Advanced Journal of Graduate Research
ISSN:2456-7108
Volume 4, Issue 1, pp. 59-67, July 2018
DOI: https://doi.org/10.21467/ajgr.4.1.59-67

GRADUATE RESEARCH ARTICLE

A New Quantum Encryption Scheme

Plesa Mihail-Iulian*, Togan Mihai

Department of Computer Science, Technical Military Academy of Bucharest,
Bulevardul George Cosbuc 81-83, Bucuresti 050141

*Corresponding Author email:
plesamihailiulian@gmail.com

Article History

Received: 4 June 2018
Revised: 15 June 2018
Accepted: 21 June 2018
Published: 22 June 2018

Student(s)

- Plesa Mihail-Iulian

Academic Year: 2017-18
Course Level: Bachelor
Course Name: B.S. (Computer Science)
Course year: 4th year

Mentor(s)

- Togan Mihai

ABSTRACT

The model of quantum computation has advanced very quickly in the last years. This model brings with it an efficient algorithm for factoring, namely the Shor algorithm. This means that the public key infrastructure will soon be obsolete. In this paper we propose a new quantum cryptographic scheme which aims to replace the RSA algorithm from current public key infrastructures. We analyze the security of our scheme and also, we describe the implementation of the scheme using IBM Q SDK, qiskit. We run a number of experiments in order to build a proof of concept application that uses the proposed scheme.

Keywords: factoring; IBM Q; PKI; qiskit; quantum cryptography; RSA; Shor

1 Introduction

The RSA algorithm is certainly one of the most widespread public key algorithms [1]. In this paper we will use two characters consecrated into cryptography, Alice and Bob. In the known context of the RSA algorithm, we note with (p_A, e_A) the pair of public and private key of Alice and with (p_B, e_B) the public and private key owned by Bob. In order to send an encrypted message to Bob, Alice must retrieve Bob's public key, p_B. Although simple to do at first glance, this procedure is an interesting issue. If Bob tells Alice his public key over an unsafe channel then we will encounter another problem, namely, the problem of impersonation. Suppose Alice uses an unsafe channel in order to ask for Bob's public key. This channel is controlled by Eve, an attacker. Suppose that Eve holds pair of public and private key.denoted with (p_E, e_E). When Alice asks Bob's public key on this channel, Eve could change Bob's key, p_B, with his public key, p_E. Alice will receive p_E key instead of Bob's key, p_B. Now she will use p_E in order to encrypt the message she wants to pass it on to Bob. Once the message has arrived to Eve, she can decrypt it with her own private key, e_E. Eve can modify the message and encrypt it with Bob's public key. In this way, Bob will think that the message has come from Alice. This is briefly the impersonation problem. Since this is a core attack in a public key infrastructure, many applications are in danger of being attacked [2]. We therefore need a

mechanism to ensure that Alice's public key is indeed her and that Bob's public key is indeed his. This mechanism is named PKI. Although the subject is being treated for several decades, protocols underpinning pki are still published [3]. PKI mechanisms appear everywhere where public key cryptography is used. What provides such an infrastructure is a network of certificates issued by a central authority (CA). Each member enrolled in the public key infrastructure receives from the central authority, following verifications, a certificate linking the participant's identity to its public key. Thus, the issue of impersonation is solved using the PKI model. The main purpose of a public key infrastructure is to provide a secure communication channel between two participants. Most often, the communication channel is a hybrid channel. In the first phase, through the public key encryption mechanisms provided by the infrastructure, a symmetric key is distributed between participants. From this moment, the participants will use a symmetric algorithm in order to communicate. The vast majority of public key infrastructures use the as an asymmetric algorithm, the RSA algorithm. This leads to the next problem of these infrastructures, a problem whose solution we have tried to elaborate in this paper. The RSA algorithm has as a cryptographic primitive the factorization problem [4]. In other words, given a RSA module, $N = p * q$, a classical polynomial algorithm for factoring is not known. Although a classical algorithm is not yet known. This is the famous N vs NP problem [5] for the factorization problem, this can not be said about the quantum algorithms. In 1994, Ptere Shor proposes an efficient quantum algorithm to solve the problem of factorization [6]. In the last past years, physical implementations was proposed for Shor algorithm [7]. Although this algorithm did not represent a good period of time a threat to the RSA algorithm and implicitly to public key infrastructures, this is no longer true at the moment. Quantum computers are no longer simple lab experiments. Such a quantum computer is provided through a cloud by IBM Q [8]. In December 2017, Microsoft announced a new quantum programming language integrated in Visual Studio, Q # [9]. At the time of writing this paper, Microsoft simulator has a number of 30 quits. IBM Q provides not only a 32 qubit simulator but also cloud access to a real device with up to 16 qubits [8]. All this advance towards a scalable quantum computer [9] raises concerns about public key infrastructures, specifically with regard to the RSA algorithm. In this paper, we propose a quantum cryptographic scheme that could replace the RSA algorithm in PKI infrastructures. The cryptographic scheme we propose aims to achieve the goal of encrypting a qubit. By definition, encryption of a qubit is a cryptographic operation that does not allow an attacker to access the probability amplitudes that define the superposition of the qubit. n other words, given that an attacker could statistically determine the probability amplitudes of the qubit superposition state, the encryption operation assures that the attacker can not reconstruct through quantum tomography the probability amplitudes [10][11]. Using an encrypted qubit we can transmit both a bit of classical information, in which case one of the two probability amplitudes would be 0 and more information bits by means of probability amplitudes. In other words, the information encrypted with this scheme can be encoded in the form of probability amplitudes. Thus, from a single run of the scheme we can transmit several bits of information. But we can use the scheme to send one bit of information each time we run. In this latter case, the complexity of the scheme may be an impediment. The scheme falls under the category of asymmetric cryptographic schemes as the encryption key differes the decryption key. The paper is structured as follows: In Section 2 we discuss an overview of quantum cryptography with a demonstration experiment for generating a random number represented on 4 bits. In Section 3 we discuss the proposed cryptographic scheme as well as an implementation using the qiskit api. In Section 4 we discuss the conclusions and future directions of research.

2 An Overview of Quantum Cryptography

The most known application of quantum effects in cryptography is the key distribution [12], [13]. The first and best known work that uses quantum effects to achieve a cryptographic objective is the work of Charles Bennett and Gilles Brassard in 1984 [14]. The paper proposes the use of quantum properties for key distribution between two parts, Alice and Bob. The protocol is called BB84 and uses two effects of quantum systems. The first effect is known as non-cloning theorem. What this theorem says is the fact that a quantum system cannot be copied without being previously measured [15]. This prompts the emergence of

cryptographic applications. A second basic effect used by the BB84 protocol is the collapse of probability wave. In other words, given a quantum system represented by a qubit in the superposition state (1), after measurement, the system will collapse into one of the two basic states, $|0\rangle$ or $|1\rangle$, each state having a probability of 50% being measured.

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \quad (1)$$

The protocol also has a statistical mechanism for detecting an attacker. The BB84 protocol is a theoretical model and an example of using quantum effects in setting cryptographic tasks but is not used in practice. Although the theoretical protocol has perfect security based on the effects of quantum physics instead of some primitive mathematics, a series of attacks against that have been found over time [16]. These attacks are also of a physical nature. An important lesson learned from the story of this protocol is that prudence is a key factor in cryptography. Although we can ensure perfect security by using quantum effects, some physical effects can endanger the security of the scheme. Another protocol for key distribution is the E91 protocol [17]. Developed in 1991 by Artur Ekert, the protocol uses the effects of the entanglement state to perform a cryptographic function. The effect of entanglement can be described most intuitively at the level of a system of qubits as described in (2).

$$|\psi\rangle = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle \quad (2)$$

Analyzing the state (2) we notice that for any qubit that we measure we have a 50% chance for $|0\rangle$ and 50% chance for $|1\rangle$. The interesting part is what happens to the other qubit, which is not measured. According to state (2), if the first qubit was measured and found in the state $|0\rangle$ for example, then we can safely say that the other qubit is also in this state. This effect has been greatly disputed in physics [18]. The effects of the entanglement state also lead to various applications in cryptography. A well-known circuit based on entanglement is the teleporting circuit which has also found applications in various cryptographic functions [19].

Another place where quantum physics finds its application in cryptography is in the field of random number generation. Random generation has extensive cryptography ramifications. The state described in (1) is a perfect generator of random numbers. Although we can only generate one bit using the qubit in state (1), a n qubit system in which each qubit is found in the state (1) will generate n random bits. The state (1) can easily be prepared by applying to a qubit originally found in the state $|0\rangle$ a Hadamard gate [20]. Here's an example of how to generate a random number that can be represented on 4 bits. Running the circuit in Figure 1 1024 times we could reconstruct the histogram of the results in Figure 2.

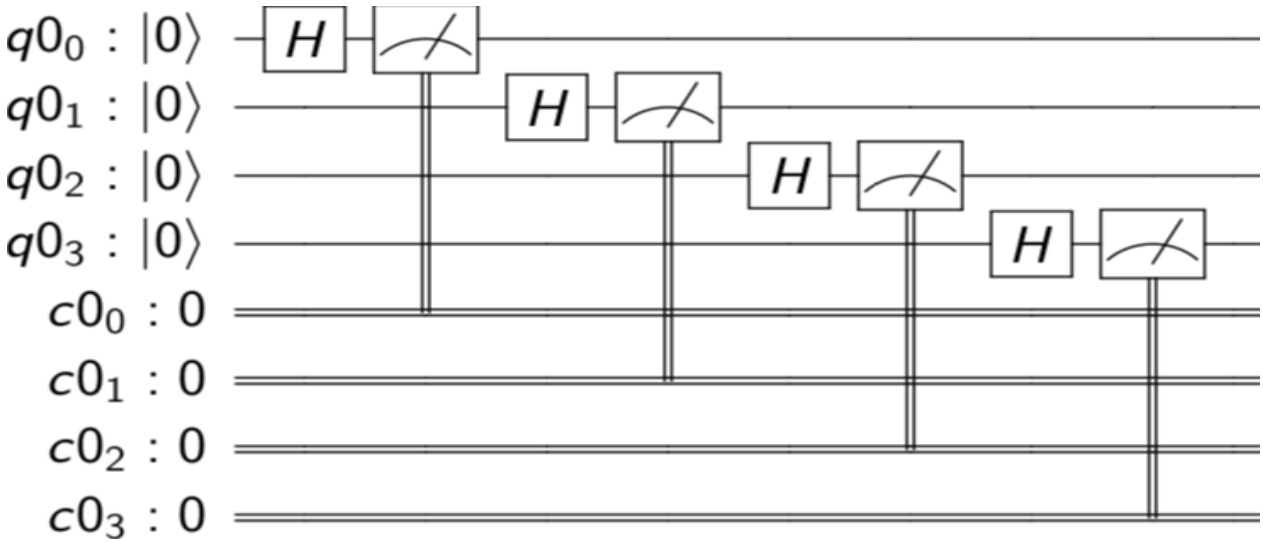

Figure 1: *Circuit used to generate a random number represented by 4 bits*

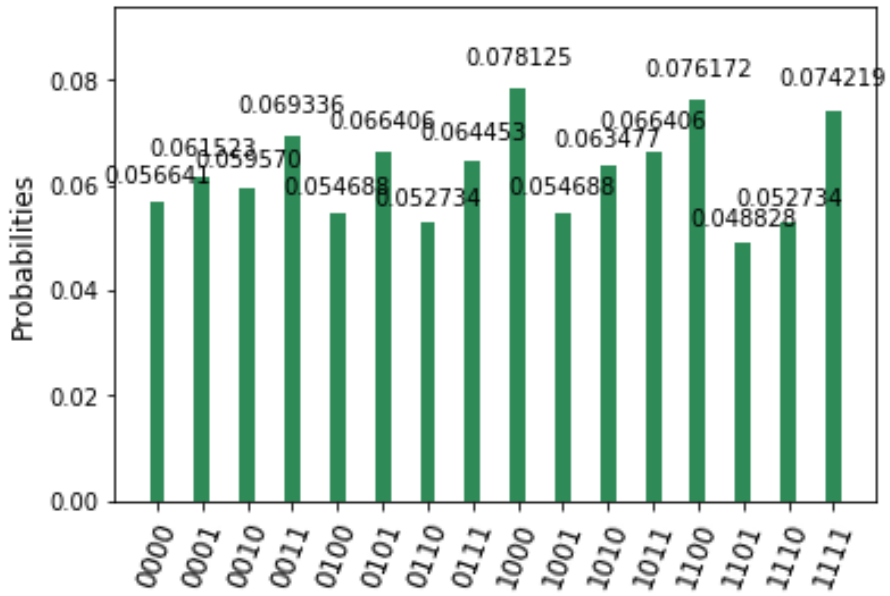

Figure 2: *Histogram of the results obtained following the execution of the circuit in Figure 1 1024 times.*

As we can see from the histogram, each number has a probability of about 6.25% to be measured.

3 The Proposed Scheme

We present in this section the proposed cryptographic scheme, and in the 3.2 section we present the security analysis in the case of a brute-force attack and in the 3.3 section we present the schematic deployment implementation using the qiskit api.

3.1 Construction of the Cryptographic Scheme

The scheme that we are proposing aims at delivering a secure transmission of a qubit from Alice to Bob. In other words, we do not want a possible Eve attacker to take possession of the qubit. Consider the qubit in state (3) as the qubit we want to transmit.

$$|\psi\rangle = a|0\rangle + b|1\rangle \tag{3}$$

Where $a, b \in \mathbb{C}$ and $|a|^2 + |b|^2 = 1$.

By encrypting the qubit in state (3) we understand an operation whereby an attacker cannot determine the probability amplitudes a and b irrespective of the number of measurements performed. In the proposed scheme we use a set of $n + 1$ unitary transformations, $M = \{U_0, U_1, \dots U_n\}$. Each of the transformations in the M set respects the general form of a quantum gate for a system consisting of a single qubit, namely the form (4).

$$U_i = \begin{pmatrix} a & b \\ -e^{i\varphi}b^* & e^{i\varphi}a^* \end{pmatrix}, 0 \le i \le n \tag{4}$$

Where $a, b \in \mathbb{C}$ and $\varphi \in \mathbb{R}$, $|a|^2 + |b|^2 = 1$.

Suppose for the security analysis of the scheme, that the numbers a, b, φ can be represented using t bits. The scheme we are proposing consists of two stages. In the first phase we propose a key generation mechanism and in the second phase we show the qubit encryption scheme. Key generation takes place as follows:

1. Alice generates random numbers a, b and φ represented on t bits, then builds the matrix $U_0 = \begin{pmatrix} a & b \\ -e^{i\varphi}b^* & e^{i\varphi}a^* \end{pmatrix}$.
2. Alice generates n random numbers each represented on t bits. We note these numbers with $p_1, p_2 \cdots, p_n$.

3. Alice calculates the transformations $U_i = U_0{}^{p_i}, 1 \leq i \leq n$.
4. Alice's public key, denoted with p_A, will be the set of transformations $\{U_1, U_2, \dots, U_n\}$ and her private key,denoted with e_A, will be the transformation U_0.

We have chosen to define the set M of transformations as exponents of the transformation U_0 to ensure that any two transformations in the set M commutes between them. Once the key generation procedure is defined, we can also define the qubit encryption procedure. Suppose Bob wants to encrypt a qubit in the state (3) for Alice. Encryption is done as follows:

1. Bob generates randomly a set of r natural numbers ranging from 1 to n. We note this set with $R = \{r_1, r_2, \dots r_r\}, 1 \leq r_i \leq n$.
2. We denote with U_R the composition of the transformations $U_{r_1}, U_{r_2}, \dots U_{r_r}$. In other words, $U_R = \prod_{i=1}^{r} U_{r_i}$. Bob applies the transformation $U_R$to the a qubit in state (3). The qubit new state is now $|\psi^1\rangle = U_R|\psi\rangle$. Bob sends Alice through an unsafe channel the qubit in the state $|\psi^1\rangle$.

For decryption, Alice will proceed as follows:

1. We note with U_T the composition of the transformations from the set M. In other words, $U_T = \prod_{i=0}^{n} U_i$. Alice applies to a qubit in the state $|\psi^1\rangle$ the transformation U_T. The qubit state is now $|\psi^2\rangle = U_T|\psi^1\rangle$. Alice sends Bob the qubit in the state $|\psi^2\rangle$.
2. Bob calculates on the basis of set R, the inverse to U_R. We denote with $U^\dagger$ the conjugate transpose of U. Thus $U_R{}^\dagger = \prod_{i=1}^{r} U_{r_i}{}^\dagger$. Bob applies to the qubit in the state $|\psi^2\rangle$ the transformation $U_R{}^\dagger$, hence the new state of the qubit is $|\psi^3\rangle = U_R{}^\dagger|\psi^2\rangle = U_R{}^\dagger U_T|\psi^1\rangle = U_R{}^\dagger U_T U_R|\psi\rangle = U_R{}^\dagger U_R U_T|\psi\rangle = IU_T|\psi\rangle = U_T|\psi\rangle$. Bob sends Alice the qubit in the state $|\psi^3\rangle$.
3. Alice applies to the qubit that she received from Bob, in the state $|\psi^3\rangle$ the transorm $U_T^\dagger$, hence the new state of the qubit is $U_T^\dagger|\psi^3\rangle = U_T^\dagger U_T|\psi\rangle = I|\psi\rangle = |\psi\rangle$. So Alice recovers the original qubit from Bob.

3.2 Security Analysis

In this section we analyze the complexity of a brute force attack. By definition, we consider that the protocol has been broken when an attacker manages to statistically obtain a qubit in state (3). For reference, we compare the complexity of the attack on the proposed scheme with the complexity of a brute force attack on AES 256, considered to be a safe algorithm [21]. That is, we will compare the complexity of the attack on our scheme with 2^{256}. The probability that an attacker guesses an AES 256 key is $\frac{u}{2^{256}}$, where u is the number of attempts he makes. For simplicity we will considered $u = 1$.

We begin the analysis by discussing the probability that an attacker can guess exactly the set of R transformations used by Bob. Since every two transformations in the M set commutes, the probability that an attacker guesses from a single test, the set R is $\frac{1}{C_n^r}$. For comparison we choose $n = 512$ and $r = fn$. Where $0 < f < 1$. In Figure 3, the fraction f is passed along the horizontal axis and the difference $(C_n^r - 2^{256})$ is passed on the vertical axis.

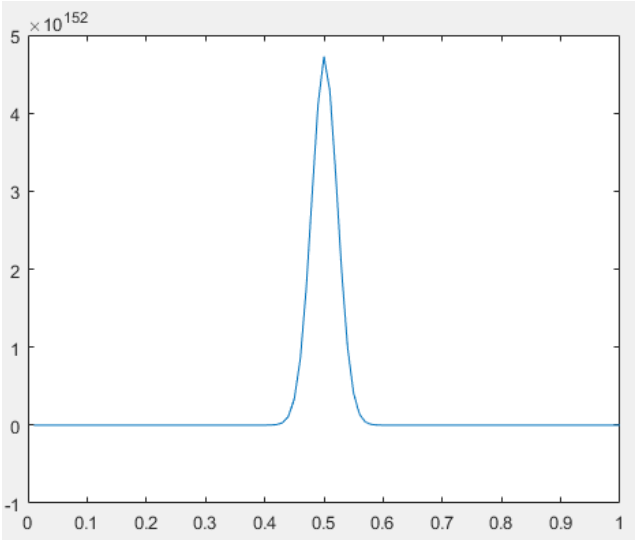

Figure 3: *The difference* $(C_n^r - 2^{256})$

It is noted that for $f \cong 0.5\ (C_n^r - 2^{256}) \gg 0$, which means that the probability that an attacker guesses the R set is much smaller than the probability that an attacker to guess an AES 256 key.

A second aspect we examine is the likelihood that an attacker will impersonate Alice. In other words, we examine the probability of an attacker guessing the transformation U_T. Given the general form of a unitary transformation (4), the transformation U_T is uniquely determined by the three numbers, a, b and φ. As specified in the description of the scheme, each of the three numbers can be represented on t bits. Therefore the probability that an attacker guesses the tranformation U_T is $\frac{1}{2^{3t}}$. Therefore for $t > 86, 3t > 256$ which implies $\frac{1}{2^{3t}} < \frac{1}{2^{256}}$.

Concluding, we can say that using a number of $n \geq 512$ unit transformations of which Bob will choose about 50% to make the R set and using numbers represented by more than 86 bits to define unitary transformations, the proposed scheme is safe for with regard to brute force attacks.

3.3 An Implementation of the Scheme in Qiskit

First, the proposed scheme involves generating random numbers. The implementation of a circuit capable of generating a random number has already been described in Section 2. Another equivalent form of expression of a unitary transformation is (5).

$$U = \begin{pmatrix} cos\left(\frac{\theta}{2}\right) & -e^{i\lambda} sin\left(\frac{\theta}{2}\right) \\ e^{i\varphi} sin\left(\frac{\theta}{2}\right) & e^{i\varphi + i\lambda} sin\left(\frac{\theta}{2}\right) \end{pmatrix} \quad (5)$$

Any unitary transformation can be implemented in qiskit with the u3 (theta, phi, lam, q) function that conforms to the notations given in form (5). Alice's public key will consist of the actual values that determine each transformation in part, but to implement the transformations in qiskit, they will be determined by those three values: θ, φ and λ. In other words, Bob does not have access to these values in the actual version of the scheme, but for these simulations, these values will be used. For simulation, we consider $\theta = \frac{\pi}{3}$, $\varphi = \lambda = \frac{\pi}{\sqrt{2}}$, $p_1 = 3, p_2 = 5, p_3 = 7, p_4 = 11$, $R = \{2,4\}$. To implement the transformations U_0, we implemented a function that applies the transformation a number of times equal to the power at which the transformation U_0 is intended to be exponential. Figure 4 describes the exponential function.

```
01.  def U(theta,phi,lam,power):
02.      for i in range(power):
03.          qc.u3(theta,phi,lam,qr[0])
04.      return;
```

Figure 4: *The exponential of U_0*

The simulation of the scheme is structured in 4 steps. The first step is made by Bob encrypting the qubit $|\psi\rangle$ by applying the $U_2$and U_4 transformations. This step is illustrated in Figure 5.

```
01.  def bobFirstPart():
02.      for i in range(len(r)):
03.          U(theta,phi,lam,p[r[i]])
04.      return;
```

Figure 5: *The application of the gates U_2 and U_4*

The second step is made by Alice and consists in the application of the gates $U_i, 0 \leq i \leq 4$, as illustrated in Figure 6.

```
01.  def aliceFirstPart():
02.      U(theta,phi,lam,1)
03.      for i in range(len(p)):
04.          U(theta,phi,lam,p[i])
05.      return;
```

Figure 6: *The application the entire set of gates to the qubit by Alice*

The third and fourth steps are Bob's application of inverse transformations to those applied at the first step and Alice's application of inverse transformations to those applied in step 2.

```
01.  def bobSecondPart():
02.      for i in range(len(r)):
03.          U(-theta,-phi,-lam,p[r[i]])
04.      return;
05.
06.  def aliceSecondPart():
07.      U(-theta,-phi,-lam,1)
08.      for i in range(len(p)):
09.          U(-theta,-phi,-lam,p[i])
10.      qc.measure(qr[0],cr[0])
11.      return;
```

Figure 7: *Applying inverse transformations by Bob, respectively Alice*

The qubit we will encrypt is the one described in (6).

$$|\psi\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle \tag{6}$$

To obtain condition (6), apply to a qubit initially in the state $|0\rangle$ the Hadamard gate [19]. To run the experiment, the functions described in Figures 5, 6 and 7 are run sequentially. Theoretically, Alice should recover the qubit (6). This is checked by quantum tomography [10]. In other words, we run the experiment circuit 1024 times and statistically reconstruct the probability amplitudes. The histogram of the obtained results is shown in Figure 8.

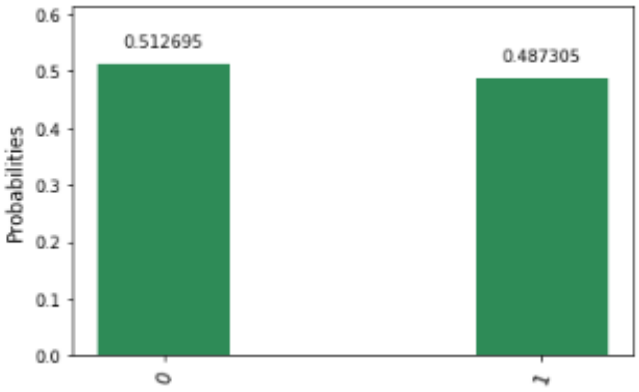

Figure 8: Histogram of the results obtained by Alice

At the same time, another point to the experiment is to convince us that an attacker can not rebuild these probability amplitudes. For this we measure statistically the state of the qubit by quantum tomography whenever it is transmitted between Alice and Bob [10]. In particular, we rebuild the state immediately after Bob's first step, after Alice's first step and Bob's second step. The results are reported in the Figures 5, 6 and 7.

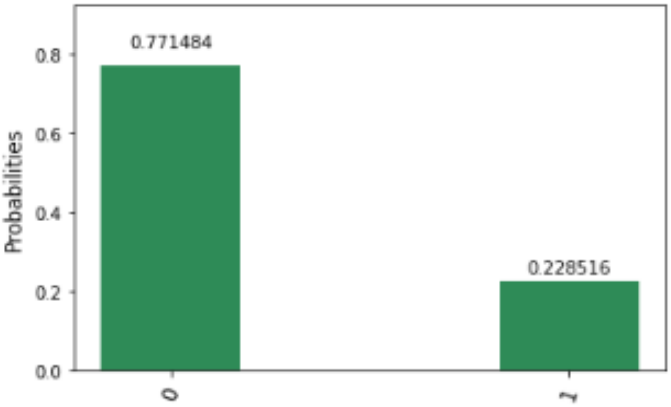

Figure 9: *The histogram of the results immediately after Bob's first step*

The histogram of the results of the qubit after the first step of Bob are described in Figure 9, the results of the measurement after the first step of Alice are described in Figure 10 and the results after the second step of Bob are described in Figure 11.

As can be seen from Figures 9, 10 and 11, the statistical reconstruction of the qubit state in one of the intermediate steps leads to differing probability amplitudes from the real probabilities. We can therefore conclude that the scheme achieves its experimental objective of encrypting the qubit.

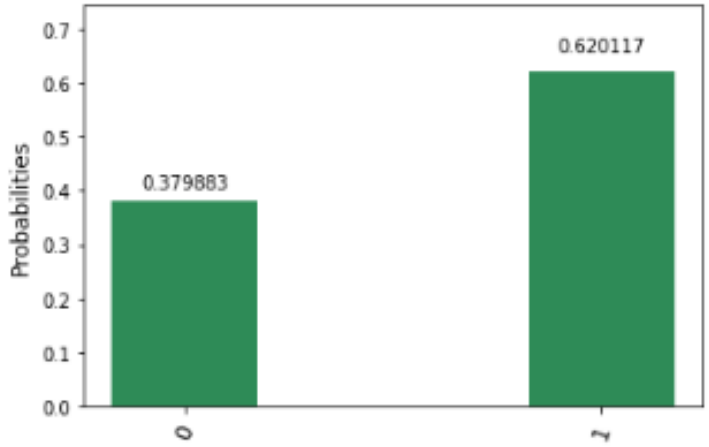

Figure 10: *Histogram of results immediately after Alice's first step*

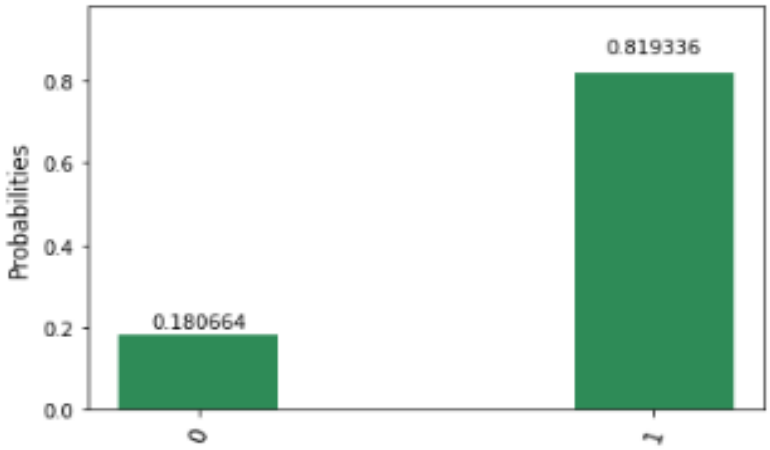

Figure 11: *Histogram of the results after Bob's second step*

4 Conclusions

In this paper we proposed a public key cryptographic scheme to replace the RSA algorithm in PKI infrastructures. The rapidity with which quantum machines evolve will jeopardize current PKI infrastructures that still use RSA. The proposed scheme provides encryption of a qubit. Thus, we can use the scheme either for passing a classical bit of information in turn, or we can encode the information we want to encrypt in the form of probability amplitudes of the qubit superposition state. We studied the complexity of a brute force attack and concluded that this is equivalent in the computational power required with that of an attack on AES-256, an algorithm considered safe even in the context of quantum computing. We also presented a demonstration tutorial on how to implement the proposed scheme. We have seen experimentally that Alice recovers through quantum tomography the probability amplitudes transmitted by Bob. We also experimentally saw that any attempt to determine the probability amplitudes in one of the intermediate phases when the qubit is transmitted between Alice and Bob leads to the determination of amplitudes that do not match the real ones. Like future research directions, we could add a way to determine the attacker's presence. Although we have seen that the attacker cannot reconstruct the correct probability amplitudes, we did not say anything about a way of deciding it. Another direction of research would be switching from classical to quantum algorithms in other areas of PKI. In this paper we only addressed the issue of confidentiality. Future papers can address the issues of authenticity, non-repudiation, or integrity of messages in a PKI infrastructure.

5 Acknowledgment

I will like to thank professor Mihai Togan for useful discussions on theoretical and practical cryptography and also, I would like to thank IBM Q for making possible for anyone to experiment with a real quantum device.

How to Cite this Article:

M.-I. Plesa and T. Mihai, "A New Quantum Encryption Scheme", *Adv. J. Grad. Res.*, vol. 4, no. 1, pp. 59-67, Jun. 2018. doi: 10.21467/ajgr.4.1.59-67

References

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[2] M. Ramadan, G. Du, F. Li, and C. Xu, "A survey of public key infrastructure-based security for mobile communication systems," *Symmetry*, vol. 8, no. 9. 2016.

[3] B. Crowe, "Proposed AeroMACS PKI specification is a model for global and National Aeronautical PKI Deployments," in *ICNS 2016: Securing an Integrated CNS System to Meet Future Challenges*, 2016.

[4] C. L. Duta, L. Gheorghe, and N. Tapus, "Framework for evaluation and comparison of integer factorization algorithms," in *Proceedings of 2016 SAI Computing Conference, SAI 2016*, 2016, pp. 1047–1053.

[5] Gowers Timothy, Ed., "The P versus NP Problem," in *The Princeton Companion to Mathematics*, Princeton University Press, 2008, p. 713.

[6] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, Jan. 1999.

[7] N. Johansson and J.-Å. Larsson, "Realization of Shor's Algorithm at Room Temperature," Jun. 2017.

[8] "Quantum Computing - IBM Q - US," *The future is quantum*. [Online]. Available: https://www.research.ibm.com/ibm-q/. [Accessed: 22-Jun-2018].

[9] "Quantum computing | Microsoft," *Empowering the quantum revolution*. [Online]. Available: https://www.microsoft.com/en-us/quantum. [Accessed: 22-Jun-2018].

[10] Z.-Y. Han *et al.*, "Efficient Quantum Tomography with Fidelity Estimation," Dec. 2017.

[11] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition," *Interspeech*, no. Cd, pp. 338–342, 2014.

[12] E. Diamanti, H.-K. Lo, B. Qi, and Z. Yuan, "Practical challenges in quantum key distribution," *npj Quantum Inf.*, vol. 2, no. 1, p. 16025, Nov. 2016.

[13] N. Ltkenhaus, "Theory of Quantum Key Distribution (QKD)," in *Lectures on Quantum Information*, Weinheim, Germany: Wiley-VCH Verlag GmbH, pp. 271–284.

[14] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theor. Comput. Sci.*, vol. 560, pp. 7–11, Dec. 2014.

[15] J. Ortigoso, "Twelve years before the quantum no-cloning theorem," Jul. 2017.

[16] L. A. Lizama-Pï-rez, J. M. Lï-pez, and E. D. C. Lï-pez, "Quantum key distribution in the presence of the intercept-resend with faked states attack," *Entropy*, vol. 19, no. 1, 2017.

[17] D. S. Naik, C. G. Peterson, A. G. White, A. J. Berglund, and P. G. Kwiat, "Entangled state quantum cryptography: Eavesdropping on the ekert protocol," *Phys. Rev. Lett.*, vol. 84, no. 20, pp. 4733–4736, 2000.

[18] N. Rosen, "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?," in *Albert Einstein*, Wiesbaden: Vieweg+Teubner Verlag, 1979, pp. 57–67.

[19] M.-I. Plesa, "Hybrid scheme for secure communications using quantum and classical mechanisms," in *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2017, pp. 1–6.

[20] A. Barenco *et al.*, "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, no. 5, pp. 3457–3467, Nov. 1995.

[21] R. Reynard, *Secret code breaker III : a cryptanalyst's handbook*. Smith & Daniel Marketing, 1999.