Comparison and Analysis of BB84 and E91 Quantum Cryptography Protocols Security Strengths

Luis Cáceres Alvarez, Patricio Collao Caiconte

***Abstract*— This research performs the comparison and analysis of BB84 and E91 Quantum Cryptography Security Protocols. In order to achieve this objective a prototype which consists of traditional (non-quantum) simulators was created, one for BB84 protocol and another for the E91. The principles of Quantum Mechanics were studied, as the foundation of Quantum Computing and Cryptography, focusing mainly on the BB84 and E91 protocols, which are Quantum Key Distribution (QKD). With the aim of comparing both protocols, tests with both simulators in different network environments were conducted, using different initial key length, capturing the required time and final length of the obtained key. The results showed that BB84 protocol converges at a 50% of the initial key, for keys which are higher than 265 initial bits, the percentage matches the expected theory. Whereas E91 protocol is approximated to 22% of the initial key for those which are higher than 64 initial bits, which does not match the 33% theoretically expected. Therefore, E91 protocol is considered to be safer, however it faces some technical difficulties, specifically with the Quantum entanglement principle.**

***Index Terms*— BB84 Protocol, BB84 Simulator, E91 Protocol, E91 Simulator, Quantum Cryptography, QKD.**

I. Introduction

This research performs a comparison and analysis of the security strengths in BB84 and E91 Quantum Cryptography Protocols, to generate and distribute a key, using channels which possess classic and quantum properties. The aim of these protocols is to improve security in network communication, where security is a crucial concern. With the purpose of comparing both protocols, a software prototype which simulates each protocol behavior was created, to execute a series of tests; obtaining data that added to the theoretical aspect providing a foundation for this analysis.

Nowadays, there is a great amount of financial transactions, including purchasing and selling of goods and services, or the delivery of sensitive information. Within the age of information, cryptography provides basic mechanisms to ensure privacy, integrity, authentication and non-rejection. Given the fast advance of technology, cryptographic algorithms are threatened by the increasing power of computation, which is capable of "breaking" security with very basic strategies, such as brute force. This situation becomes clearer with the arrival of quantum computing, which will practically outdate current cryptographic algorithms; and the arrival of quantum cryptography comes from quantum computing, which have their foundations in the principles of Quantum Mechanics.

There are 2 important aspects in the foundations of cryptography, which are: key distribution and data encryption. For the first one, there are several quantum cryptography algorithms, classified as Quantum Key Distribution (QKD) such as the following: BB84 [1], E91 [2], B92 [3] and SARG04 [4], which in this case are BB84 and E91, the main research area of this work.

The main goal of this research is the comparison and analysis of security strengths in BB84 and E91 Quantum Cryptography protocols, through a simulation prototype. This prototype maintains the line of study, which begun with the development of BB84 original simulator [5] that was improved in [6] and then, continued to the development of the E91 simulator [7], converging in the current research.

For a better understanding, this article has been structured in 6 sections; beginning with the introduction, followed by the theoretical framework, Quantum Cryptography Protocols, QCP simulation, prototype tests and finally, the conclusions.

II. Theoretical Framework

Due to the increase of computational power, and the vulnerabilities discovered by researchers which allows the reduction of the effort to perform an attack, permitted that some well-known traditional cryptographic algorithms have suffered assaults, such as the RSA-768 where its factorization is reported in [8] also RSA-1024 which was broken in approximately 100 hours [9]. In the other hand, the security level of the AES[1] algorithm has been reduced, because its 128/8[2], 192/9 and 256/9 variables have been reduced in terms of computing complexity, in approximately 2 bits [11]. Another report [12], points out that AES 128/10 is safer than AES 256/14 for any key. AES 256/14 has the same strength that a theoretical AES 119/14, below than AES 123/10 which would obtain with an attack to AES 128/10.

Although the security level of these algorithms have been reduced, the power of computing remains enormous, though with these reductions the possibility of using supercomputers from the TOP500 [13] to exploit these vulnerabilities, is increased.

Currently, there is a quantum computer used for commercial purposes, called D-Wave Two, which possess a quantum computing system, consisting of a chipset of 512 qubit [14]. This system proved to be slightly quicker than the traditional computer, although with a price difference of 6000 times [15].

However, in the same article they mention one of the criticisms that this quantum computer has received: the fact that it is only useful for optimization problems, due to its use

Luis Cáceres Alvarez, Área de Computación e Informática, Universidad de Tarapacá, Arica, Chile.
Patricio Collao Caiconte, Área de Computación e Informática, Universidad de Tarapacá, Arica, Chile.

[1] AES: Advanced Encryption Standard (AES), which is an encryption algorithm by symmetric blocks [10].

[2] Which means "bits versus rounds' number", in this case, 128 bits and 8 rounds [10].

of quantum annealing[3]. A real quantum computer should use quantum entanglement[4]. Given all this advantages and disadvantages, D-Wave Two is to be acknowledged as a great advance. On the other hand, Google has been working in the development of a quantum computer that can identify objects in a database of images and videos [18].

Another crucial issue to achieve quantum cryptography spreading are communication networks. This point has been demonstrated in a new scalable approach to guarantee quantum information, called Network-centric Quantum Communications (NQC) [19]. This approach has been used in governmental networks since 2010. Efforts have been made at commercial level, as was done by the company Toshiba [20]. Another major project was the first computer network protected by QKD called SECOQC, launched in 2008 and funded by the European Union. Details of the finished project (2010) and related publications are available on their website [21].

Despite these advances in networks, current technology still needs to be improved, as was discussed in [22] where current technology is shown to be vulnerable to the "blinding effect", which can be used by a spy to identify emitted photons, without causing errors to the receptor, allowing the spy to avoid being detected.

On the other hand, it must be addressed that quantum computing algorithms, as Shor algorithm [23], can efficiently solve problems of exponential equations by using a quantum computer. The previously mention characteristic endangers current asymmetric cryptography algorithms (or public key) as RSA, while symmetric cryptography algorithms (or secret key) are able to remain undamaged, as was mentioned by Hellman and Scolnik [24]. As the previous evidence showed, QKD protocols may replace asymmetric cryptography algorithms given their superior security level. Currently, there are companies that sell systems to use QKD protocols, such as IDQ, MagiQ, QuintessenceLabs and SeQureNet.

Finally, quantum cryptography has reached relevance levels that have generated an annual conference aimed to discuss advances in that area since 2011. This conference, called QCRYPT gathers universities and companies around this topic, and it has taken place in several continents.

Within this context, this research is intended to contribute to quantum cryptographic advances, using the implementation of a prototype for the simulation of quantum cryptography protocols BB84 and E91 as a research case. Using traditional informatics technological resources, plus the analysis performed to these results, will facilitate the access and understanding of this topic. Due to the fact that a real application requires high amounts of investment and advanced knowledge of quantum mechanics, this experiment provides a simplified solution to quantum cryptography issues.

III. Quantum Cryptography Protocols

One of the problems with higher practical difficulty, when performing a secure communication through a secret key system, is the safe distribution of keys. Quantum mechanics laws address the problem of safe distribution of secret keys. Individuals are able to transmit the secret key through a quantum channel, such as an optical fiber cable. In this case, polarization states of a photon can be used to design a quantum cryptography protocol, for the distribution of a single use random key.

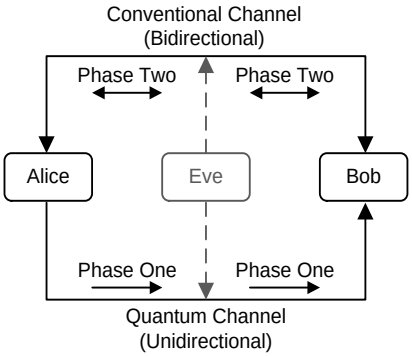


Fig. 1: Quantum communication model [5].

As shown in Fig. 1, these quantum cryptography protocols possess two channels; a quantum channel through which the key is transmitted (Phase One) and a conventional channel through which communication participants communicate (Phase Two).

A. Protocol BB84

This protocol was proposed by Charles Bennet and Gilles Brassard in 1984 [1]. The idea is to transmit a binary key through an unsafe channel. To transmit the zero bit, Alice (the sender) can randomly choose the base $\{|0\rangle,|1\rangle\}$ (which will be called scheme $+$) and consider $0 \Leftrightarrow |0\rangle$ and $1 \Leftrightarrow |1\rangle$, or the base $\{|-\rangle,|+\rangle\}$ (called scheme $\times$) and consider $0 \Leftrightarrow |-\rangle$ and $1 \Leftrightarrow |+\rangle$. Bob will perform a measure on the received state, randomly picking between scheme $+$ and scheme $\times$.

The following is the full process of key exchange, as mentioned in [25]:

1. Alice begins to transmit a random sequence of $0s$ and $1s$, alternating the schemes $+$ and $\times$ randomly.
2. Bob receives the sequence, and alternates the measures between schemes $+$ and $\times$ randomly.
3. Alice transmits to Bob the succession of schemes used.
4. Bob reports to Alice in what cases he was able to guess the origin scheme.
5. By using only the bits of two match identical schemes, they both have defined a random succession of bits that will do as one time pad[5] encrypted for future transmissions through any channel.
6. Alice and Bob exchanges key hashes to accept or reject the key.

Associating to Fig. 1, steps 1 to 2 are considered to be Phase One, using the quantum channel; while steps 3 to 6 from Phase Two go through the traditional channel.

This protocol is theoretically unbreakable. Lets assume that Eve spies the communication channel between Alice and Bob, and tries to retrieve the key. Eve is in the same situation as Bob, and does not know what the correct scheme is, $+$ o

[3] Adiabatic quantum computing type [16].

[4] Quantum phenomenon, where the quantum states of two or more objects must be described referencing to the quantum states of all the objects in the system, even if the objects are spatially separated [17].

[5] The one-time pad is a type of encryption algorithm, invented by 1917, where the original text is combined with a random key of the same length of the text and that is used just once [26].

×. Therefore she chooses randomly and, on an average she will fail half of the times (as if she directly tried to guess the key). On step 5 Alice and Bob agree on which values to be taken into account (coincidences of scheme sequences). This information is not useful for Eve because only half of the times she will hit the target, which means that she will misinterpret the final values [25]. Table I shows an example of this steps' sequence.

Table I: Communication example using BB84 protocol [5]

Alice's schemes	×	+	+	×	×	+
Alice's values	$\|-\rangle$	$\|0\rangle$	$\|0\rangle$	$\|+\rangle$	$\|-\rangle$	$\|0\rangle$
Bob's schemes	+	×	+	×	+	+
Bob' values	$\|0\rangle$	$\|+\rangle$	$\|0\rangle$	$\|+\rangle$	$\|1\rangle$	$\|0\rangle$
Coincidences			yes	yes		yes
Key			0	1		0

Also, QKD provides a method for Alice and Bob to detect the spy potential of Eve. Imagining that Alice sends a zero with the scheme × (represented by $|-\rangle$), Eve uses the scheme + forcing he qubit to define itself as $|0\rangle$ or $|1\rangle$. If Bob uses the scheme × and measures $|-\rangle$ it matches with what was sent by Alice, but if it measures $|+\rangle$, Alice and Bob will discover this discrepancy during the hashes exchange, therefore they will discard the block [25].

B. Protocol E91

The second protocol to be discussed is E91 [2], which was developed by Arthur Ekert in 1991. This protocol uses entangled photons. These can be prepared by Alice, Bob or any third person, and are distributed in a way that Alice and Bob have one photon of each pair. The model is based on quantum entanglement. To start, entangled photons are produced, so if Alice and Bob measure the photon's orientation (whether is vertical or horizontal) they will always obtain opposite responses, the same way as if they measure diagonal bases. The individual results are completely random, meaning that it cannot predict what Alice will obtain on her measure, for instance, a vertical or horizontal orientation.

On the other hand, any attempt to listen made by the spy (Eve) will ruin the correlation, that Alice and Bob will be able to detect the intrusion. According to [27] this protocol is based on the following algorithm:

1. N pairs of entangled states are generated in a random way, being n the initial length of the key.
2. For each pair of entangled state, one is sent to Alice and another to Bob.
3. Alice and Bob independent and randomly choose a measure bases and apply them to each photon.
4. After the measures, Alice and Bob uncover their schemes, which are the bases' lists used in each measure (keeping secretly the obtained results).
5. In cases which they used the same base, the coincidence is assured, while on the other cases the photons are not considered, and in this way they obtain the common key.
6. Alice and Bob Exchange the key's hashes to accept or reject the key.

A general diagram of the previous steps can be seen on Fig. 2. A particular example is presented on Fig. 3, showing the bases used by Alice and Bob, who used 8 entangled qubits, which are seen in the protocol procedure.

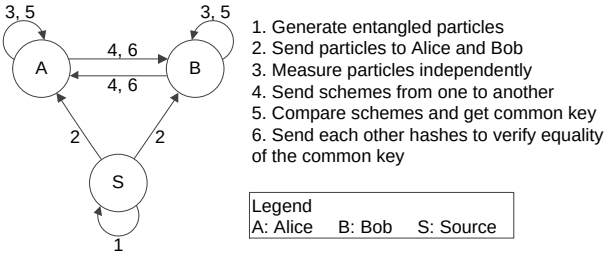


Fig. 2: E91 protocol general procedure

$$\begin{cases} B_0 = \{|0\rangle, |1\rangle\} \\ B_1 = \{|-\rangle, |+\rangle\} \end{cases}$$

(a)

$$\begin{cases} \frac{1}{\sqrt{2}} & (|1\rangle|-\rangle + |1\rangle|+\rangle) \\ \frac{1}{\sqrt{2}} & (|1\rangle|-\rangle - |0\rangle|-\rangle) \\ \frac{1}{\sqrt{2}} & (|0\rangle|-\rangle + |0\rangle|+\rangle) \\ \frac{1}{\sqrt{2}} & (|1\rangle|+\rangle - |0\rangle|+\rangle) \end{cases}$$

(b)

Received photon	$\|1\rangle\|-\rangle$	$\|1\rangle\|-\rangle$	$\|0\rangle\|-\rangle$	$\|1\rangle\|+\rangle$
Alice's scheme	B_0	B_1	B_0	B_1
Alice's values	1	0	0	1
Received photon	$\|1\rangle\|+\rangle$	$\|0\rangle\|-\rangle$	$\|0\rangle\|+\rangle$	$\|0\rangle\|+\rangle$
Bob' scheme	B_1	B_1	B_1	B_1
Bob's values	1	0	1	1
Coincidences		yes		yes
Key		0		1

(c)

Fig. 3: E91 protocol example [5]. (a) Bases. (b) 8 Entangled qubits. (c) Procedure

The amount of fluxes of this protocol is 5, which are presented on Table II, which indicates the communication channel, the flux and the time needed to achieve the flux transition (Period). In our case, the time for that transition is 1 and is taken as reference for implementation models.

Table II: Communication fluxes of E91 protocol

Channel	Flux	Period
Quantum	$Alice \overset{particles}{\leftarrow} Source \overset{particles}{\rightarrow} Bob$	1
Conventional	$Alice \overset{scheme}{\rightarrow} Bob$	1
Conventional	$Alice \overset{scheme}{\leftarrow} Bob$	1
Conventional	$Alice \overset{hash}{\rightarrow} Bob$	1
Conventional	$Alice \overset{hash}{\leftarrow} Bob$	1

Summary: 5 fluxes, total period 5

C. E91 protocol models of implementation

In this section 3 models of implementation are presented, which are the alternative ways to implement the step 1 to 3 of E91 protocol, in a traditional computer, according the quantum entanglement which is related with the sending of

particles from the source and the measuring that Alice and Bob make over the anti-correlated particles[6]. Each model of implementation is accompanied by one or more figures, where Alice, Bob and the Source (third person emitting particles) are pointed out by their initial letter of their names.

1) Model A

This model considers that the Source only sends undefined particles $|?\rangle$ to Alice. This way, undefined particles are sent to Alice first, and as long as Alice measures, she defines them and sends complementary particles to Bob (Fig. 4). The fluxes of communication in this model can be seen on Table III, where 6 fluxes are shown which corresponds to the original amount of fluxes plus one related to the original protocol. The second flux of particles from Alice to Bob increases the total period, considering the "sudden" achievement of quantum entanglement.

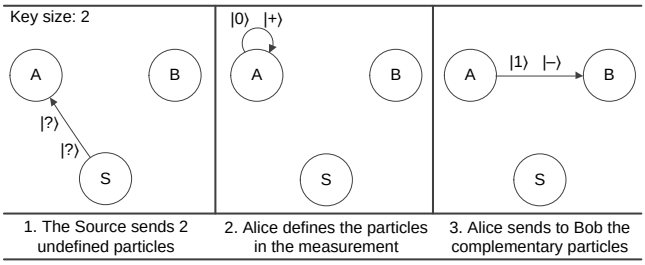


Fig. 4: Example of implementation model A

Table III: Communication fluxes of implementation model A

Flux	Period
$Source \xrightarrow{particles} Alice$	1
$Alice \xrightarrow{particles} Bob$	1
$Alice \xrightarrow{scheme} Bob$	1
$Alice \xleftarrow{scheme} Bob$	1
$Alice \xrightarrow{hash} Bob$	1
$Alice \xleftarrow{hash} Bob$	1

Summary: 6 fluxes, total period 6

2) Model B

This model proposes that the Source sends half of the particles to Alice and the other half to Bob, while they take measures, they send the complementary particle to each other (Alice or Bob). This is shown in Fig. 5. Fig. 6 presents a detailed example presents a detailed example of synchrony between Alice and Bob, for a 64 initial bits key, where Alice receives the first 32 particles and Bob the 32 left from the Source. And simultaneously Alice and Bob send each complementary particle in the corresponding position. With this, both can measure as continuously as possible, to represent how instantaneous quantum entanglement is, and to express clearly the uncertainty of measures. The amount of communication fluxes is 6 (see Table IV). These fluxes are equivalent to 4 ½**,** because the first 2 fluxes have a half duration, given that they are approximately parallel, achieving this way a lesser period than the original protocol. This protocol represents the entangled process of the communication, but the source does not perform that entanglement, which reduces its fidelity with the quantum entanglement.

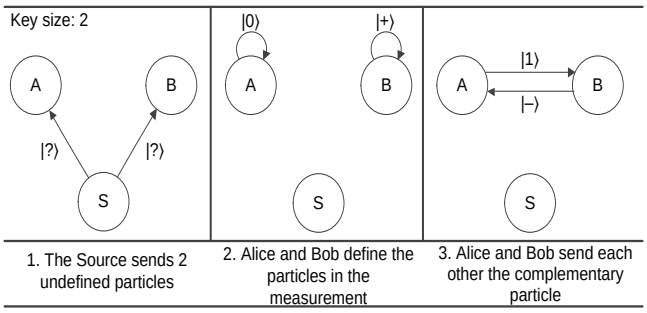


Fig. 5: Example of implementation model B

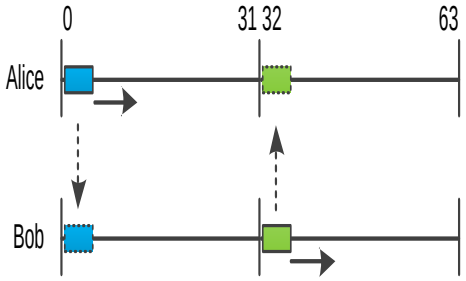


Fig. 6: Synchrony of implementation model B

Table IV: Communication fluxes of implementation model B

Flux	Period
$Alice \xleftarrow{particles} Source \xrightarrow{particles} Bob$	1/2
$Alice \xleftrightarrow{particles} Bob$	1/2
$Alice \xrightarrow{scheme} Bob$	1
$Alice \xleftarrow{scheme} Bob$	1
$Alice \xrightarrow{hash} Bob$	1
$Alice \xleftarrow{hash} Bob$	1

Summary: 6 fluxes, total period 4 ½

3) Model C

This third model proposes that the Source sends entangled particles, but already defined with a spin-up orientation $|\uparrow\rangle$ or spin-down $|\downarrow\rangle$. This is presented in Fig. 7, where a pair of anti-correlated particles are sent and then measured, considering $0 \Leftrightarrow |\downarrow\rangle$ and $1 \Leftrightarrow |\uparrow\rangle$. In this way for a 64 initial bits key, 64 particles are sent to Alice and 64 to Bob, sticking to the original protocol. The communication fluxes of this model are identical to the original one, with the only difference that the conventional cannel is used for all communication (see Table V). However, it discards the entanglement communication, because it considers the particle as already defined.

[6] Naming of particles that always have complementary state, that means for one particle $|\uparrow\rangle$ its anti-correlated will be $|\downarrow\rangle$, and vice-versa [2].

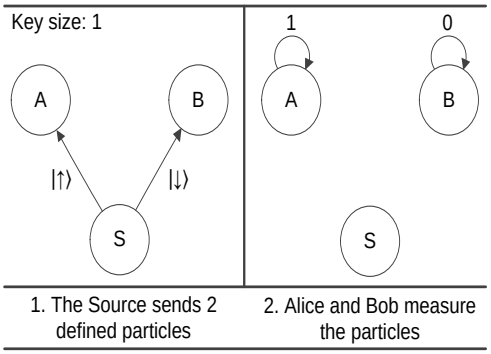


Fig. 7: Example of implementation model C

Table V: Communitation fluxes of implementation model C

Flux	Period
$Alice \xleftarrow{particles} Source \xrightarrow{particles} Bob$	1
$Alice \xrightarrow{scheme} Bob$	1
$Alice \xleftarrow{scheme} Bob$	1
$Alice \xrightarrow{hash} Bob$	1
$Alice \xleftarrow{hash} Bob$	1

Summary: 5 fluxes, total period 5

To choose a model, 2 parameters are considered, taking an ideal implementation as a reference. The values Low, Mid and High are used to qualify each parameter, where High is the best and Low the lesser ideal. Parameters are: (1) Fluxes' Fidelity, related to the degree of similarity to theoretic fluxes and (2) Synchronization simplicity, according the simplicity to synchronize the fluxes and achieve the desired behavior. The comparison is presented in Table VI.

Table VI: Comparison of 3 implementation models

	Fluxes' fidelity	Synchronization simplicity
Model A	Low	Mid
Model B	Mid	Low
Model C	High	Mid

Considering the information summarized in Table VI, Model of implementation C was chosen.

IV. Quantum Cryptography Protocols Simulation

The prototype considers BB84 Simulator, as E91 Simulator. Therefore, this section is focused on the functionality extension of BB84 Simulator [5] and the design and implementation of Simulator E91 [7] based on the structure of BB84 Simulator to implement the simulation of E91 protocol.

A. BB84 Simulator module

The original BB84 Simulator (Fig. 8) is focused only on the simulation, generating a shared key between Alice and Bob (called sender and receiver) reason why to add a module that extends its functionality, sending encrypted messages with the shared key. Considering that the key is identical between the participants of the communication, the symmetric encrypting algorithm AES was used. This function receives the name of "Message Sender Module".

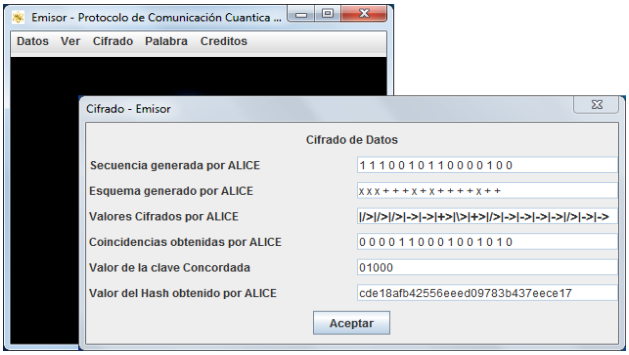


Fig. 8: Graphical user interface of original BB84 Simulator – Alice (sender).

The requirements are a description of the application needs. The following, indicates functional and non-functional requirements of the module.

Functional requirements: indicate what the system must do.

(a) The Sender must encrypt and send the messages using the shared key.

(b) The receiver must receive and decrypt the messages using the shared key.

Non-functional requirements: restrictions or quality requirements that condition the system.

(a) Encrypt and decrypt messages using the AES symmetric cryptography algorithm.

The use case[7] and other diagrams were designed under these requirements. The implementation was made using Java language and its RMI Java technology, due to the fact that the original simulator uses the same technologies. A problem was the impossibility of using keys higher than 128 bits, which is a legal restriction to the environment of Java execution. However, this restriction can be overcome as explained in [29], but must be applied in every host, reducing the portability of the software.

An execution example of this module, after configuring the simulator and generating a key, is presented in Fig. 9, where Alice sends a message to Bob, encrypting the message with the previously generated key.

B. E91 Simulator

The implementation of E91 Simulator is based on Implementation model C, previously presented. The following are previous definitions:

- Base: Both Alice and Bob have their own bases, while Alice has the 0, 45 and 90 degree bases; Bob uses the 45, 90 and 135 degree bases [30], which are represented in Table VII.
-

Table VII: Bases used by Alice and Bob.

	Bases
Alice	(−) (/) (|)
Bob	(/) (|) (\)

[7] The Use Case diagram represents the way an actor interacts with the system, in the context of Software Engineering [28].

(a) (b) (c)

Fig. 9: Message sent with BB84 simulator. (a) Alice introduces a message and sends it. (b) Bob receives the notification of the message reception. (c) Bob visualizes the message.

- Scheme: is a collection of symbols that represent the bases chosen by Alice or Bob. A valid scheme for Alice would be (/)(|)(/)(–)(|) and a valid scheme for Bob can be (/)(|)(\)(|)(\) .
- Particle: is the quantum entity, its symbol will depend on its spin, $|\uparrow\rangle$ for spin-up and $|\downarrow\rangle$ for spin-down.
- Entangled particles: it represents the particles under the physical phenomenon of quantum entanglement, but in a reversed way, which means that those are anti-correlated. This way, if Alice receives the particles $|\uparrow\rangle$ $|\downarrow\rangle$ $|\uparrow\rangle$ $|\downarrow\rangle$ $|\uparrow\rangle$ Bob must have the particles $|\downarrow\rangle$ $|\uparrow\rangle$ $|\downarrow\rangle$ $|\uparrow\rangle$ $|\downarrow\rangle$.

Considering previous definitions, the simulator requirements are the following:

Functional requirements

(a) To generate the sequence of anti-correlated particles to be transmitted.

(b) To generate the scheme that will be used to measure the particles.

(c) To allow the transmission of particles through a communication channel.

(d) To allow the measure of the particle sequence.

(e) To allow the transmission of the scheme through a communication channel.

(f) To allow the comparison of the used schemes.

(g) To allow the exchange of hashes of keys in common.

Non-functional requirements

(a) The system must be able to work in a distributed way.

(b) The system must be able to recover itself from input data errors.

(c) The system will present an environment based on windows.

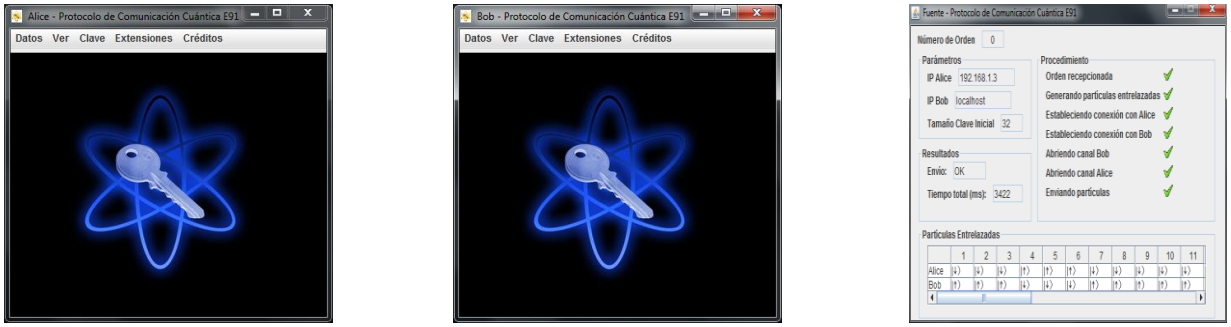
(d) The system will present a simple interface.

(e) The system will use the memory in an adequate way.

The use case and other diagrams where designed keeping in mind these requirements. For matters of implementing, the same technology that “Message Sending Module” of BB84 Simulator was used. Once the implementation is finished, 3 graphical user interfaces were obtained, corresponding to Alice, Bob and the Source, as it is shown in Fig. 10.

An example of the E91 Simulator execution is presented. The participants work in different machines: Alice uses Windows 7 (IP 192.168.1.2); Bob uses Ubuntu 12.04 (IP 192.168.1.5) and the Source also uses Ubuntu 12.04 (IP 192.168.8). The process begins when the simulator is configured as follows: a) Bob initiates rmiregistry, b) Bob introduces the key initial length, c) Bob introduces Alice’s IP address and (d) Bob indicates that it is ready. The Bob’s configuration is also valid for Alice. With the Source there is no need to enter any parameter, as it is shown in Fig. 11, the Source starts in the highest part and rmiregistry command in the one below, exposing its IP address.

Then, the generation of the key can be seen on Fig. 12, where is shown the procedure after selecting the option Initiate Generation (Iniciar generación) in Alice’s key menu. Finally, if intended, Alice can send an encrypted message with the generated key, being identical to the process of using the module developed for BB84 Simulator (Fig. 9).



(a) (b) (c)

Fig. 10: Graphical user interfaces of E91 Simulator. (a) Alice. (b) Bob. (c) Source.

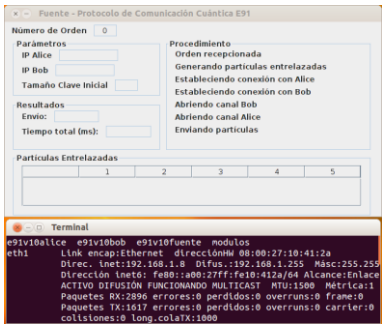
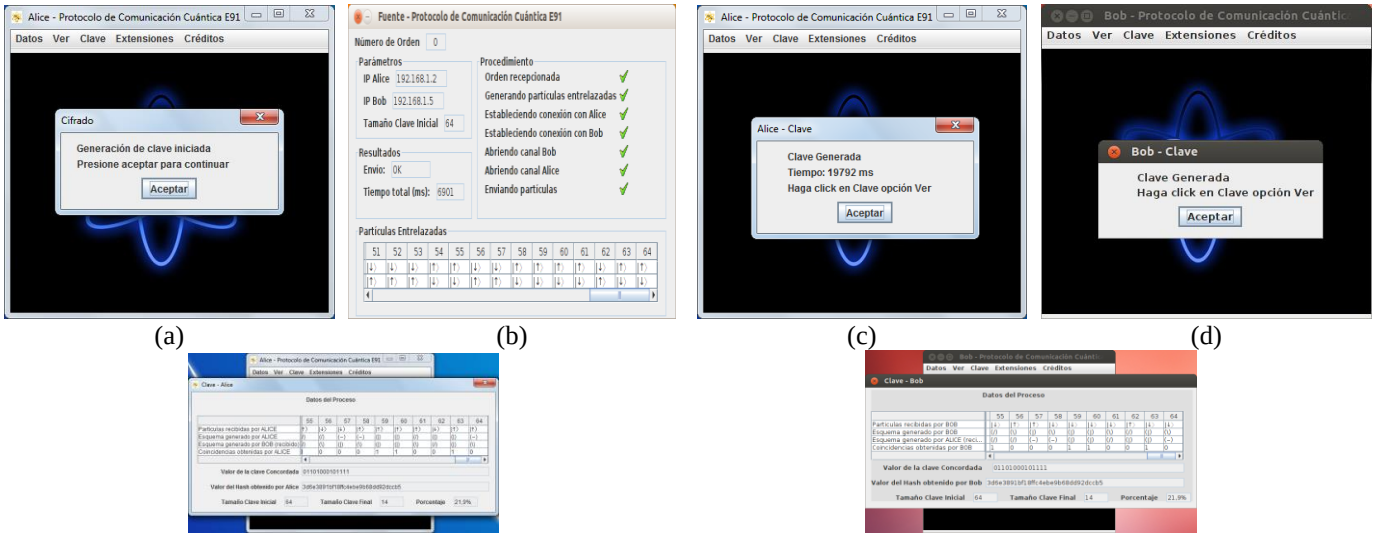


Fig. 11: Source E91 simulator setting.



(a) (b) (c) (d)

(e) (f)

Fig. 12: E91 simulator key generation. (a) Alice notifies starting the key generation. (b) The Source processes the order and sends the particles. (c) Alice notifies ending the key generation. (d) Bob notifies ending the key generation. (e) Alice providing details of the process. (f) Bob providing details of the process.

V. PROTOTYPE TESTS

In this section, the test method used to evaluate the prototype are presented, then with the evaluation results, BB84 and E91 simulators which are part of the prototype, are analyzed and compared.

A. Testing Methodology

The implemented prototype, considers the BB84 Simulator and E91 Simulator, which will be examined with a case test battery. The concepts involved are detailed as follows:

1. Simulator: it represents the testing simulator (BB84 or E91).

2. Case: are the experimental scenarios, these are structured with 3 different network architectures, only receiving a few adjustments according to the simulator being used. The difference is noted by the number of participators, being 2 in BB84 Simulator (Alice and Bob) and 3 in the E91 Simulator (Alice, Bob and the Source). Case 1 corresponds to a wireless network (WLAN – Wireless Local Area Network); Case 2 corresponds to a local network (LAN – Local Area Network) and Case 3 to different local networks, as seen on Fig. 13.

3. Test: represents the number of initial bits with which the generation of a shared key begins. Its initial values are: 8, 16, 32, 64, 128, 512, 1024 and 2048 bits.

4. Test case: it represents the conjunction of the simulator, the case and the test. An example of this test case will be with BB84 Simulator for case 1 with 8 initial bits. Each

   test case is repeated 3 times, using the arithmetic averages for later analysis. The obtained values in each test are time and size of the final key.

For an optimal experimentation, in every machine the non-essential processes load is minimized. The data obtained in the test cases is observable in Table VIII, for both simulators.

B. Prototype analysis and comparison

The first big difference between these protocols is visible in Fig. 14, where is observed the final key size obtained respect to the initial bits in the 3 cases for both protocols, where the BB84 Simulator rounds 50% of the initial bits (three high curves) while E91 Simulator rounds 22% of the initial bits (three low curves). Seeing the detail of the BB84 Simulator, it is revealed that convergence begins at 32 initial bits, becoming clearly visible from 256 bits on. On the other hand, E91 Simulator converges from 64 initial bits. The final key of 50% of the initial bits for BB84 Simulator matches the analysis made by its creators and the final key of 22% of the initial bits for E91 Simulator is approximated with a difference of 11% to the theoretical 33% proposed by Ilic [30].

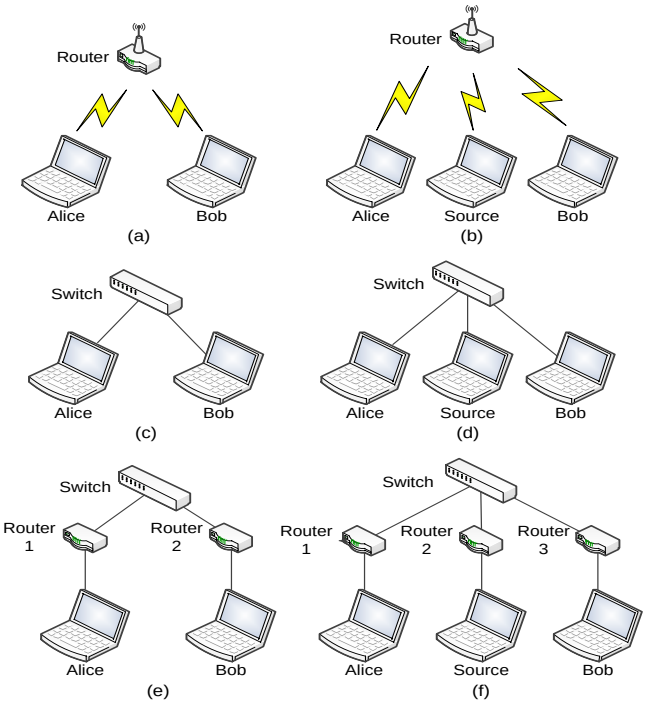


Fig. 13: Cases in test methodologies. (a) Case 1 BB84 Simulator. (b) Case 1 E91 Simulator. (c) Case 2 BB84 Simulator. (d) Case 2 E91 Simulator. (e) Case 3 BB84 Simulator. (f) Case 3 E91 Simulator.

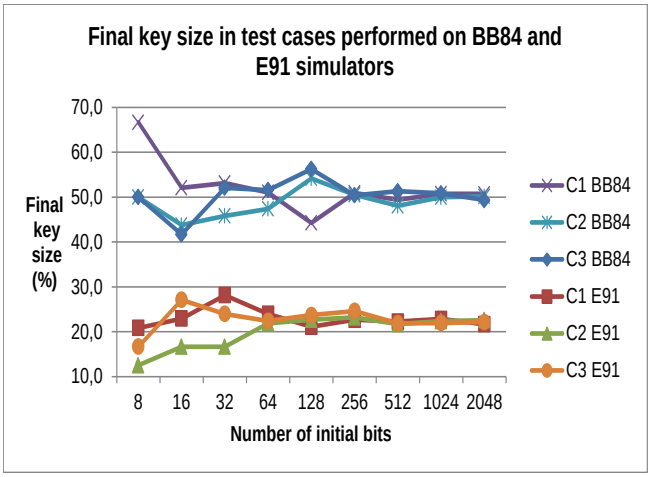


Fig. 14: Final key size in test case performed on BB84 and E91 simulators.

Fig. 15 shows times in the test cases for both simulators. In every test case on both simulators, they present similar behavior and values. The former, is due to the fact that both simulators have the same software structure, and besides the amount of information flux (Table IX) is identical for both, having 5 fluxes. The only difference detected was on the first flux where the Source for the E91 Simulator intervenes. A detailed analysis evidences that "Case 1: wireless network" for both simulators, has a general major time; due to its nature, it is vulnerable to interference and to have high latency (temporary delays within a computer network). From this finding, it can be concluded from both simulators that "Case 1: wireless network" is the slowest; followed by "Case 3: different local networks" with a higher speed, and finally with a slight difference, "Case 2: local network" which is the best in terms of network communication performance.

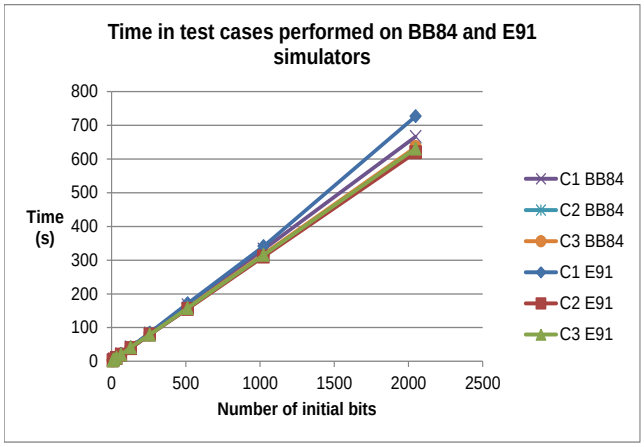


Fig. 15: Time in test cases performed on BB84 and E91 simulators.

Table IX: Communication fluxes in both simulators.

BB84 Protocol	E91 Protocol
$Alice \xrightarrow{photons} Bob$	$Alice \xleftarrow{particles} Source \xrightarrow{particles} Bob$
$Alice \xrightarrow{scheme} Bob$	$Alice \xrightarrow{scheme} Bob$
$Alice \xleftarrow{scheme} Bob$	$Alice \xleftarrow{scheme} Bob$
$Alice \xrightarrow{hash} Bob$	$Alice \xrightarrow{hash} Bob$
$Alice \xleftarrow{hash} Bob$	$Alice \xleftarrow{hash} Bob$

After analyzing and comparing the simulators, two main aspects are to be highlighted:

1. BB84 Simulator has a final key size of 50% of the initial bits; E91 Simulator has a 22%. The first one resulted as expected by its creators, but in the case of E91 Simulator, it leaves a gap of an 11% in relation to the 33% theoretical proposed by Ilic [30]. This makes the E91 protocol safer, because it discards a higher amount of bits. On the other hand, the technological limitations make the implementation of BB84 protocol more viable, due to the fact that it does not require quantum entanglement. This is already in use for quantum networks, at government level [19] and at corporate level by Toshiba [20].
2. Average times for BB84 Simulator as well as for the E91 Simulator are very similar in every test case. Due to the use of the same software structure and the same amount of communication fluxes. Still, for a real implementation of both protocols, time should be very similar, because the only difference between both protocols is that E91 protocol uses quantum entanglement, keeping the same quantity of communication fluxes.

Finally, it must be considered time presented is merely referential, because it is a simulation on channels and classic computers, in a real implementation with quantum behavior devices time should be less than presented. Therefore, with the presented analysis is considered that simulators are a useful and efficient tool that allows demonstrating these protocols behavior.

Table VIII: Test case data with both simulators

		BB84 Simulator			E91 Simulator		
	No. of initial bits	No. of final bits	Generation time (s)	Final bits portion (%)	No. of final bits	Generation time (s)	Final bits portion (%)
Case 1 Wireless network	8	5,3	3,0	66,7	1,7	5,3	20,8
	16	8,3	5,2	52,1	3,7	7,3	22,9
	32	17,0	10,4	53,1	9,0	11,2	28,1
	64	32,7	21,1	51,0	15,3	22,2	24,0
	128	56,7	42,4	44,3	27,0	41,3	21,1
	256	130,3	83,1	50,9	58,0	84,9	22,7
	512	253,0	166,6	49,4	114,0	171,8	22,3
	1024	519,7	332,2	50,7	233,7	341,4	22,8
	2048	1039,0	666,1	50,7	443,3	726,4	21,6
Case 2 Local network	8	4,0	2,6	50,0	1,0	2,5	12,5
	16	7,0	5,0	43,8	2,7	4,9	16,7
	32	14,7	9,9	45,8	5,3	9,8	16,7
	64	30,3	19,8	47,4	14,0	19,5	21,9
	128	69,3	39,4	54,2	29,0	38,8	22,7
	256	129,3	78,7	50,5	59,3	79,0	23,2
	512	246,0	157,3	48,0	111,0	155,5	21,7
	1024	511,7	314,5	50,0	229,0	310,4	22,4
	2048	1026,7	628,7	50,1	462,0	619,4	22,6
Case 3 Different local networks	8	4,0	2,6	50,0	1,3	2,7	16,7
	16	6,7	5,1	41,7	4,3	6,1	27,1
	32	16,7	10,1	52,1	7,7	11,0	24,0
	64	33,0	20,0	51,6	14,3	19,8	22,4
	128	72,0	39,9	56,3	30,3	41,5	23,7
	256	129,0	79,6	50,4	63,0	79,0	24,6
	512	262,7	159,1	51,3	112,0	157,9	21,9
	1024	520,7	317,9	50,8	224,7	315,4	21,9
	2048	1010,3	635,1	49,3	453,3	630,8	22,1

VI. Conclusion

It is necessary to understand the principles of Quantum mechanics, to fully grasp the concepts of Quantum Cryptography and Quantum Computing, because it generates new rules, with no classic equivalents, which provide the grounds for security of Quantum Cryptography.

It can be seen that some cryptography algorithms used nowadays have been violated, as RSA and AES. Also, the advances of Quantum computing as D-Wave quantum computer and currently use of quantum networks have been revised. These technologies are still to be improved, since they present certain flaws or do not fulfill the ideal expectations. However, the existence of the quantum algorithm Shor has proved its potential to exploit the weakness of traditional asymmetric cryptography.

Quantum cryptography protocols such as BB84 and E91 contribute to solve the problem of keys distribution, which is currently achieved by the use of asymmetric cryptography algorithms, the same ones that can be disabled with the development of quantum computing and its algorithms. Quantum cryptography protocols use both conventional and quantum channels to distribute the key. BB84 protocol was the first to propose the use of quantum mechanics principles, which make it theoretically unbreakable. The previously mentioned idea was utilized by protocol E91, adding the use of quantum entanglement.

During the prototype design, the most complex task was the representation of the quantum entanglement, considering that it has no classic equivalent. Several models were proposed, trying to represent quantum behavior as faithfully as possible. Once selected, and incorporated to the software design stage, a clearer view of its consequent implementation was obtained.

The tests stage was performed under a methodology that guaranteed the quality of the measures.

The results obtained, showed that BB84 protocol is stable at around 50% of the initial bits, for keys of 256 initial bits and higher; whereas E91 protocol, converges at a 22% of the initial bits, for keys of 64 initial bits and higher. Regarding to the general timing, both simulators performed the task at similar amounts of time, due to the fact that both use the same quantity of information fluxes and the same software structure.

Therefore, the analysis of the obtained results allowed validating the theoretical final key's size for BB84 protocol. However, in the case of E91 protocol, it was shown a discrepancy below the theoretically size proposed. For this reason, E91 protocol is considered to be safer. However, it faces technological difficulties of computing resources, because it uses quantum entanglement.

The analysis and comparison of a simulation prototype of quantum cryptography protocols is considered achieved. This showed that E91 protocol provides a higher security level.

VII. Future Work

This work presented the design and development of two simulators, allowing understanding the principles of quantum cryptography algorithms, in this case protocols BB84 and E91. Under this context, future work considers the following aspects:

1. To continue this line of investigation, it is necessary to have a study and analysis of other quantum cryptography protocols, which will allow obtaining deeper and more reliable knowledge of these algorithms strengths; and in this way, formalizing through implementation and

simulation of the real behavior for these cryptographic algorithms.

2. Once the global context of this investigation is achieved, it is proposed to work in a multidisciplinary team to achieve the implementation of quantum cryptography protocols.

ACKNOWLEDGMENT

This work was supported by the Universidad de Tarapacá (Proyecto Mayor No. 8721-12 (2012)) and the Área de Ingeniería en Computación e Informática. Arica, Chile.

REFERENCES

[1] C. H. Bennet and G. Brassard, "Quantum cryptography: public key distribution and coin tossing," in International Conference on Computers, Systems & Signal Processing, Bangalore, India, 1984.

[2] A. K. Ekert, "Quantum cryptography based on Bell's theorem," in Phys. Rev. Lett. [Online], vol. 67, no. 6, pp. 661–663, 1991. Available: http:dx.doi.org/10.1103/PhysRevLett.67.661

[3] C. H. Bennet, F. Bessette, G. Brassard, L. Salvail and J. Smolin, "Experimental quantum cryptography," Journal of Cryptology [Online], vol. 5, no. 1, pp. 3–28, 1992. Available: http:dx.doi.org/10.1007/BF00191318

[4] V. Scarani, A. Acín, G. Ribordy and N. Gisin: "Quantum Cryptography Protocols Robust against Photon Number Splitting Attacks for Weak Laser Pulse Implementations," Phys. Rev. Lett. [Online], vol. 92, no. 5, American Physical Society, 2004. Available: http:dx.doi.org/10.1103/PhysRevLett.92.057901

[5] M. Pinto Bernabé, "Simulación de un Protocolo de Comunicación Cuántica entre Procesos en un Ambiente Distribuido," MSc thesis, Universidad de Tarapacá, Chile, 2010.

[6] L. Cáceres Alvarez, M. Pinto Bernabé and P. Collao Caiconte, "Implementación de un Simulador de Criptografía Cuántica – Protocolo BB84," Jornadas Chilenas de la Computación, Chile, 2013.

[7] L. Cáceres Alvarez, R. Fritis Palacios and P. Collao Caiconte, "Desarrollo de un simulador para el protocolo de criptografía cuántica E91 en un ambiente distribuido," Ingeniare, Rev. chil. ing [Online], vol. 23, no. 2, pp. 245–258, issn: 0718-3305, 2015. Available: http://dx.doi.org/10.4067/S0718-33052015000200009

[8] T. Kleinjung, K. Aoki, J. Franke, A. K. Lenstra, E. Thomé , J. W. Bos, P. Gaudry, A. Kruppa, P. L. Montgomery, D. A. Osvik, H. te Riele, A. Timofeev and P. Zimmermann, "Factorization of a 768-bit RSA Modulus," Advances in Cryptology – CRYPTO 2010 [Online], pp. 333–350, Springer Berlin Heidelbergdoi, 2010. Avaible: http:dx.doi.org/10.1007/978-3-642-14623-7_18

[9] A. Pellegrini, V. Bertacco and T. Austin, "Fault-based attack of RSA authentication," Design, Automation Test in Europe Conference Exhibition (DATE) [Online], pp. 855–860, IEEE, Dresden, 2010. Available: http:dx.doi.org/10.1109/DATE.2010.5456933

[10] National Institute of Standard, "ADVANCED ENCRYPTION STANDARD (AES)," Federal Information, Processing Standards Publication 197 [Online], 2010. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf [October 9, 2014].

[11] A. Bogdanov, D. Khovratovich and C. Rechberger, "Biclique cryptanalysis of the full AES," Advances in Cryptology - ASIACRYPT 2011 [Online], vol. 7073, pp. 344–371, 2011, Springer Berlin Heidelberg. Available: http:dx.doi.org/10.1007/978-3-642-25385-0_19

[12] F. Acero, "? Fue buena idea usar AES256 con el archivo INSURANCE de Wikileaks?," [Online], 2010. Available: http://www.kriptopolis.org/wikileaks-insurance-aes-256 [April 3, 2013].

[13] Top 500 Supercomputer Sites, [Online]. Available: http://www.top500.org [August 14, 2013].

[14] D-Wave Systems Inc., "The D-Wave Two system," [Online]. Available: http://www.dwavesys.com/en/products-services.html [August 7, 2013]

[15] S. Anthony, "Quantum computer finally proves its faster than a conventional PC, but only just," [Online], 2013. Available: http://www.extremetech.com/extreme/155380-quantum-computer-wins-first-ever-speed-test-against-a-conventional-intel-pc [July 7, 2013].

[16] H. Nishimori, "Quantum Annealing," [Online], 2014. Available: http://www.stat.phys.titech.ac.jp/ nishimori/QA/q-annealing_e.html [October 10, 2014].

[17] R. Fernandez Delicado, D. Bellver Cabello and I. Lloro Boada, "The quantum cryptograpy: Communication and computation," Acta Astronautica [Online], vol. 57, no. 2-8, pp. 348–355, 2005. Available: http://dx.doi.org/10.1016/j.actaastro.2005.03.021

[18] P. Marks, "Google demonstrates quantum computer image search," NewScientist [Online], 2009. Available: http://www.newscientist.com/article/dn18272-google-demonstrates-quantum-computer-image-search.html [July 7, 2013].

[19] R. J. Hughes, J. E. Nordholt, K. P. McCabe, R. T. Newell, C. G. Peterson and R. D. Somma, "Network-Centric Quantum Communications with Application to Critical Infrastructure Protection," [Online], 2013. Available: http://arxiv.org/abs/1305.0305v1

[20] B. Fröhlich, J. F. Dynes, M. Lucamarini, A. W. Sharpe, Z. Yuan and A. J. Shields, "A quantum access network," Nature [Online], vol. 501, pp. 69–72, 2013. Available: http://dx.doi.org/10.1038/nature12493

[21] Integrated Project SECOQC, "Development of a Global Network for Secure Communication based on Quantum Cryptography," [Online], 2010. Available: http://www.secoqc.net [January 7, 2015]

[22] S. N. Molotkov, "On the vulnerability of basic quantum key distribution protocols and three protocols stable to attack with "blinding" of avalanche photodetectors," Journal of Experimental and Theoretical Physics [Online], vol. 114, no. 5, pp. 707–723, 2012. Available: http://dx.doi.org/10.1134/S106377611203017X

[23] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM J.Sci.Statist.Comput. [Online], vol. 26, 1996. Available: http://arxiv.org/abs/quant-ph/9508027v2

[24] M. Hellman and H. Scolnik, "El desarrollo de la criptografía cuántica descifrará todas las claves actuales," Diario El País [Online], 2008. Available: http://elpais.com/diario/2008/01/17/ciberpais/1200537623_850215.html [July 7, 2013].

[25] A. Díaz Caro and J. Samborski Forlese, "Brevísima Introducción a la Computación Cuántica," Departamento de Ciencias de la Computación [Online], Universidad Nacional de Rosario, 2006. Available: http://www.fceia.unr.edu.ar/diazcaro/QC/Brevisima.Introduccion.pdf [August 12, 2013].

[26] N. R. Wagner, "The Laws of Cryptography with Java Code," [Online], pp. 82–83, 2003. Available: http://www.cs.utsa.edu/wagner/lawsbookcolor/laws.pdf [October 10, 2014].

[27] M. Baig, "Criptografía Cuántica," Grup de Física TeòÃ²rica - IFAE [Online], Facultad de Ciencias, Universidad Autónoma de Barcelona, 08193 Bellaterra, 2001. Available: http://giq.ifae.es/EducationalMaterial/Cripto.pdf [August 13, 2013].

[28] R. S. Pressman, "Ingeniería del Software, Un enfoque práctico," 6th ed, pp. 149, 2005. Mc Graw Hill, México, isbn: 970-10-5473-3.

[29] B. Koops, "Crypto Lay Survey," [Online], 2013. Available: http://www.cryptolaw.org [October 13, 2013].

[30] N. Ilic, "The Ekert Protocol," Department of Physics [Online], University of Waterloo, Canada, 2004. Available: http://www.ux1.eiu.edu/ nilic/Nina%27s-article.pdf [April 5, 2015].

Luis Cáceres Alvarez received the MSc degree in Mechanics Engineering in 1999 from University Federal of Santa Catarina of Brazil and the Ph.D. degree in Computer Science in 2004 from the University Federal of Santa Catarina of Brazil. He is a full time lecturer and courses' director in the Computer Engineering Department at the Universidad de Tarapacá of Chile. His current research is focused on data security and his expertise areas are data security, network computers and distributed systems.

Patricio Collao Caiconte received his licensed degree in engineering science and MSc degree in software engineering from Universidad de Tarapacá, Chile.