# A COMPARISON OF SEVERAL IMPLEMENTATIONS OF B92 QUANTUM KEY DISTRIBUTION PROTOCOL

Cătălin ANGHEL

University “Dunărea de Jos” of Galați, Faculty of Automation, Computers, Electrical Engineering and Electronics,
Department of Computer Science and Information Technology, 2 Științei Street, 800146, Galați, Romania.
E-mail: catalin.anghel@ugal.ro

**Abstract.** This paper presents the development, comparison and analysis of several implementations of the B92 Quantum Key Distribution (QKD) protocol. In order to achieve this objective a prototype which consists of traditional (non-quantum) simulators was created, one for B92 protocol, one for B92 protocol with eavesdropper and one for B92 protocol with Quantum Bit Travel Time (QBTT) eavesdropper detection method. The principles of quantum mechanics were studied, as a foundation of quantum cryptography, for the realization of 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 implementation of B92 protocol without eavesdropper, B92 protocol with eavesdropper and B92 protocol with QBTT eavesdropper detection method and found that QBTT eavesdropper detection method significantly reduces the QBER from the final key.

***Key words***: quantum cryptography, quantum physics, quantum key distribution, qkd, quantum protocol, B92 protocol, QBTT eavesdropper detection method, qbit.

## 1. INTRODUCTION

In this paper we will present the development, comparison and analysis of several implementations of B92 Quantum Key Distribution (QKD) protocol. We developed simulation programs, written in C++, to compare the Quantum Bit Error Rate (QBER) from the final key obtained by the B92 protocol without enemy attack, B92 protocol with enemy attack and B92 protocol with Quantum Bit Travel Time (QBTT) eavesdropper detecting method. We compared the Quantum Bit Error Rate (QBER) for implementation of B92 protocol without eavesdropper, B92 protocol with eavesdropper and B92 protocol with QBTT eavesdropper detection method. The results showed that QBER for B92 protocol without eavesdropper is approximatively 75%, QBER for B92 protocol with eavesdropper is around 89% and 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.

Quantum key distribution protocol B92, proposed by Charles Bennett in 1992 [1], makes it possible for two entities, the *Sender* and the *Receiver*, to establish a perfectly secret, common and unique key sequence using polarized photons – qbits, and also can detect the *Eavesdropper* that intercepted the quantum channel.

The unconditional security of the B92 quantum key distribution system, has been demonstrated only for mathematical models [2, 3]. In practice, this unconditional security cannot be achieved due to the technical imperfections of the devices, used for polarization or the reading of polarization of the photons, involved in the exchange of quantum keys.

The security of the quantum key distribution systems is also given by the effectiveness of the method of detecting possible attacks. There are also several methods of detecting attacks on quantum key distribution systems [4], but the most reliable is the Quantum Bit Travel Time (QBTT) detection method [5]. The QBTT eavesdropper detection method can be implemented in any quantum key distribution system and has the advantage that the enemy can be detected accurately and especially immediately after it has read a qbit, during the quantum transmission [5].

## 2. B92 QKD PROTOCOL – ESSENTIALS

In B92 quantum key distribution system, *Sender* will encode classical bits in qbits polarized in two non-orthogonal states, figure 1 [1] and *Receiver* will measure the qbits, in order to decode the bits, in all four states used in BB84 protocol [6], figure 2 and figure 3.

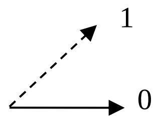

Fig. 1 – *Sender's* nonorthogonal polarization states

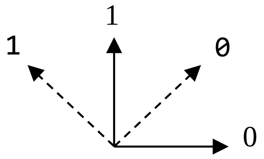

Fig. 2 – *Receiver's* measurement states

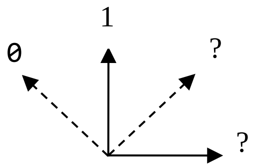

Fig. 3 – *Receiver's* interpretation states

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 *Receiver* will use for the bit coding respectively for photon reading the convention set out in the table 1.

Table 1. Polarization states for *Sender* and *Receiver*

| **Sender** | | |
|---|---|---|
| Base | L | D |
| State | $0^o$ | $45^o$ |
| Qbit | → | ↗ |
| Bit | 0 | 1 |

| **Receiver** | | | | |
|---|---|---|---|---|
| Base | L | D | L | D |
| State | $0^o$ | $45^o$ | $90^o$ | $135^o$ |
| Qbit | → | ↗ | ↑ | ↖ |
| Result | 0 | 0 | 1 | 1 |
| Bit | ? | ? | 1 | 0 |

### 2.1 Steps of the B92 protocol

1. *Sender* create a random bit string noted **s**.
2. *Sender*, use polarization states ($0^o$, $45^o$) to represent bits in **s**.
3. *Sender*, using a special equipment, creates a sequence **p** of polarized photons – qbits, according to table 1.
4. *Sender* sends the qbits **p** to *Receiver* over the quantum channel.
5. *Receiver*, for each received qbit, randomly chooses a polarization base (linear "L" or diagonal "D").
6. *Receiver*, using a special equipment, measures each received qbit with respect to the basis chosen in step 5. According to table 1 for each qbit detected as a '0' announce the *Sender* and eliminate it from **s'**.
7. If *Receiver* detects '0' then **s' = ?**

doi:10.20944/preprints202102.0486.v1

If *Receiver* detects ‘1’ then **s' = 0** if **b' = L**

or **s' = 1** if **b' = R**

8. *Sender* eliminate from **s** the corresponding bit where the *Receiver* detected ‘0’.

### 2.2 Detecting eavesdropper’s presence

For *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 quantum transmission.

Quantum bit error rate is defined as:

$$QBER = \frac{Q_I - Q_F}{Q_I} * 100 \tag{1}$$

where $Q_I$ represent the number of qbits from primary key, and $Q_F$ represent the number of qbits from final key.

In the absence of the *Eavesdropper*, for the qbit in the i$^{th}$ position if we have **s'**[i] ≠ ?, then **s'**[i] = **s**[i].

If the *Eavesdropper* tried to read the qbit 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.

### 2.3 Stages of B92 protocol

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

Stage 1: Quantum channel, a one-way communication through optical fiber.

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

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

## 3. IMPLEMENTATION OF B92 PROTOCOL WITHOUT EAVESDROPPER

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

This software application 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*.

### 3.1 Hardware setup

Block diagram of B92 protocol without *Eavesdropper* implementation is presented in figure 4.

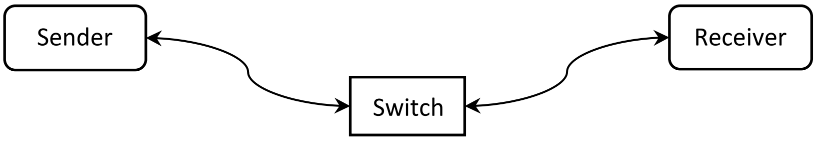

Fig. 4 – Hardware implementation of B92 protocol without Eavesdropper

To implement the B92 protocol without Eavesdropper we used: 2 workstations and 1 switch or router.

doi:10.20944/preprints202102.0486.v1

Each workstation represents the *Sender* and the *Receiver*. Static IP are used so that workstations can communicate through the switch. Specific simulation software is installed on each workstation.

### 3.2 Hardware setup

For this simulation, each of object *(Sender*, *Receiver*, *Quantum Channel* and *Classic Channel)* play different role, as shown in figure 5.

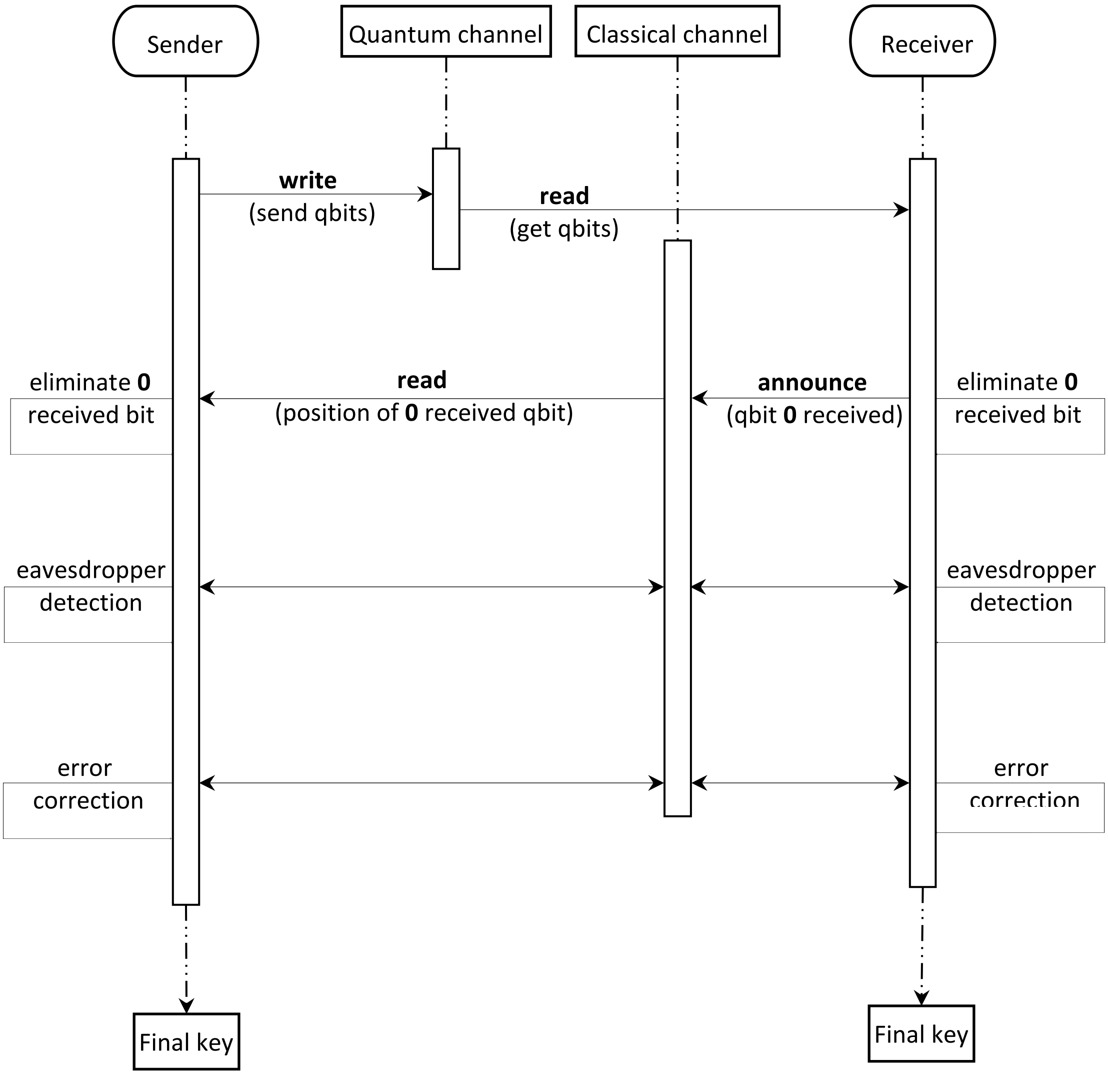

Fig. 5 – Software protocol of B92 without eavesdropper

### 3.3 Pseudocode of the B92 protocol

**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] = →
           IF s[i] = 1 THEN polarize photon in state (45°)
```

```
            generate a qbit p[i] = ↗
            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] → val
            IF val = 0 THEN s'[i] = ?
                    eliminate the corresponding bit from s'
                    send value '0' to Sender
            ELSE  // val = 1
                    IF b'[i] = D THEN s'[i] = 0
                            ELSE  // b'[i] = L
                                    THEN s'[i] = 1
            ENDIF
    ENDFOR
```

### Stage 2 – Classical channel

***Step 1 – Eliminate bits***

***Sender:***

```
    IF receive '0'
            THEN eliminate bit from s
    ENDIF
```

***Step 2 – Detecting Eavesdropper presence:***

***Sender and Receiver:***

```
    IF Pr{s'[i] = ?} is higher than usual
            Eavesdropper was present
    ENDIF
```

***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 didn't match, they are divided it 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.*)

***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*).

### 3.4 Logical diagram of B92 protocol

Logical block diagram of B92 of quantum key distribution protocol is presented in figure 6.

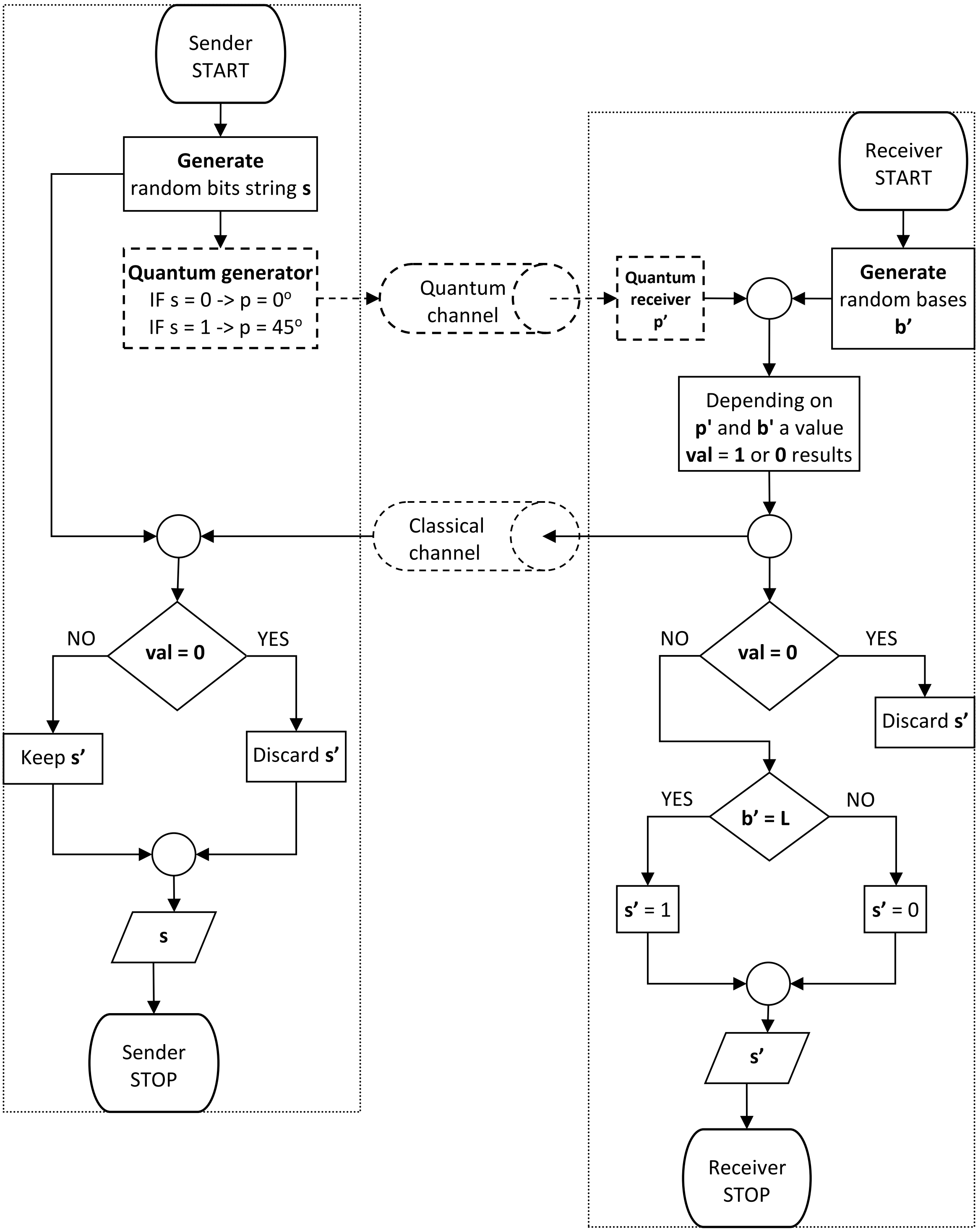

Fig. 6 – Logical block diagram of B92 protocol

### 3.5 Experimental results

After running the B92 without eavesdropper simulation program 10 times, for 512 bits primary key, we obtain the results from table 2.

Analyzing these data, we can see that QBER value is approximately 75%, as shown in figure 7.

Table 2. Simulation results of B92 without eavesdropper

| Primary key | Final key | QBER % |
|---|---|---|
| 512 | 132 | 74.22 |
| 512 | 120 | 76.56 |
| 512 | 115 | 77.54 |
| 512 | 138 | 73.05 |
| 512 | 121 | 76.37 |
| 512 | 137 | 73.24 |
| 512 | 125 | 75.59 |
| 512 | 142 | 72.27 |
| 512 | 140 | 72.66 |
| 512 | 132 | 74.22 |

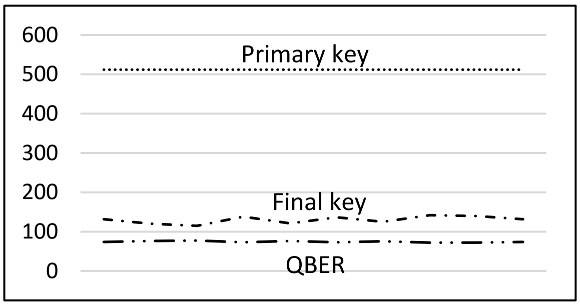

Fig. 7 – *QBER* – Primary key vs. Final key

## 4. IMPLEMENTATION OF B92 PROTOCOL WITH EAVESDROPPER

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

This software consists of 5 objects: *Sender*, *Receiver*, *Eavesdropper*, Quantum Channel and Classical Channel.

*Eavesdropper* use Intercept-Resend attack [9] in which he will interrupt the quantum channel as shown in figure 8, intercept those qbits, read them and send to *Receiver* other qbits according to his randomly choice of bases.

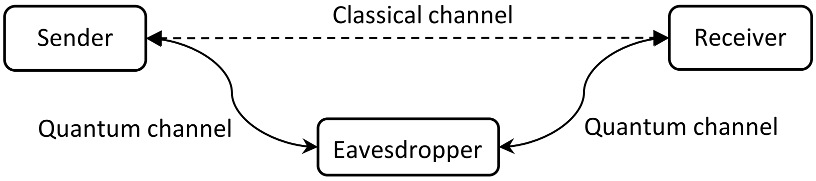

Fig. 8 – Intercept - Resend attack

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

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

### 4.1 Hardware setup

Block diagram of B92 protocol with eavesdropper implementation is presented in figure 9.

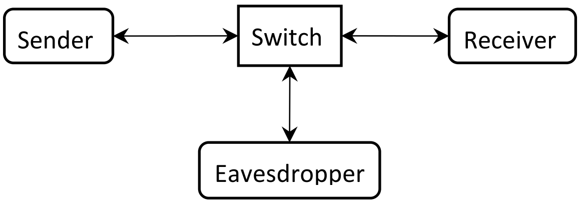

Fig. 9 – Implementation of B92 with Eavesdropper

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

Each workstation represents the *Sender*, *Receiver* and *Eavesdropper*. Static IP are used so that workstations can communicate via the switch. Specific simulation software is installed on each workstation.

### 4.2 Software setup

For this simulation, each of object (*Sender*, *Receiver*, Eavesdropper, Quantum Channel and Classic Channel) play different role, as shown in figure 10.

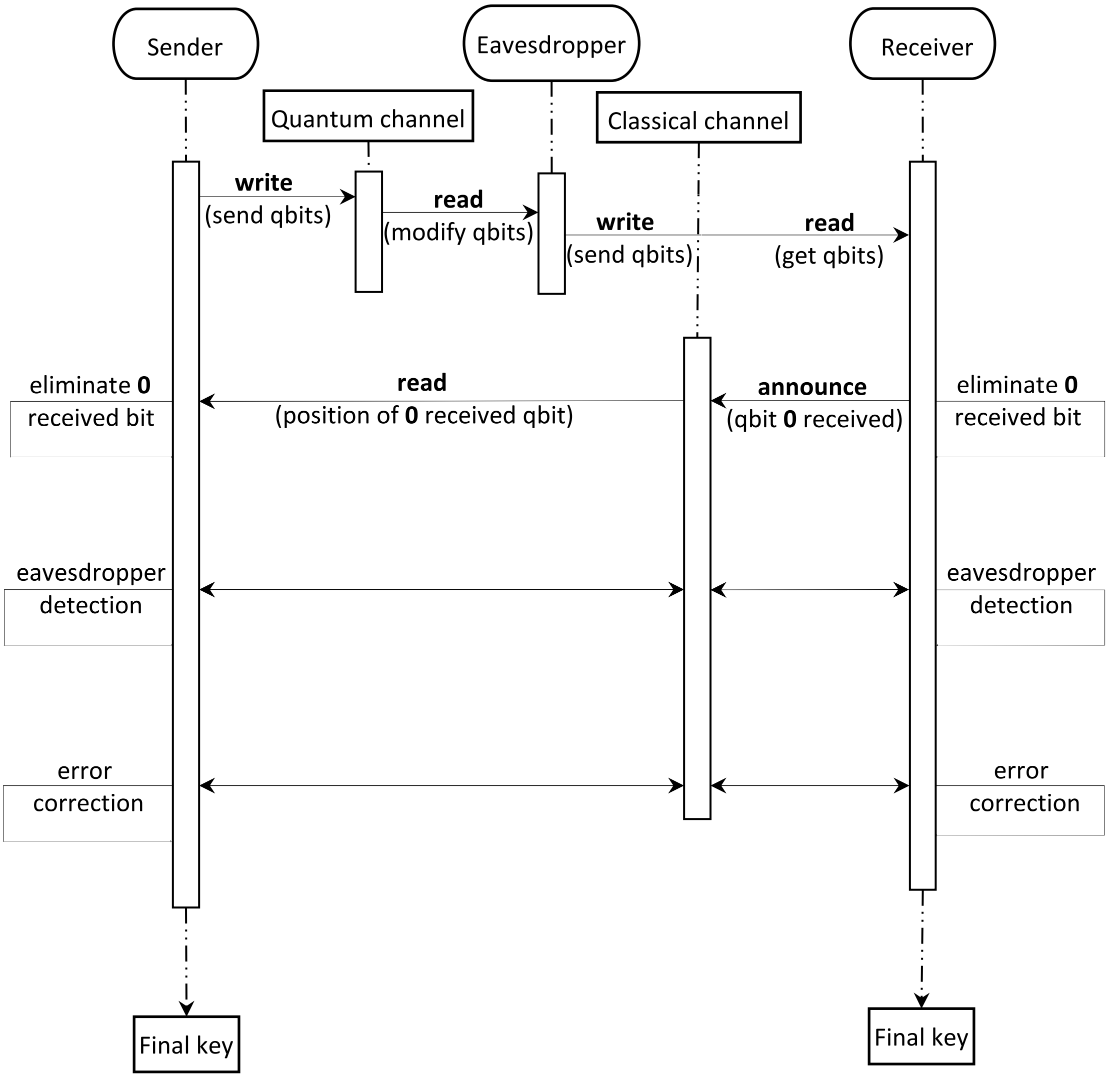

Fig. 10 – Software protocol of B92 with eavesdropper

### 4.3 Experimental results

After running the B92 with eavesdropper simulation program 10 times, for 512 bits primary key, we obtain for raw key, final key and quantum bit error rate - QBER the results from table 3.

Analyzing these data, we can see that QBER for B92 with eavesdropper is approximately 89%, figure 11, with 14% greater than QBER for B92 without eavesdropper, because the eavesdropper tapped the quantum channel.

Table 3. 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 |

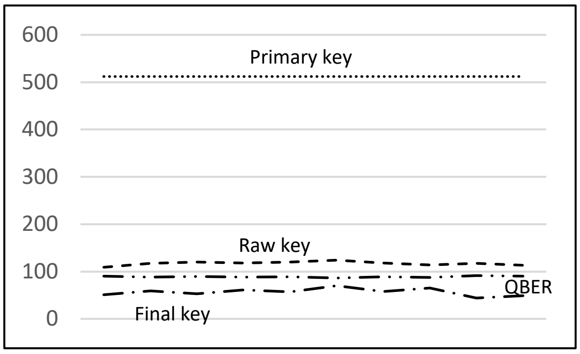

Fig. 11 – *QBER* – Primary key vs. Raw key vs. Final key

## 5. SIMULATION OF B92 PROTOCOL WITH QBTT EAVESDROPPER DETECTION METHOD

For the simulation of B92 protocol with QBTT eavesdropper detection method, the software has been developed using C++ language. *Sender* and *Receiver* will communicate through quantum channel and classical channel with or without the presence of the *Eavesdropper*.

### 5.1 Hardware setup

Block diagram of B92 protocol with QBTT eavesdropper detection method implementation is presented in figure 12.

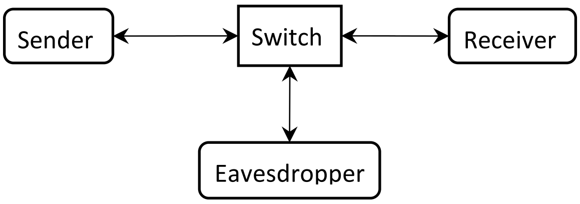

Fig. 12 – Implementation of B92 with Eavesdropper and QBTT eavesdropper detection method

To implement the B92 protocol with Eavesdropper and QBTT eavesdropper detection method we used: 3 workstations and 1 switch or router.

Each workstation represents the *Sender*, *Receiver* and *Eavesdropper*. Static IP are used so that workstations can communicate via the switch. Specific simulation software is installed on each workstation.

### 5.2 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 13.

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

*Receiver*, for each measured qbit, calculate $\Delta T = T' - T$, where *T'* represent the timestamp of received qbit. If the delay time $\Delta T$ is not in normal limit, limits established by earlier communications, *Receiver* will stop the transmission.

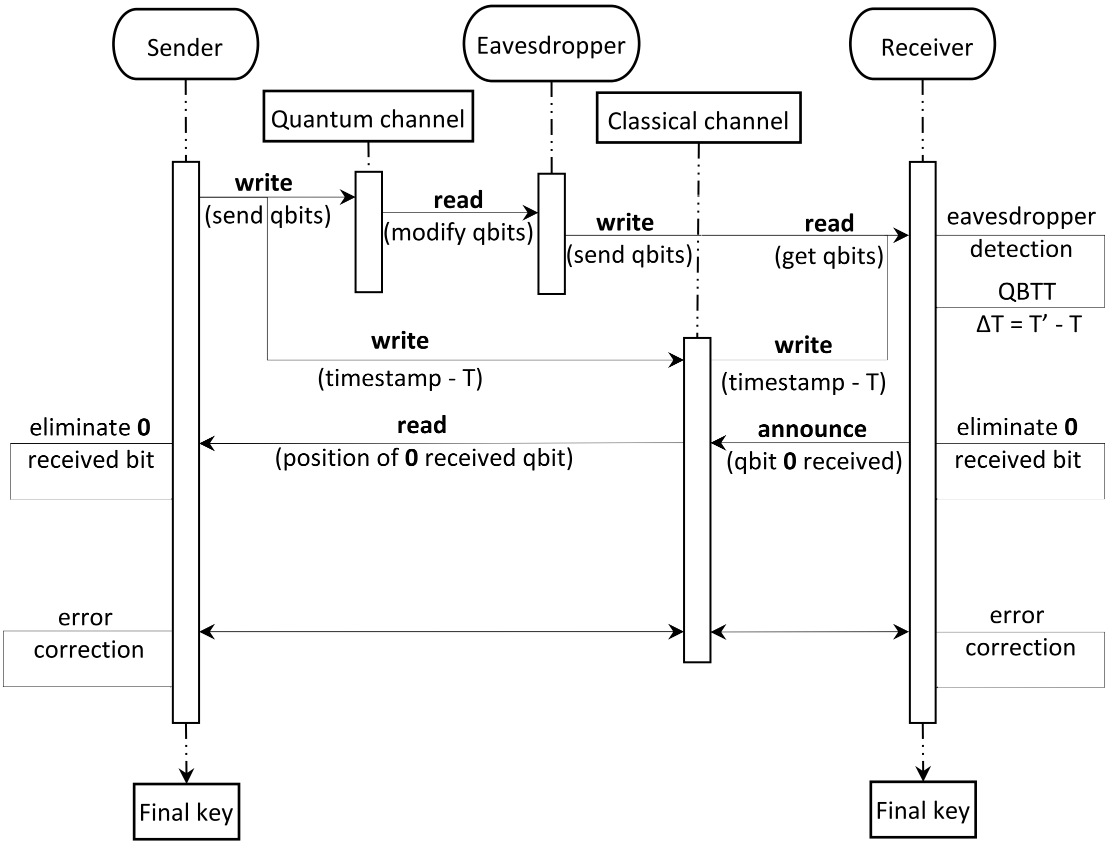

Fig. 13 – Software protocol of B92 with QBTT eavesdropper detection method

### 5.3 Pseudocode of the B92 with QBTT

**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°)
               generate a qbit p[i] = →
   IF s[i]=1 THEN polarize photon in state (45°)
               generate a qbit p[i] = ↗
               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
```

### Stage 2 – Classical channel

***Step 1 – Eliminate bits***
***Sender:***

```
    IF receive ‘0’
            THEN eliminate bit from s
    ENDIF
```

***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 parity are compared. If the parity didn't match, they are divided it 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.)*

***Step 3 – 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).*

## 5.4 Experimental results

After running the B92 with QBTT simulation program 10 times, for 512 bits primary key, we obtain the results from table 4.

Analyzing these data, we can see that QBER value is approximately 75%, figure 14, same QBER as in simulation of B92 algorithm without eavesdropper, although the eavesdropper was present.

Table 4. Simulation results of B92 with QBTT eavesdropper detection method

| 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 |

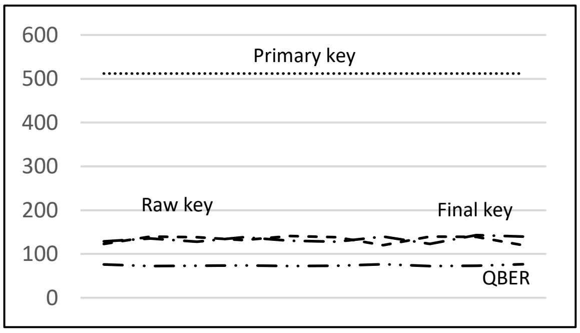

Fig. 14 – *QBER* – Primary key vs. Raw key vs. Final key

## 6. CONCLUSIONS

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

This paper presents a comparison of the Quantum Bit Error Rate (QBER), between B92, B92 with eavesdropper and B92 with QBTT eavesdropper detection method, figure 15.

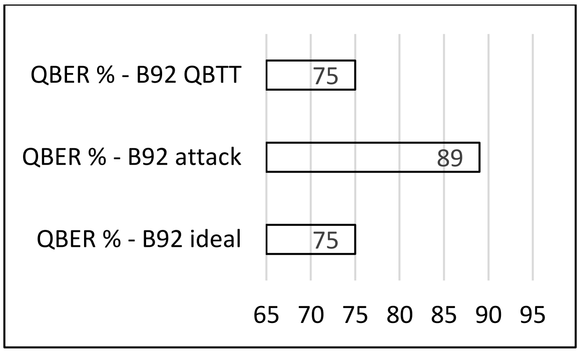

Fig. 15 – *QBER* comparison

The results showed that QBER for B92 protocol without eavesdropper is approximatively 75%, QBER for B92 protocol with eavesdropper is around 89% and 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.

We can observe the advantages of the Quantum Bit Travel Time – QBTT eavesdropper detection method by reducing the percentage of the Quantum Bit Error Rate – QBER from the *final key*.

## REFERENCES

1. C. H. BENNETT, *Quantum Cryptography using any two Nonorthogonal States*, Physical Review Letters, **vol. 68**, pp. 3121-3124, 1992.
2. H. INAMORI, N. LÜTKENHAUS & D. MAYERS, *Unconditional Security of Practical Quantum Key Distribution*, European Physical Journal D, **vol. 41**, pp. 599-627.
3. V. SCARANI, H. BECHMANN-PASQUINUCCI, N.J. CERF et al., *The security of practical quantum key distribution*, Review of Modern Physics, **vol. 81**, pg. 1301-1350, 2009.
4. J. S. BELL, *On the Einstein-Podolsky-Rosen paradox*, Physics 1, **vol. 1**, nr. 3, pg. 195-200, 1964.
5. C. ANGHEL, 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.
6. C. H. BENNETT AND G. BRASSARD, *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. A. TREIBER, *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. S. WIJESEKERA, S. PALIT, AND B. 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. V. MAKAROV, A. ANISIMOV & J. SKAAR, *Effects of detector efficiency mismatch on security of quantum cryptosystems*, Physical Review A, **vol. 74**, pg. 1-11, 2005.

*Received February 2021*