime

FOR each qbit $p'[i]$ received
pick randomly from (“L”, “D”) → base $b'[i]$
measure qbit $p'[i]$ in respect to base $b'[i]$ → $v$
generate timestamp $T'$
calculate $\Delta T = T' - T$
IF $\Delta T$ NOT in normal limits
Eavesdropper was present
STOP communication
ENDIF
IF $v = 0$ THEN $s'[i] = ?$
eliminate the corresponding bit from $s'$
send value ‘0’ to Sender
ELSE // $v = 1$
IF $b'[i] = D$ THEN $s'[i] = 0$
ELSE // $b'[i] = L$
THEN $s'[i] = 1$
ENDIF
ENDIF
ENDFOR

2) Stage 2 – Classical channel

a) Step 1 – Eliminate bits

Sender:
IF receive ‘0’
THEN eliminate bit from s
ENDIF

b) Step 2 – Secret key reconciliation:

Sender and Receiver:

Interactive binary search for errors in $s$ and $s'$.
(Strings $s$ and $s'$ are divided in small blocks of bits and their parities are compared. If the parity does not match, they are divided into smaller blocks and their parities are compared again, repeating this process until the exact location of the error is found. Once the error is found. Once the error is found the corresponding bits from $s$ and $s'$ are discarded.)
c) Step 4 – Privacy Amplification: 
Send and Receiver:

Apply random type of permutation in \(s\) and \(s'\).
(A binary transformation – usually a random permutation – is applied to \(s\) and \(s'\), and a subset of bits is discarded from them).

D. Experimental results

After running the B92 with QBTT simulation program 10 times, for a 512 bits primary key, the results from table IV were obtained.

| Primary key | Raw key | Final key | QBER % |
|-------------|---------|-----------|--------|
| 512         | 123     | 123       | 75.98  |
| 512         | 140     | 140       | 72.66  |
| 512         | 138     | 138       | 73.05  |
| 512         | 132     | 132       | 74.22  |
| 512         | 120     | 120       | 76.56  |
| 512         | 138     | 138       | 73.05  |
| 512         | 120     | 120       | 76.56  |
| 512         | 140     | 140       | 72.66  |
| 512         | 139     | 139       | 72.85  |
| 512         | 120     | 120       | 76.56  |

Analyzing these data, we can see that QBER value is approximately 75%, figure 12, same QBER as in simulation of B92 algorithm without eavesdropper, although the eavesdropper was present.

CONCLUSIONS

These software simulation programs are meant to give an alternative to physical implementation of the quantum devices used in quantum transmission.

This paper presents a comparison of the Quantum Bit Error Rate (QBER), obtained at the end of quantum transmission, between the B92 protocol without an eavesdropper, the B92 protocol with an eavesdropper and the B92 protocol with QBTT eavesdropper detection method, figure 13.

The results showed that QBER for B92 protocol without an eavesdropper is approximately 75%, the QBER for B92 protocol with eavesdropper is around 89% and the QBER for B92 protocol with QBTT eavesdropper detection method is at 75% thus we can say that the QBTT eavesdropper detection method, implemented in the B92 protocol, significantly reduces the QBER from the final key.

We can observe the advantages of the implementation of Quantum Bit Travel Time – QBTT eavesdropper detection method in B92 protocol by reducing the percentage of the Quantum Bit Error Rate – QBER from the final key and the instant detection of the eavesdropper during the quantum transmission.

As we shown in [1], that QBTT eavesdropper detection method can be implemented in the BB84 protocol. As a future work we consider implementing QBTT eavesdropper detection method in a quantum entanglement scheme like E91.

REFERENCES

[1] Anghel C., „New eavesdropper detection method in quantum cryptography”, The annals of “Dunărea de Jos” University of Galati, fascicule III, vol.34, nr. 1, pg. 1-8, 2011.
[2] Bennett C.H., „Quantum cryptography using any two nonorthogonal states”, Physical Review Letters, vol. 68, pg. 3121-3124, 1992.
[3] Inamori H., Lütkenhaus N. & Mayers D., „Unconditional Security of Practical Quantum Key Distribution”, European Physical Journal D, vol. 41, pg. 599-627, 2007.
[4] Scarani V., Bechmann-Pasquinucci H., Cerf N.J., Dušek M., Lütkenhaus N. and Peev M., „The security of practical quantum key distribution”, Review of Modern Physics, vol. 81, pg. 1301-1350, 2009.
[5] Bell J.S., „On the Einstein-Podolsky-Rosen paradox”, Physics 1, vol. 1, nr. 3, pg. 195-200, 1964.
[6] Bennett, C. H. and Brassard, G., “Quantum cryptography: Public key distribution and coin tossing”, Proceedings of IEEE International Conference on Computers Systems and Signal Processing, Bangalore, India, pg. 175-179, 1984.
[7] Treiber A., „A fully automated quantum cryptography system based on entanglement for optical fiber networks”, New Journal of Physics, Vol. 11, nr. 4, pg. 1–19, 2009.
[8] Wijesekera, Shirantha, Sajal Palit, and Bala Balachandran., „Software development for b92 quantum key distribution communication protocol”, 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS 2007), pg. 274-278, IEEE, 2007.
[9] Makarov V., Anisimov A. & Skaar J., „Effects of detector efficiency mismatch on security of quantum cryptosystems”, Physical Review A, vol. 74, pg. 1-11, 2005.