International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:12, No:10, 2018

FPGA Implementation of the BB84 Protocol

Jaouadi Ikram, Machhout Mohsen

***Abstract*—**The development of a quantum key distribution (QKD) system on a field-programmable gate array (FPGA) platform is the subject of this paper. A quantum cryptographic protocol is designed based on the properties of quantum information and the characteristics of FPGAs. The proposed protocol performs key extraction, reconciliation, error correction, and privacy amplification tasks to generate a perfectly secret final key. We modeled the presence of the spy in our system with a strategy to reveal some of the exchanged information without being noticed. Using an FPGA card with a 100 MHz clock frequency, we have demonstrated the evolution of the error rate as well as the amounts of mutual information (between the two interlocutors and that of the spy) passing from one step to another in the key generation process.

***Keywords*—**QKD, BB84, protocol, cryptography, FPGA, key, security, communication.

I. Introduction

THE need to secure communication is eternal. Since the earliest civilization, people have sought to develop effective resources to preserve their territory and their power. They sought to ensure the security of communications by inventing and implementing codes to hide the exchange of important information.

The first encryption code used was the Caesar code used by Julius Caesar to secure his correspondence. This code was based on the mono-alphabetic substitution technique. It consists of shifting each letter of the alphabet a few notches to the right or the left. Although this technique does not appear to be robust, the low literacy, at that time, made it sufficiently effective.

Many others encryption systems have been proposed whose objective was always to guarantee the security and confidentiality of the communication. The history of codes is a persistent and eternal battle between coders and code breakers. However, the first civilizations did not really use codes, but rather techniques to hide the existence of the message.

In January 1983, August Kerckhoffs defined the principles of modern cryptography in his article "Modern Cryptography" published in the "Journal of Military Sciences" [1]. He asserts that the security of a cryptosystem must rest only on the secrecy of its key and that all the other parameters must be supposed publicly known. These principles were formulated later by Claude Shannon under the name of "The maxim of Shannon" [2].

Jaouadi Ikram (PhD student) is with the National Engineering School of Tunis ENIT, Tunisia, Researcher with Electronics and Microelectronics laboratory in Sciences Faculty of Monastir, BP 37, 1002 TUNIS LE BELVEDERE, Tunisia (e-mail: ikramjaouadi_2006@ yahoo.fr).

Machhout Mohsen (Lecturer) is with the Sciences Faculty of Monastir, Electronics and Microelectronics laboratory, Avenue of the environment 5019, Tunisia (e-mail: mohsen.machhout@fsm.rnu.tn).

In 1984, Shannon demonstrated that the unconditional security of a cryptographic protocol depends on its key length which must be at least as long as the message to encode. This constraint was the basis of one-time-pad protocols. Such a protocol has unconditional security provided that each encryption key is used only once. If the Shannon criteria are not respected, the protocol security cannot be formally demonstrated. The designer must ensure that the key is long enough to prevent an exhaustive attack to test all possible keys.

The basic element of any cryptographic system is the encryption key. This parameter was the "weak link" in classical cryptographic systems. Indeed, even for the one-time-pad protocol, considered to be the most perfect encryption code, the main problem was the exchange of the key. Therefore, this code provides a means to secure the transport of this key before its use. For example, to code the "red phone" between the USA and Russia, Washington was careful to carry the key in diplomatic bags, a non-robust solution. In case the key has been stolen, we can intercept the data flow and without it being tagged or touching the integrity of the message.

The appearance of quantum mechanics revealed procedures for controlling data transmission and restricting intrusions. Quantum cryptography exploits the principles of quantum mechanics to ensure the confidentiality of exchanges. It does not consist on encrypting the information transmitted, but rather on establishing a perfectly secret key that can be used with classic cryptosystems. Quantum cryptography can be considered as a complement to classical cryptography.

In 1984, Bennet and Brassard proposed the first QKD protocol named the BB84 protocol [3]. It was at the base of various discrete variable experiments by single photon coding [4], [5].

In this paper, we propose an implementation of the BB84 protocol on a FPGA platform. For this, we start by suggesting an algorithm for this protocol. Later, we present implementation steps and components needed.

II. The BB84 Protocol

A. Algorithm

This protocol was originally proposed by polarization coding. It proposes to use a quantum channel for quantum transmission and a classic one for public discussion. We apply polarization states constituting two orthonormal unbiased bases:

- The rectangular base $\{|\rightarrow\rangle, |\uparrow\rangle\}$
- The diagonal base $\{|\nwarrow\rangle, |\nearrow\rangle\}$

Table I represents the different polarization bases.

TABLE I
POLARIZATION BASES

Base	0	1
⊕	→	↑
⊗	↗	↖

The protocol is triggered when Alice randomly chooses a sequence of qubits, encodes each one with a randomly selected based among the four predetermined bases, and finally, passes her measurement results to Bob. That is why this protocol is called "measure and send" protocol. Then, when receiving the polarized qubits transmitted by Alice, Bob proceeds to their measurement in order to determine the initial state of each received qubit. He uses for the measurement a sequence of bases chosen randomly.

The protocol is composed of five major phases:

- Quantum transmission phase

Alice randomly chooses a sequence of symbols among {0, 1} and a sequence of bases among {⊕, ⊗}, codes each symbol with the correspondent base and finally, send polarized qubits to Bob through the quantum channel.
Bob receives successively transmitted qubits and measures each one with a randomly chosen base among {⊕, ⊗}.

- Reconciliation phase

Also called sifting phase, the two interlocutors use the public channel to compare their bases choices. They only keep bits corresponding to consistent choices. The chains obtained represent the raw keys or sifted keys.

- Error rate estimation

Alice and Bob determine the error rate on their raw keys. If the error rate is very high, it is a sign of spy presence in the quantum channel. So, the two interlocutors have to proceed to error correction.

- Error correction and privacy amplification

In this phase, Alice and Bob aim to reduce the error rate by using a correction algorithm and to scramble the spy information, by applying a hashing technique to the corrected keys. The key obtained after this process is perfectly secret.

III. SPY STRATEGY

As we have already mentioned, a QKD protocol uses two channels: a quantum channel to which the spy has access with the ability to manipulate the information circulating on this channel, and a classic channel to which the spy can access without being able to maneuver the data exchanged via this channel.

The security of a QKD protocol relies mainly on the quantum non-cloning theorem [6] that prevents the spy from duplicating the intercepted quantum states. Thanks to this theorem, any action taken by the spy on the data exchanged is remarkable by the two interlocutors. However, there are several strategies for attacking. In this paper, we will present Intercept/Resend attacks which consists of:

- Intercept a fraction η of the data transmitted by Alice.
- Measure each element of the fraction η with a randomly selected base among {⊕, ⊗}.
- Substitute the state transmitted by Alice to Bob by the measurement result of Eve. Note here that the spy did not violate the non-cloning theorem.

Eve's interest is not to choose the right base but rather to choose the same basis as Bob. There are then four possible situations:

- Bob and Eve have chosen the same base and it is the right one.
- Bob and Eve have chosen the same base and it is the wrong one.
- Only Bob has chosen the right base.
- Only Eve has chosen the right base.

In two situations, the Eve base is compatible with Alice's coding. Therefore, she can clearly determine the transmitted state without marking her presence. For other situations, she will have no significant information on the data transmitted. For Bob, who receives the re-emitted states by Eve, one in every two times, he will have a wrong result even when using the same coding base as Alice.

To summarize, Eve can only obtain $\frac{\eta}{2}$ of all the data transmitted by Alice. But this increases Bob's error rate by $\frac{\eta}{4}$.

In this section, we proposed an algorithm for the BB84 protocol. We chose the version of polarization coding. We now move to its implementation on an FPGA platform.

IV. FPGA IMPLEMENTATION

For implementation, we used the Nexys4 DDR board [7] with a clock frequency of 100 Mhz.

The basic step is Alice's choice of symbols and bases to prepare the polarized qubits as well as the choice of bases for Eve and Bob. For random sequences, a random number generator is required.

In our design, we used a pseudo-RNG defined by an LFSR register [8], [9]. LFSRsare used in cryptography [10], [11] as an alternative to real RNGs but should be used with caution. There are various versions of the LFSR registry. We chose LFSRs in Fibonacci mode that strictly apply the definition of an LFSR.

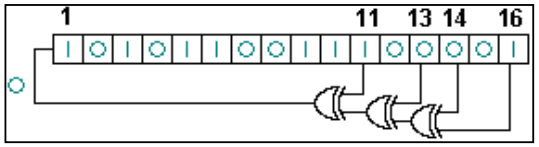

Fig. 1 16-bit Fibonacci LFSR

We implemented a 16-bit LFSR registry. We will then have 2^{16} possible combinations. Since we need several PRNGs (Alice bases, Alice symbols, Eve bases, Bob bases), we will use several LFSRs with the same algorithm and playing on the taps (positions of the bits that determine the next state).

The system can be described as follows:

- Quantum communication

At each instant t, the Alice FPGA randomly chooses a qubit and a polarization base, and applies a measurement to define the polarized state. The basis and qubit selections are downloaded to the RAM.

The FPGAs Bob and Eve choose at each instant t their bases of polarization. These databases are saved in RAMs.

Eve intercepts the transmitted state on the channel, reads from the RAM the corresponding polarization base and applies her measurement. She then replaces what she intercepted by the result of her measurement.

At each instant t, Bob receives the polarized qubit transmitted, reads from the RAM the corresponding polarization base and applies his measurement. Then he saves obtained results in the RAM.

Here we have to note that as RAM memory, we used FIFO (First In First Out) memories to save data at each stage of the protocol.

- Public discussion

Bob, via a classic channel, sends to Alice his choice of bases; she downloads them to the FPGA. Recall that for the classic channel, the spy can only follow the exchanges without interacting.

Alice's FPGA begins a comparison to determine common positions for establishing the raw keys. On her part Eve, being able to see the choices of Bob's bases, also proceeds to define her raw key.

- Errors corrections

Alice and Bob proceed to correct transmission errors in order to decrease the error rate and Eve information. So, they apply an algorithm for correction. In our implementation, we corrected errors by sacrificing part of the raw keys. The methods consist of choosing the same sub blocks from the two raw keys, then to estimate the error rate of these sub blocks and finally delete them from the keys.

If the error rate is still high, they have to repeat the process. Else, they can generate the secret key. This method seems to be simple to implement. However, the final key length is decreased compared to raw keys.

- Privacy amplification

The final key is supposed to be secret. However, the spy's access to the public channel leads us to believe that he has some of the corrected key. In this case, Alice and Bob will switch to parity verification confidence enhancement on their corrected keys.

V. Results and Discussions

For implementation, we used the Nexys4 DDR board. The two interlocutors are connected to the board by a USB link. As software, we used ISE Design Suite tool from Xininx.

The following diagram shows the evolution of the mutual information of Alice and Bob (I_{AB}) and that of Eve (I_{BE}).
We can say that the errors correction phase and the privacy amplification phase allow to extend I_{BE} to 0. That is why we said that the objective of the errors correction is to scramble Eve's information.

The second diagram shows the evolution of the QBER.

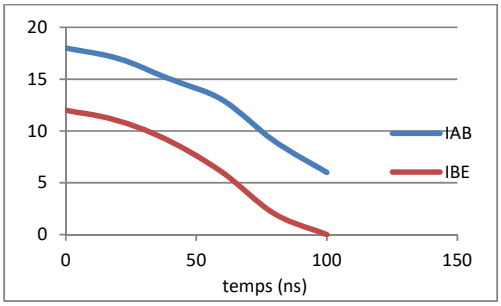

Fig. 2 Diagram of mutual information evolution

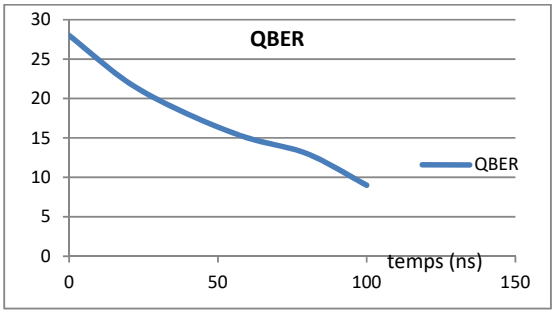

Fig. 3 Diagram of QBER evolution

The QBER is decreasing when moving from one phase to another. We fixed the threshold of the errors rate to 10%.

VI. Conclusion

In this paper, we proposed an algorithm of the BB84 protocol, the first QKD protocol. We implemented this algorithm on an FPGA platform where we used the Nexys4 board.

QKD protocols take advantage of the quantum mechanics principles to ensure unconditional security of the communication process, even in the presence of a spy.

Acknowledgment

This work is supported by Electronics and Microelectronics Laboratory, Sciences Faculty of Monastir-Tunisia (code: LR99ES30) and National Engineering School of Tunis-Tunisia, Communication System Department.

First author thanks Mr. Tayari Lassaad, master technologist aggregated in computer science of industrial systems at higher institute of technological studies Gabes-Tunisia, who provides Nexys4 board.

References

[1] Auguste Kerckhoffs. Modern Cryptography. Journal militaries sciences, IX: pages 5–38 et 161–191, January – February 1883. Available on http: // www.cl.cam.ac.uk/~fapp2/kerckhoffs/index.html.

[2] C. E Shannon " A Mathematical Theory of Communication " Bell System Technical journal, Vol.27 N°4 1999.pp 379-423,623-656

[3] Charles H. Bennett and Gilles Brassard. Quantum cryptography: public-key distribution and coin tossing. In Proceedings of the IEEE International conference on Computers, Systems and Signal Processing, pages 175–179. IEEE, 1984.

[4] M. Wegman. New hash functions and their use in authentication and set

equality. Journal of Computer and System Sciences, vol. 22, no. 3, pages 265–279, Juin 1981. (Cité en pages 45 et 57.).
[5] A. Muller, T. Herzog, B. Huttner, W. Tittel, H. Zbinden, N. Gisin, “Plug and play systems for quantum cryptography”, Appl. Phys. Lett., pp. 793, 17 Février 1997.
[6] W. K. Wootters, W. H. Zurek: Nature99, 802 (1982) A single quantum cannot be cloned.
[7] https://reference.digilentinc.com/reference/programmable-logic/nexys-4-ddr/reference-manual.
[8] Andreas Klein, Linear Feedback Shift Registers, 20 avril2013, p. 17-58.
[9] M. Koutsoupia, E. Kalligeros and X. Kavousianos, LFSR-based test-data compression with self-stoppable seeds, Design, Automation & Test in Europe Conference & Exhibition, 20-24 April 2009, p. 1482-1487.
[10] W. Liang et Jing Long, « A cryptographic algorithm based on Linear Feedback ShiftRegister », Computer Application and System Modeling (ICCASM), 2010 International Conference on, 22-24 Octobre 2010.
[11] J. A. Reeds and J. A. Sloane, Shift Register Synthesis (Modulo m), SIAM Journal on Computing, August 1985, p. 505-513.