ROMANIAN JOURNAL OF MATHEMATICS AND COMPUTER SCIENCE, 2019, VOLUME 9, ISSUE 2, p.78-86

SET PARTITIONS: SOLUTION FOR SHARING SECRET KEYS

SADEK BOUROUBI, FELLA CHARCHALI AND NESRINE BENYAHIA TANI

Abstract. Confidentiality was and will always remain a critical need in secret exchanges either between simple persons or official parties. In this paper, we present a new cryptographic protocol for sharing a key over insecure channels, even in the face of powerful adversaries. This protocol, called *BCB* (*B*ouroubi *C*harchali *B*enyahia Tani), is developed in order to provide a secret key, which can be as long as the message to be encrypted and used for instance in the One-Time Pad using a set partitions. To evaluate the performance of secret encryption keys provided by *BCB*, we did a series of tests on the images. These tests included histogram analysis, randomness test, information entropy, correlation analysis and encipherment quality. Experimental results show that *BCB* satisfies security criterion and ensure confidentiality of the secret exchanges.

Mathematics Subject Classification (2010): 11P82, 94A60.
Key words: Cryptography; Set Partitions; *BCB* protocol; One-Time Pad Cipher.

Article history:
Received 19 September 2018
Received in revised form 25 September 2018
Accepted 03 April 2019

1. Introduction

Confidentiality of information has been solved by cryptography. The certificate that the One-Time Pad is unconditionally secure, has transformed the problem from ensuring the confidentiality of information to the distribution problem of the secret key. Until the eighties, one way to distribute the secret key, apart from hand to hand, was to use algorithms whose security is based on the computational complexity. The generated keys by such algorithms are reasonably secret but not unconditionally secret. Let us define Alice and Bob two participants involved in the encryption process and Eve, the intruder who wants to spy on them. Our approach is based on the complexity of finding a set partition over an exponential number of partitions when the set is large enough. The expected objective from the protocol is to produce secret keys, that will be used to ensure confidential communications between the participants by interchanging enciphered messages through a classical channel.

First, Alice and Bob must share $\pi = \{A_1, A_2, \ldots, A_k\}$, a partition of a set $[n] = \{1, 2, \ldots, n\}$ into k-disjoint blocks (n is assumed to be large enough). Then, they will use the k blocs of the partition π as holders of components constructing the secret key that they will both construct separately by doing manipulations on each bloc while keeping the results secret.

This paper is organized as follows: In Section 2, we briefly recall the concept of unconditional security illustrated by the One-Time Pad. In Section 3, we introduce the definition of set partition, Stirling and Bell numbers on which the protocol is based. The new secret key conception protocol is proposed in Section 4. In Section 5, an illustrative example is given to show how the protocol runs. In Section 6, we discuss the security of the protocol and the amount of information that can an intruder obtain by spying on the secret exchange between legal parties. Section 7 deals with image encryption and the evaluation of the performance of the encryption key provided by *BCB* through various tests, such as histogram

analysis, information entropy, correlation analysis and encipherment quality. Finally, a conclusion is given in Section 8.

2. Unconditional Security Concept

The One-Time Pad, also known as disposable mask or Vernam cipher, provides perfect security, despite its simplicity. In its classic form, it is nothing but a very long random sequence of letters, written on pages bound together to form a block. The sender uses each letter of the mask in turn to encipher exactly one plaintext character. The One-Time Pad text C is a function of both the message M and the key K. The resulting string is also random and reveals no information about the message. The One-Time Pad fits very well to the definition of a perfect system ensured via the concept of entropy introduced in cryptography by Shannon in 1949 [1]. For more details see for instance [2], [3], [4] and [7].

Using binary logic, the encipherment algorithm E can be written as:

$$E_K(M) = (m_1 \oplus k_1, m_2 \oplus k_2, \ldots, m_n \oplus k_n),$$

where $M = (m_1, m_2, \ldots, m_n)$ is the plaintext, and $K = (k_1, k_2, \ldots, k_n)$ the binary random key. The ciphertext is the output of the exclusive-or operation (or addition modulo 2), denoted $\oplus$, between the message and the key.

Decipherment process D is the same as encipherment, it is given by:

$$M = D_K(C) = (c_1 \oplus k_1, c_2 \oplus k_2, \ldots, c_n \oplus k_n).$$

Unfortunately, just like any other cryptographic system, it has significant drawbacks which can cause its vulnerability such as when the key is not as long as the plaintext, or when the same key is used more than once. The most important issue which makes One-Time Pad no practice is the sharing of the secret key. We should note that the safest way to transport the key is the diplomatic bag which requires users from the diplomatic sector only.

To remedy major drawbacks of this cipher, *BCB* can be a solution, as soon as it eliminates the problem of exchanging the secret key, it allows to generate separately random secret encryption key by the concerned parties whatever they are and that for each new exchange, the key can be as long as the plaintext.

3. Set Partitions, Stirling and Bell numbers

Let E be a set. A partition of E is an unordered collection of pairwise disjoint, nonempty subsets of E whose union is all of E. For $E = \pi_1 \cup \pi_2 \cup \cdots \cup \pi_k$ to be a partition of E, two things are required:

$$\pi_i \neq \emptyset,\ 1 \leq i \leq k,$$

$$\pi_i \cap \pi_j = \emptyset \text{ whernever } i \neq j.$$

Let define E as $[n] = \{1, 2, \ldots, n\}$ and let P_n be the set of all partitions of E. The number partitions of $[n]$ into k blocks is denoted $S(n, k)$ and called a Stirling number of the second kind. If $n \geq k \geq 2$, then we have:

$$S(n+1, k) = S(n, k-1) + kS(n, k).$$

Evidently, $S(n, k) = 0$ if $k < 1$ or $k > n$. The total number of partitions of $[n]$ is denoted B_n and called the n^{th} Bell number. They are defined by:

$$B_0 = 1 \text{ and } B_n = \sum_{k=1}^{n} S(n, k).$$

Furthermore, Moser and Wyman [6] established, when $n \to \infty$,

$$B_n \backsim \frac{1}{\sqrt{n}} r^{n+\frac{1}{2}} e^{r-n-\frac{1}{2}}, \quad re^r = n.$$

So, B_n gets an exponential asymptotic behavior. For example $B_{26} = 49631246523618756274$.

ROMANIAN JOURNAL OF MATHEMATICS AND COMPUTER SCIENCE, 2019, VOLUME 9, ISSUE 2, p.78-86

4. *BCB* protocol

The two parties dealing with the protocol should share a partition $\pi = \{A_1, A_2, \ldots, A_k\}$ of $[n] = \{1, 2, \ldots, n\}$, which must be kept in secret.

Consider M, the message that Alice wants to share with Bob. So, the steps to follow are:

(1) Alice calculates L_m, the binary length of M to be enciphered.

(2) Let $p > k$ and s be a positive integers fixed by Alice who generates a sequence of positive 2-tuples $(v_n, w_n)_n$ of length $s + L_m$, using the formula:

$$\begin{pmatrix} v_n \\ w_n \end{pmatrix} = f\left(v_{n-1}, w_{n-1}\right) \pmod{p+1}, \ 1 \leq n \leq s + L_m,$$

with f a function fixed by Alice, among a list of chaotic generators and (v_0, w_0), the starting 2-tuple.

Alice sends $(v_n, w_n)_{1 \leq n \leq s+L_m}$ to Bob.

(3) The two parties calculate the sequence $(x_n, y_n)_n$ from $(v_n, w_n)_n$ as follows:

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} v_n \\ w_n \end{pmatrix} \pmod{k} + \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \ 1 \leq n \leq s + L_m.$$

(4) Alice and Bob eliminate the redundant 2-tuples (x_m, y_m) from $(x_n, y_n)_{1 \leq n \leq s+L_m}$, to get a common list without repetition $\left(x_{\varphi(n)}, y_{\varphi(n)}\right)_{1 \leq n \leq l}$, where $l \leq s + L_m$.

(5) The two parties build the secret key using the obtained common list $\left(x_{\varphi(n)}, y_{\varphi(n)}\right)_{1 \leq n \leq l}$ by unrolling the following algorithm.

Algorithm 1 *Key* generator

Require: Sequence: $(x_{\varphi(n)}, y_{\varphi(n)})_{1 \leq n \leq l}$

Ensure: *Key*,

for n **from** 1 **to** l **do**

Calculate $g\left(x_{\varphi(n)}, y_{\varphi(n)}\right)$ and concatenate it to the *Key*

end for

The function g in Algorithm 1 plays an important roll for the key construction. Indeed, bits provided by g must be non predictable. Practically, g is defined as follows:

$$g\left(x_{\varphi(n)}, y_{\varphi(n)}\right) = \left[\sum_{i=1}^{|A_{x_{\varphi(n)}}|} a_i\right]_2 \oplus \left[\prod_{j=1}^{|A_{y_{\varphi(n)}}|} b_j\right]_2 ; \ a_i \in A_{x_{\varphi(n)}} \text{ and } b_j \in A_{y_{\varphi(n)}},$$

where $[a]_2$ denotes the binary representation of an integer a.

5. Illustrative example

Here is a hopefully illustrative example to show how *BCB* runs. The first step, consists to generate a pseudo-random partition of $[n]$ into k-blocks. The second consists of unrolling *BCB*, then injecting the provided key in One-Time Pad Cipher, the adopted cryptosystem, to get out finally with the enciphered text.

We consider first the following shared parameters between Alice and Bob:

$$n = 20, \ k = 6 \text{ and } \pi = \{A_1, A_2, A_3, A_4, A_4, A_5, A_6\},$$

ROMANIAN JOURNAL OF MATHEMATICS AND COMPUTER SCIENCE, 2019, VOLUME 9, ISSUE 2, p.78-86

where, $A_1 = \{5, 8, 9, 14, 17\}$, $A_2 = \{1, 7\}$, $A_3 = \{12, 19, 20\}$, $A_4 = \{2, 10, 11\}$, $A_5 = \{3, 4, 6, 15, 18\}$ and $A_6 = \{13, 16\}$.

Let **"It rains"** be the secret message written in binary as follows:

0100100101111010000100000011100100110000101110100101101110011100 11

Alice sets the parameters s at 5 and p at 100 and uses the two-dimensional Linear Congruential Generator (LCG):

$$\begin{pmatrix} v_n \\ w_n \end{pmatrix} = \begin{bmatrix} 2 & 3 \\ 4 & 7 \end{bmatrix} \begin{pmatrix} v_{n-1} \\ w_{n-1} \end{pmatrix} \pmod{101}, \text{ with } \begin{pmatrix} v_0 \\ w_0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

So she gets her sequence $(v_n, w_n)_n$ of length 69:

$\{(5,11),(13,51),(29,9),(61,43),(24,78),(51,16),(4,71),(11,89),(25,60),(53,45),(8,86),(19,48),(41,98),$
$(85,96),(72,88),(46,56),(95,29),(92,22),(86,95),(74,84),(50,40),(2,66),(7,69),(17,81),(37,28),$
$(77,18),(56,79),(14,20),(31,87),(65,52),(32,13),(67,59),(36,41),(75,70),(52,85),(6,44),(15,82),$
$(33,32),(69,34),(40,42),(83,74),(68,0),(38,7),(79,35),(60,46),(22,90),(47,64),(97,61),(96,49),(94,1),$
$(90,11),(82,51),(66,9),(34,43),(71,78),(44,16),(91,71),(84,89),(70,60),(42,45),(87,86),(76,48),$
$(54,98),(10,96),(23,88),(49,56),(0,29),(3,22),(9,95)\}$.

Alice sends $(v_n, w_n)_n$ to Bob.

Alice and Bob calculate, independently, the sequence $(x_n, y_n)_n$ from $(v_n, w_n)_n$ as follows:

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} v_n \\ w_n \end{pmatrix} \pmod 6 + \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \ 1 \leq i \leq 69,$$

and get

$\{(6,6),(2,4),(6,4),(2,2),(1,1),(4,5),(5,6),\underline{(6,6)},(2,1),\underline{(6,4)},(3,3),(6,3),\underline{(2,1)},(1,5),(5,3),\underline{(6,6)},(3,5),$
$(3,6),(3,1),\underline{(3,5)},\underline{(3,1)},\underline{(2,4)},\underline{(6,4)},(2,5),\underline{(6,1)},(3,2),\underline{(3,3)},\underline{(2,4)},(6,5),\underline{(3,2)},(2,6),(1,6),\underline{(4,5)},(5,2),$
$(1,3),\underline{(4,5)},\underline{(4,3)},\underline{(4,5)},(5,1),\underline{(6,3)},\underline{(3,1)},\underline{(2,6)},\underline{(1,5)},\underline{(5,1)},\underline{(6,5)},\underline{(2,2)},\underline{(1,2)},\underline{(5,2)},\underline{(1,6)},(5,4),\underline{(5,2)},$
$\underline{(6,1)},\underline{(3,5)},\underline{(2,6)},\underline{(5,1)},(1,4),\underline{(4,3)},\underline{(1,3)},\underline{(5,1)},\underline{(6,5)},(2,3),\underline{(1,6)},\underline{(4,5)},(4,6)\}$.

Alice and Bob eliminate the redundant 2-tuples from $(x_n, y_n)_{1 \leq n \leq 69}$, underlined above, to get the following common list without repetition $\left(x_{\varphi(n)}, y_{\varphi(n)}\right)_{1 \leq n \leq 30}$, which is:

$\{(6,6),(2,4),(6,4),(2,2),(1,1),(4,5),(5,6),(2,1),(3,3),(6,3),(1,5),(5,3),(3,5),(3,6),(3,1),(2,5),(6,1),$
$(3,2),(6,5),(2,6),(1,6),(5,2),(1,3),(4,3),(5,1),(1,2),(5,4),(1,4),(2,3),(4,6)\}$.

The two parties build the secret key using the obtained common list by unrolling the function g as follows:

$$g(6,6) = \left[\sum_{i=1}^{|A_6|} a_i\right]_2 \oplus \left[\prod_{j=1}^{|A_6|} b_j\right]_2 \ ; \ a_i \in A_6, \ b_j \in A_6,$$

$$g(2,4) = \left[\sum_{i=1}^{|A_2|} a_i\right]_2 \oplus \left[\prod_{j=1}^{|A_4|} b_j\right]_2 \ ; \ a_i \in A_2, \ b_j \in A_4,$$

and so one, until the last 2-tuple

$$g(4,6) = \left[\sum_{i=1}^{|A_4|} a_i\right]_2 \oplus \left[\prod_{j=1}^{|A_6|} b_j\right]_2 \ ; \ a_i \in A_4, \ b_j \in A_6.$$

In other word

$$g(6,6) = [29]_2 \oplus [208]_2 = 11101 \oplus 11010000 = 11001101,$$
$$g(2,4) = [8]_2 \oplus [220]_2 = 1000 \oplus 11011100 = 11010100,$$

and

$$g(4,6) = [23]_2 \oplus [208]_2 = 10111 \oplus 11010000 = 11000111.$$

After concatenation, the builded key in hexadecimal is then:

$$cdd4c1f14e854be7fe14eb811e311cd4bc511fe4bc3e314e834bf814ead344bedd8e5$$
$$2911e511c711c71b32f2e911d8c7$$

Bob, having the key, obtains the plain text by performing the exclusive-or operation between the enciphered message and the key.

6. Security of *BCB*

The proposed protocol carries out two objectives:

(1) The production of a secret key at least as long as the message to be enciphered with assurance of the synchronization between the transmitter and the receiver.
(2) The inability of a third person, to determine the generated secret key in a reasonable time.

If Eve intercepts all data exchanged between Alice and Bob, she gets no information neither on the partition π nor on the shared secret key. If Eve tries to find out the used encryption key, she has to determine n, the length of the partitioned set, the specific used partition π and its block's number k. All this lead to an exhaustive parameters search which is not feasible in a reasonable time. For example, assuming that Eve could spy on $n = 50$ (which is a small value in practice) she has to seek for the right partition used for the construction of the key among all the partitions of $[50]$ that are B_{50} partitions, where $B_{50} \geq 10^{47}$! Because we cannot be trusted to judge by ourselves whether the key is random or not. Some unbiased mechanical tests must be applied. The theory of statistics provides us with some quantitative measures for randomness. But, there is literally no end to the number of tests that can be conceived. For that, we analyzed all generated keys using image statistical tests, which involve determining whether or not a specific binary sequence is random. The following results showed that the key generated by *BCB* has a random behavior.

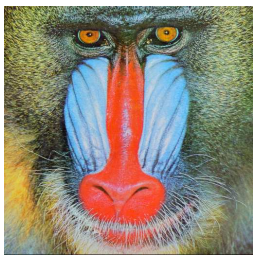
7. Image encryption based on One-Time Pad using *BCB* provided key

In this section, some experiments have been done to evaluate the performance of the proposed protocol, by enciphering images due to its some intrinsic features such as bulk data capacity and high redundancy, which are generally difficult to handle by traditional methods. The security analysis has demonstrated the satisfactory security of the protocol, as proved in the following.

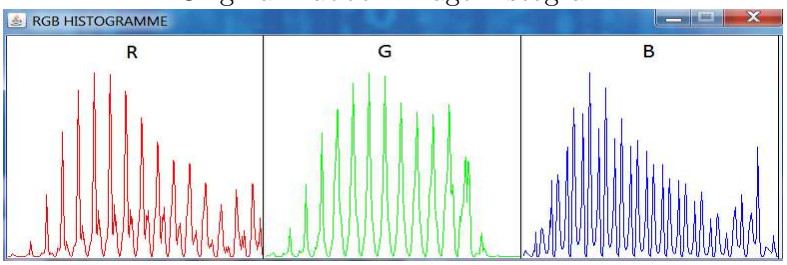
7.1. **Experimental results and security analysis.** Key space size is the total number of different keys that can be used in the encipherment. For instance, to decrypt an image of size 512×512, the key should be chosen from a key space of size exceeding $2^{2^{18}}$ which is large enough to resist all kinds of brute-force attacks. Experimental analysis has been done with Baboon and Peppers, the well known images. Figure1 and Figure2 are the 512×512 RGB-scale plain-images and their encrypted images using One-Time Pad algorithm based on the secret key provided by *BCB*. The most important measure for the quality of any cryptosystem comes from its capability to withstand the attempts of an unauthorized participant to gain knowledge about the unencrypted information. This measure is called security. In the following, we analyze the security of the proposed protocol by enciphering images using distinct keys as follows: key space analysis, histogram analysis, information entropy and correlation analysis. Finally, we end up by a quality control. The performed experiments were done on a $2,6$ GHz Intel Core Processor $i7-5600U$ CPU.

7.1.1. *Histogram analysis.* In the right side of Figure 1 and Figure 2 are depicted the plain-images and cipher-images histograms for Baboon and Peppers. From visual perception, it is clear that the enciphered images histograms, approximated by a uniform distribution, is quite different from plain-images histograms. Uniformity caused by the One-Time Pad and the used key is justified by the chi-square test [10]. Therefore, the encipherment key provided by the proposed protocol does not provide any clue for statistical attack.

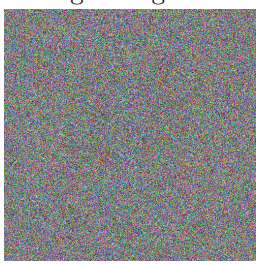
Original Baboon image



Original Baboon image histogram



Encrypted Baboon image using *BCB*



Encrypted Baboon image histogram

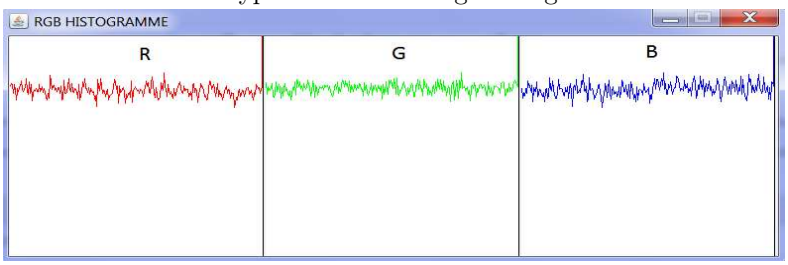
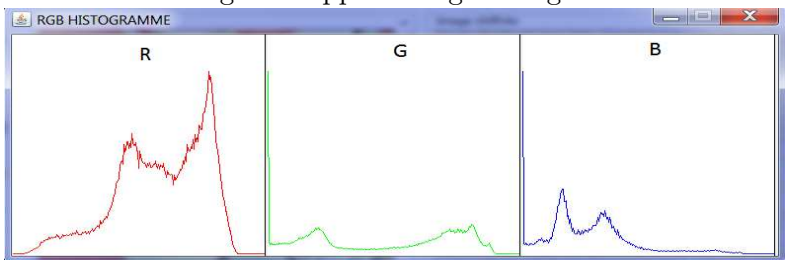


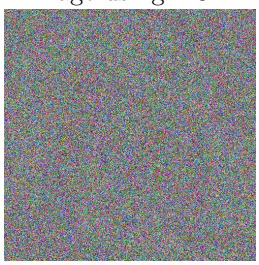
Figure1. Baboon image and its histograms

Original Peppers image

Original Peppers image histogram



Encrypted Peppers image using *BCB*



Encrypted Peppers image histogram

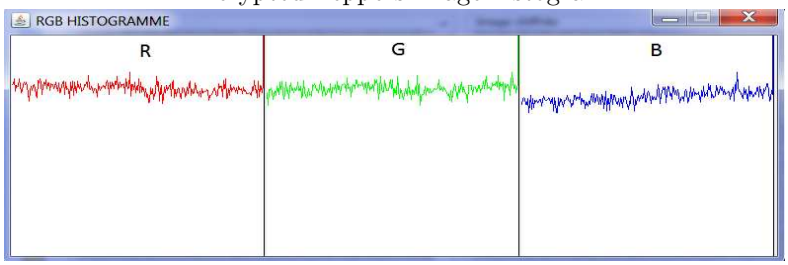


Figure2. Peppers image and its histograms

ROMANIAN JOURNAL OF MATHEMATICS AND COMPUTER SCIENCE, 2019, VOLUME 9, ISSUE 2, p.78-86

7.1.2. *Information entropy.* The concept of entropy was introduced by Shannon [1]. The Entropy of a source of information m is a mathematical measure of the amount of information provided by an observation of m. Equivalently, it is the uncertainty about the outcome before an observation of m.

To calculate the entropy of a source m, we have:

$$H(m) = \sum_{i=1}^{2^N} P(m_i) \log_2 \left(\frac{1}{P(m_i)} \right),$$

where $P(m_i)$ represents the probability of occurrence of symbol m_i. Let us suppose that the source emits 2^8 symbols with equal probability, i.e., $m = \{m_1, m_2, \ldots, m_{2^8}\}$. After evaluating Equation 1, we obtain its entropy $H(m) = 8$, corresponding to a truly random source. Actually, given that a practical information source seldom generates random messages, in general its entropy value is smaller than one. However, when the messages are enciphered, their entropy should ideally be 8. If the output of such a cipher emits symbols with entropy less than 8, there exists certain degree of predictability, which threatens its security.

Let us consider the cipher-images obtained by One-Time Pad algorithm using four secret keys provided by *BCB*. The following table summarizes the experimental results that we obtained:

Images	Key1	Key2	Key3	Key4
Baboon	7.99832	7.99922	7.99861	7.99651
Peppers	7.99843	7.99863	7.99851	7.99739

TABLE 1. Entropy of the enciphered images

7.1.3. *Correlation analysis of two adjacent pixels.* The superior confusion and diffusion properties are showed by a test on the correlations of adjacent pixels in the ciphered image [8]. We calculated and analyzed the correlation between diverse plain-image and its respective cipher-image pairs. The following formula was used to evaluate the correlation coefficient of the Baboon's and Peppers's cipher-images provided by One-Time Pad using four distinct keys:

$$Coef_{P,C} = \frac{\sum_i^H \sum_j^W (P_{i,j} - \bar{P})(C_{i,j} - \bar{C})}{\sqrt{\left(\sum_i^H \sum_j^W (P_{i,j} - \bar{P})^2\right) \cdot \left(\sum_i^H \sum_j^W (C_{i,j} - \bar{C})^2\right)}},$$

where P and C are plain-image channel and cipher-image channel, $\bar{P} = \frac{1}{H \times W} \sum_i^H \sum_j^W P_{i,j}$ and $\bar{C} = \frac{1}{H \times W} \sum_i^H \sum_j^W C_{i,j}$ are there mean values respectively, H and W are the height and the width of the plain-image and cipher-image. Table 2 bellow shows the correlation coefficients of the two adjacent pixels in the Baboon's and Peppers's plain-images and there cipher-images respectively.

Images	Key1	Key2	Key3	Key4
Baboon	0.006761	-0.004919	-0.006861	-0.0073922
Peppers	0.004241	-0.018511	-0.0092430	-0.005957

TABLE 2. Coefficient correlations of the enciphered images

7.1.4. *Image Encipherment Quality.* Image Encipherment Quality (IEQ) measure is a figure of merit used for the evaluation of image encipherment techniques and for the evaluation of the key has been used. With the encipherment of an image a change takes place in pixels values as compared to those values before encipherment. Such change may be irregular. This means that the higher the change in

pixels values, the more effective will be the image encipherment. Hence, the key quality used for the encipherment. Therefore, the IEQ may be expressed in terms of the total changes in pixels values or the deviation between the plain-image and the enciphered one [9]. The quality of image encipherment may be determined as follows. Let P and C denote the plain-image and the cipher-image respectively, each of size $H \times W$ pixels with L grey levels. $P(x,y),\ C(x,y) \in 0,\ldots,L-1$ are the grey levels of the images P and C at position (x,y), $0 \leq x \leq H-1,\ 0 \leq y \leq W-1$. We will define $H_L(P)$ as the number of occurrence for each grey level L in the plain-image, and $H_L(C)$ as the number of occurrence for each grey level L in the cipher-image. The IEQ represents the average number of changes to each grey level L and it can be expressed as:

$$IEQ = \frac{\sum_{L=0}^{255} |H_L(C) - H_L(P)|}{256}.$$

The following table illustrates the IEQ of the encipherment of Baboon and Peppers images using four distinct keys provided by *BCB*.

Images	Key1	Key2	Key3	Key4
Baboon	68,359	69.9218	67.08	69.72
Peppers	70.019	67.285	67.87	70.410

TABLE 3. Encipherment Quality of the enciphered images

8. Conclusion

In this paper, a new protocol of sharing secret keys is proposed. The protocol provides secret keys which improve the security of encipherments using One-Time Pad, based on set partitioning problem. In fact, given a fixed secret partition, the two parties involved in the secret plaintext exchange can separately unroll the protocol to generate a secret key to cipher the plaintext and decipher the ciphertext. In all secret key encryption systems, the parties concerned by the secret information exchange have to share the secret encryption key, that fact constitutes a weakness for those systems. Our prosed protocol eliminates this drawback carrying out two objectives: the production of a pseudo-random key at least as long as the message to be enciphered with assurance of the synchronization between the transmitter and the receiver, and the inability of a third person, to determine the generated secret key in a reasonable time. Some experiments have been done to evaluate the performance of the proposed protocol, we enciphered images by the One-Time Pad using the provided key by *BCB*. Experimental results indicate that the cipher-image histograms distribution are so even that the entropy measured is almost equal to the ideal value. The measured IEQ of four different encipherment keys is good. Correlation analysis showed that correlation coefficients between adjacent pixels in the plain-images and their corresponding cipher-images are significantly decreased. Consequently, the proposed protocol realizes the security goals and resists brute attacks.

References

[1] C.E. Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, Volume: 28 , Page: 657-715, 1949.

[2] D. Stebila. Classical Authenticated Key Exchange and Quantum Cryptography. *Thesis for degree of doctor of philosophy in Combinatorics and Optimisation, University of Waterloo, Ontario, Canada*, 2009.

[3] S. Vaudenay. A Classical Introduction to Cryptography: Applications for Communications Security. *Swiss Federal Institute of Technologies, Springer Science+Bussiness Media*, 2006.

[4] G. S. Vernam. Cipher Printing Systems for Secret Wire and Radio Telegraphic Communications. *J. AIEE 45*, Pages: 109-115, 1926.

[5] G. Dobinski. Summing der Reihe $\sum \frac{n^m}{n!}$ for $m = 1, 2, 3, 4, 5, \ldots$ *Grunert's Archiv*, Volume 61, 1877, Pages 333−336.
[6] L. Moser, M. Wyman. An asymptotic formula for the Bell numbers. *Trans. Royal Soc. Canada III*, Volume 49, Pages 49−54, 1955.
[7] J.L. Roch. Security Models: Proofs, Protocols and Certification, Master2-Security. *Cryptography and Coding of Information Systems ENSIMAG-INP-UJF, Grenoble university, France*, 27 Octobre 2007.
[8] G. Chen, Y. Mao, C. Chui A symmetric image encryption scheme based on 3d chaotic cat maps. *Chaos, Solitons Fractals 21*, Pages 749-761, 2004.
[9] A. Jolfaei, A. Mirghadri. Survey: Image Encryption Using Salsa20. *International Journal of Computer Science Issues*, Volume 7, Issue 5, September 2010.
[10] P. L'Ecuyer, R. Simard. TestU01: A C Library for Empirical Testing of Random Number Generators. *ACM Transactions on Mathematical Software*, Volume 33, Number 4, Article 22, August 2007.

University of Sciences and Technology Houari Boumediene, Faculty of Mathematics, P.Box 32 16111 El-Alia, Bab-Ezzouar, Algiers, Algeria
E-mail address: sbouroubi@usthb.dz or bouroubis@gmail.fr

University of Sciences and Technology Houari Boumediene, Faculty of Mathematics, P.Box 32 16111 El-Alia, Bab-Ezzouar, Algiers, Algeria
E-mail address: fellacharchali@gmail.com

Algiers 3 University, Algiers, Algeria
E-mail address: benyahiatani@yahoo.fr

ROMANIAN JOURNAL OF MATHEMATICS AND COMPUTER SCIENCE, 2019, VOLUME 9, ISSUE 2, p.78-86