THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 13, Number 2/2012, pp. 157–166

A CLOSER VIEW OF RUNNING-KEY CIPHER ON NATURAL LANGUAGES AND ITS EXTENSION FOR NEW APPLICATIONS IN CRYPTOGRAPHY*

Adriana VLAD[1,2], Azeem ILYAS[1], Adrian LUCA[1]

[1]"Politehnica" University of Bucharest, Faculty of Electronics, Telecommunications and Information Technology,
1-3, Iuliu Maniu Bvd. Bucharest 6, Romania
[2]The Research Institute for Artificial Intelligence,
Romanian Academy, 13, Calea 13 Septembrie, Bucharest 5, Romania
E-mail: avlad@racai.ro, adriana_vlad@yahoo.com

The paper supports a debate concerning the meaning and the importance of redundancy and ergodicity in cryptography. The discussion is mainly organized around a very thought-provoking enciphering method, *running-key* cipher. The paper comes with a new view of *running-key* method which permits to resume and extend it for ergodic sources in order to provide good quality key generators for cryptography. We first apply the *running-key* method on natural language and then we extend it on logistic map. Finally some conclusions are drawn even for image enciphering.

Key words: running-key cipher, noisy channel, redundancy, ergodic source, logistic map, image encryption.

1. INTRODUCTION

In 1945, Claude Shannon wrote his results concerning *what is so specific in a message source that makes possible to break a cipher* in a classified paper, [1]. The content of the paper appeared in the open literature in 1949, [2] – a fundamental paper in the field; since then, cryptography turned from an art into science.

The notions of entropy, redundancy, ergodicity and uncertainty channel that Shannon worked out for his evolving information theory [3] *had found a perfect support in cryptography both for their perceiving and for their utility in practice*.

In what follows we pay attention to redundancy and ergodicity without trying to provide an overview on the main contributions existing in the large field of the information theory; the topic is too complex and it remains a challenging area of research. The aim is to provide more insight and to open a discussion on the effect of the redundancy and ergodicity in practical secrecy systems. That will be done by means of a new view of the *running-key* cipher approach with a perspective of new applications in cryptography.

First we shall recall some main issues Shannon defined and worked out in his theory on cryptography.

Redundancy – guilty of breaking ciphers

The above sentence was demonstrated by Shannon in the context of defining the *ideal cipher*. Let us have the discrete uncertainty channel with X and Y the input and output spaces (an input element from X is a plain message and an output element from Y is a cryptogram).

Be a closed secrecy system (a secrecy systems where the total number of possible messages is equal to the number of possible cryptograms). In [2], it is shown that if there is no redundancy on the X message space, then any cipher, even a very simple one, is an *ideal cipher* and the cryptanalyst will never find the key. In this situation the message equivocation (the secrecy amount of the cipher) will differ from zero, so there is no unique solution for the cipher.

* This paper was presented in part at the 1st Conference *Romanian Cryptology Days* – RCD2011, October 11–12 2011, Bucharest, Romania.

What practically means zero redundancy?

Let us consider the input elements from X corresponding to a sequence of symbols emitted by a message source having q symbols in the alphabet, $S=\{s_1,s_2,\ldots,s_q\}$. Zero redundancy on the X message space ($R_X=0$) means in fact a zero-memory information source S with equally likely symbols in the alphabet (corresponding to the throwing of a fair dice model).

Obviously, in practical messages there is redundancy and Shannon illustrated his theory considering two types of ergodic message sources: zero-memory information sources with various redundancy values and multiple ergodic Markov chains approximating to Natural Languages (NL).

R_S redundancy assigned to the S source is computed as $R_S=\log q-H(S)$. The relative redundancy is $\rho_S=1-H(S)/\log q$. For the S zero-memory source, the entropy is $H(S)=-\sum_i^q P(s_i)\log P(s_i)$.

For Natural Languages, which are well approximated by multiple Markov ergodic chains, the entropy is calculated by a series of approximations of the m-gram entropy [4]:

$$H(S)=-\sum_i\sum_j P(b_i,j)\log P(j/b_i).$$

b_i is a block of $m-1$ adjacent letters ($m-1$ gram); theoretically, there are q^{m-1} distinct b_i blocks in total. j is the letter following b_i. $P(j/b_i)$ is the conditional probability that letter j follows block b_i. $P(b_i,j)$ is the probability of the m-gram (b_i,j). For small m values ($m\le 6$) the entropy can be evaluated from the statistical model existing in most of NL [4–6]. For large m values there is an interesting solution to approximate $H(S)$ by using *running-key* cipher [2, 7]. Note that NL are considered to be well approximated by multiple ergodic Markov chain with the multiplicity order larger than 30.

Coming back to the interest in cryptography, an *ideal cipher* cannot be put into practice because one cannot get rid of the redundancy from the message (that would imply to code the message by using Shannon's source coding theorem). Practical ciphers do not try to eliminate the redundancy, neither to diminish it, but to diffuse the redundancy on large linguistic entities. Thus, the discussion is shifted to the *diffusion* and *confusion* as features of a good practical cipher.

The statement that the redundancy is guilty for enabling to break a code does not imply the fact that one cannot design unbreakable ciphers still preserving the message space in its natural form (with intrinsic redundancy). At this moment we bring into discussion Vernam type cipher and *running-key* cipher; for both of these ciphers the cryptogram is obtained by summing up modulo q the message and the key (letter-by-letter), as in Fig. 1.

Shannon defined *perfect secrecy* systems and also demonstrated that the Vernam cipher (or one-time pad in which a coin-tossing is added bit-by-bit modulo 2 to the message) is unbreakable [8]. In the Vernam cipher the key is a sequence of symbols from the same alphabet as the message source and of the same length as the message, with the specification that the key source is a zero-memory source with zero redundancy. The Vernam type cipher is a *perfect cipher* (having the same number of messages, keys and cryptograms, and equally likely keys). It is clear from [2] that a *perfect cipher* does not provide any information to the cryptanalyst, no matter the message source will be (how much redundancy it contains), even if we extend the X message space such as to correspond to other messages sources, either ergodic or not. However, its use is very much restricted in practice concerning the keys management.

For *running-key* cipher, the key is also a sequence of the same length as the message, but it is a meaningful message (a typical sequence like the plain message). For example, if we encipher English language texts, then the keys will be also natural texts in English. In Section 2, we shall see that there are variants of *running-key* cipher that lead to unbreakable ciphers.

To conclude with, practical ciphers do not eliminate, generally neither diminish the redundancy, but diffuse the redundancy on long message units. On the other hand, non-zero redundancy on the message space (that is to consider natural sources with their intrinsic redundancy) does not mean that one cannot design

unbreakable ciphers. The above results of Shannon's theory represent a challenge both for the designing of ciphers with good diffusion property and for pseudo-random generators inspired from the requirements of the *perfect cipher*.

	Vernam cipher													*Running-key* cipher											
x_i (plain message):	T	H	I	S	I	S	M	E	S	S	A	G	E	T	H	I	S	I	S	M	E	S	S	A	G	E
k_S (key):	V	S	D	X	Q	C	H	O	Y	Z	E	Z	Y	A	R	E	Y	O	U	A	F	R	A	I	D	O
y_j (cryptogram):	B	R	Y	C	L	H	G	F	D	E	R	S	P	G	L	Z	D	J	Z	Z	W	W	F	V	W	F
$y_j=(x_i+k_s)$ mod 26																										

Fig. 1 – Example for Vernam cipher and *running-key* cipher for English language with $q = 26$ letters in alphabet.

Section 2 comes with a closer look at the *running-key* method on NL, illustrated on Printed Romanian. We are searching for a lesson to be learned out of the *running-key* in order to take advantage for developing new key generators useful in cryptography.

Section 3 addresses to the chaos-based cryptography by resuming and extending the *running-key* cipher with the purpose to point out good quality enciphering keys. In Section 3 we also depict the role of redundancy in cryptography by means of image enciphering.

2. STUDY ON RUNNING-KEY CIPHER UNDERLYING THE INVOLEMENT OF REDUNDANCY AND ERGODICITY

We shall next analyze the *running-key* method on NL (here illustrated by the Printed Romanian), trying to bring into evidence issues related to the practical significance, perception and numerical evaluation of existing redundancy in natural languages, implying into discussion the ergodicity of message source. More than that, it puts into evidence the role of redundancy in the cryptography, this time regarding not only the redundancy existing in the message source but also in the keys-source.

Figure 2 presents several versions of the application of *running-key* cipher, each time X_1 representing a clear message (plaintext). In first version of Fig. 2, cryptogram Y_1 is obtained by summing up clear message X_1 with the key T_1 which is also a natural text. In version 2 of the encryption, the key consists of two natural texts, T_1 and T_2. Cryptogram Y_2 is obtained as the sum of three natural texts, X_1, T_1 and T_2. Similarly, for variant $n \geq 3$, cryptogram Y_n is obtained by summing up $n+1$ natural texts X_1, T_1,T_2,T_3,...,T_n (where the n natural texts, T_1, T_2,T_3,...,T_n, stand for the key).

The encryption process was applied on Romanian language in five variants (meaning 6 natural texts). To have in depth analysis of the running key approach we worked on a literary Romanian corpus consisting of 58 books previously used in some studies dedicated to modelling printed Romanian [5]; the size of this corpus is of 29 293 212 characters. In the analysed corpus, the alphabet consists of the 31 letters specific to the Romanian alphabet without blank, orthography and punctuation marks. The 31 letters are here presented in decreasing order of their relative frequencies (presented as percentage in parentheses):

E(11.47); I(9.96); A(9.95); R(6.82); N(6.47); U(6.20); T(6.04); C(5.28); L(4.48); S(4.40); O(4.07); Ă(4.06); D(3.45); P(3.18); M(3.10); Ş(1.55); Î(1.40); V(1.23); F(1.18); B(1.07); Ţ(1.00); G(0.99); Â(0.92); Z(0.71); H(0.47); J(0.24); X(0.11); K(0.11); Y(0.07); W(0.03); Q (≈0,00)

The codes for A-Z were 0-25 (as in English alphabet, the order considered in ASCII). The codes for the Romanian diacritics are the numbers assigned to them in parentheses: Ţ (26), Î (27), Ş (28), Ă (29), Â (30).

Taking advantage of the ergodicity involved in NL, our main concern in the experimental study is to evaluate the statistical dependence/independence existing between the cryptogram and the plain message. So we apply several statistical tests to investigate the assumption of independence between the original message and cryptogram in the variants of Fig. 2.

The discussion on the statistical dependence between input message and cryptogram is next organized in two approaches: ***(A)*** *For a quick decision concerning statistical independence, we first apply a Chi-square test in contingency tables;* ***(B)*** *An approach with cascaded information channels where the input is the plain message and the output is the cryptogram, enabling us to reach a final conclusion on the statistical independence.*

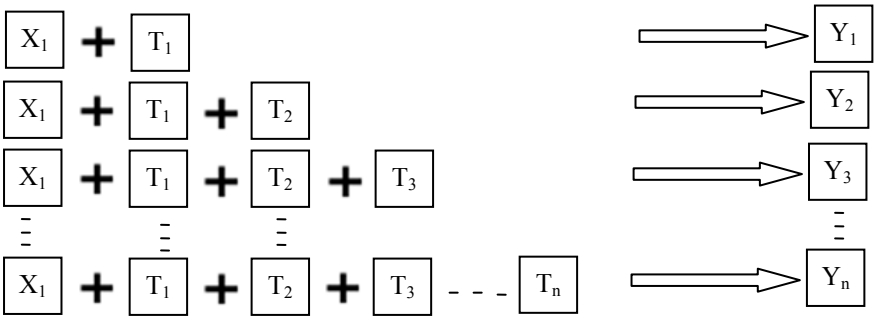


Fig. 2 – *Running-key* cipher variants.

(A) *A decision based on the Chi-square test in contingency tables*

The Chi-square test in contingency tables (a test of independence) [9, 10], has the following test hypotheses: the null hypothesis $H_0 : p(x_i, y_j) = p(x_i)p(y_j)$; the alternative hypothesis $H_1 : p(x_i, y_j) \neq p(x_i)p(y_j)$.

Note. x_i and y_j stand for letters in the source alphabet of the plain message and cryptogram, respectively.

The test is applied on *i.i.d* data pairs (data coming out from *independently* and *identically distributed* random variables) extracted from the message X_1 and cryptogram Y_n, where $n = 1, 2, 3, \ldots$. Figure 3 illustrates how to obtain the experimental *i.i.d.* data pairs submitted to test in the first version of *running-key* cipher, (X_1, Y_1). The method is similarly extended for the rest of five versions of the *running-key* cipher applied to Printed Romanian, namely (X_1, Y_2), (X_1, Y_3), (X_1, Y_4) and (X_1, Y_5) from Fig. 2. To obtain the *i.i.d.* data we applied a periodical sample with a large enough sampling period (200 symbols) so as to practically eliminate the dependency between successive symbols in the NL texts. By shifting the sampling origin with one symbol, we could obtain 200 *i.i.d.* experimental data sets. (As a result of the NL ergodicity assumption, these 200 *i.i.d.* data sets are practically equivalent to each other in terms of the investigation made). Each sample obtained in this way consists of N observations (where $N = L/200$, L is the length of the plain message). Note that the independence among the observations is a consequence of the large sampling period and identically distribution derives from the stationary hypothesis (involved by the ergodicity assumption). The way to extract the 200 *i.i.d.* data sets follows the procedure for investigating NL stationarity [5, 6].

Note. In this study $L = 4.5 \times 10^6$ characters and $N = 22\,500$.

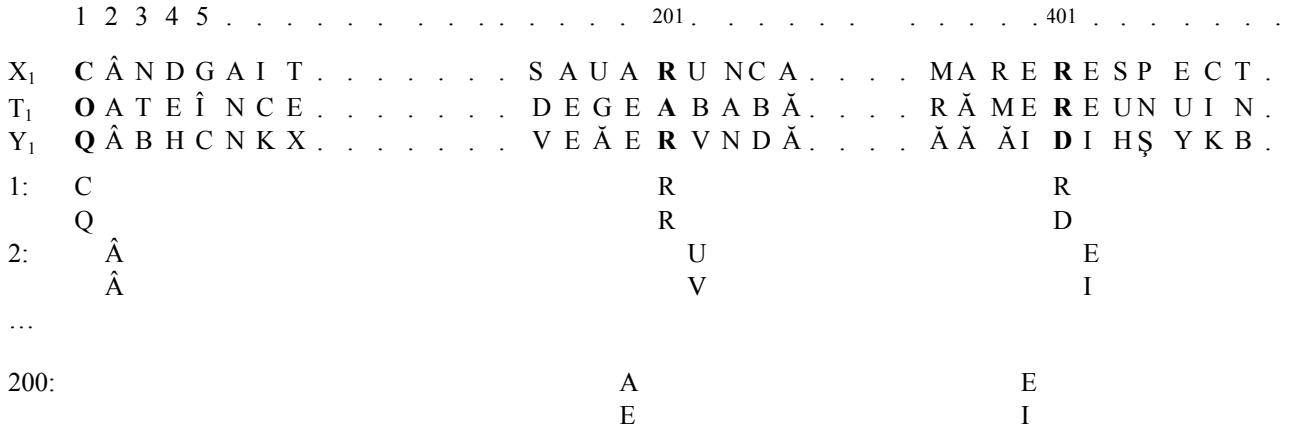
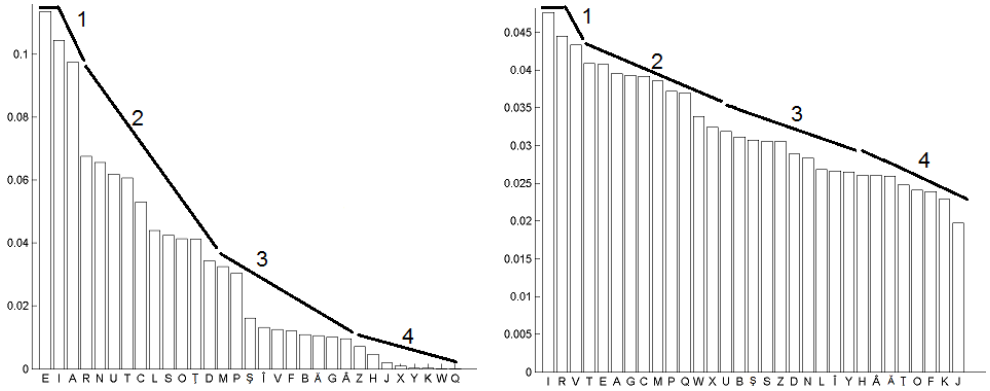


Fig. 3 – How to obtain the 200 sets of *i.i.d.* pairs.

For a quick result of the Chi-square test in contingency tables, the symbols in the alphabet are grouped into four classes according to the descending hierarchy of the relative frequencies of letters. Figs. 4(a) and (b) illustrate how we organized the four classes for investigating the independence between X_1 plain message and Y_1 cryptogram.



a b

Fig. 4 – Organization of the four classes: a) X_1 plain message; b) Y_1 cryptogram.

The test is based on the z test value:

$$z = \sum_{i=1}^{4}\sum_{j=1}^{4}\left(m_{ij} - \frac{m_i m_j}{N}\right)^2 / \left(\frac{m_i m_j}{N}\right). \tag{1}$$

In relation (1), N is the *i.i.d.* data size. m_i is the occurrence number of the i class assigned to the plain message and m_j is the occurrence number of the j class assigned to the cryptogram. m_{ij} is the occurrence number assigned to the pair *(i, j)* in the *i.i.d.* data. For example, m_{13} represents the occurrence number of a pair consisting of letters from class 1 assigned to X_1 plain and letters from class 3 assigned to Y_1 cryptogram. If z test value satisfies the condition $z \le z_\alpha$ (where z_α is the α point value corresponding to the Chi-square law of $9 = (4-1)^2$ degrees of freedom), we accept the null hypothesis H_0, thus we accept the statistical independence between the input message and cryptogram. We considered $\alpha = 0.05$ significance level, that leads to $z_\alpha = 16.90$.

All the 200 *i.i.d.* data sets were submitted to the quick Chi-square test in contingency tables (the *i.i.d.* data size is $N = 22\,500$). As the 200 *i.i.d.* data sets are *a priori* equally good to convey the information whether the statistical independence exists or not between plain and cryptogram, Table 1 presents in column 2 the proportion of accepting H_0 hypothesis out of 200 data sets, each row corresponding to the investigated cryptogram Y_n (where $n = 1, 2, ..., 5$).

Table 1

Running-key approach on NL: results for the statistical independence between plain message and cryptogram

	Chi-Square test in contingency tables pair (X_1 ,Y_n)	Chi-Square test goodness of fit (One tail)	ε_r^* relative departure for letter structure in the cryptogram	No. of equally likely digrams in the cryptogram (within the 5% error)
(1)	(2)	(3)	(4)	(5)
Y_1	0.00	0.00	0.4769	118 (0.122789)
Y_2	0.00	0.00	0.1091	407 (0.423517)
Y_3	0.795	0.78	0.0343	909 (0.945890
Y_4	0.935	0.945	0.0108	958 (0.996878)
Y_5	0.955	0.95	0.0061	961 (1.000000)

* $\varepsilon_r = \max\left|\hat{P}_i - (1/31)\right| / (1/31)$; $\hat{P}_i$ values stand for the relative letter frequencies in Y_n sequence

Important remarks: **(1)** The 200 *i.i.d.* samples as shown in Fig. 3 are not independent data sets, so we cannot assign a confidence level to the respective proportion in column 2. However, the proportion conveys information about the possibility to break a cipher and also about the homogeneity involved by the results. **(2)** The statistical independence illustrated in Table 1 is carried out only on the first order statistics of the assigned NL random process**. (3)** We performed the independence test in contingency tables up to five

additions meaning 6 natural texts to be summed up, *i.e.* more than the practical use for the cryptanalysis. We did that in order to get more insight regarding the dependence/independence relationship between plain message and cryptogram. The results are put together in Table 1, column 2 and show that after four additions (meaning Y_4 or Y_5), the dependence decreases much, so that we can practically consider that Y_5 (even Y_4) cryptogram is statistically independent of the plain message. Note that we performed the Chi-square test in contingency tables organizing the alphabet in four classes in various ways (not only like in Fig. 4) to make sure that this does not affect the independence decision and the experimental results remain similar to those presented in Table 1, column 2.

(B) *Statistical independence decision based upon cascaded information channels*

The paper comes with a new view of the schema in Fig. 2, which in fact puts into evidence Shannon's basic ideas representing the cipher by information channel, see Fig. 5. In fact, based on this schema of cascaded information channels, we can draw a conclusion concerning the statistical independence in the *running-key* approach. More exactly, we want an answer concerning how many additions we have to make in the *running-key* method so that the final Y_n cryptogram is practically independent of the plain message.

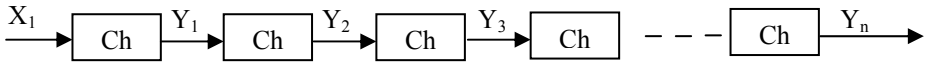


Fig. 5 – Information channel associated with the Y_S sequences.

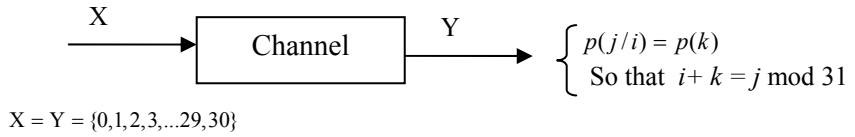


Fig. 6 – An information channel depiction.

As it is known, an information channel is formally described by the input/output alphabet (X/Y) and the noisy matrix. All the channels in Fig. 5 are identical and uniform zero-memory noisy channels. Fig. 6 describes one of these channels; the input and output alphabet are the same and consist of numbers {0, 1, 2 ,..., 30}. Each channel corresponds to the encryption of one letter in the simplest variant of the *running-key* approach, which means adding two texts to obtain a cryptogram. So, for the channel matrix, the notations *i*, *j* and *k* stand for the input, output and key symbols and they are connected by the relationship from Fig. 6. Each row in the noisy matrix corresponds to a fixed *i* input and the terms in the respective row of the noisy matrix are just the probabilities assigned to NL letter coded by *k* ; *p(j/i)* is a probability of obtaining *j* symbol at the output of the channel if the *i* input symbol is sent, $p(j/i) = p(k)$. Note that every row in the noisy matrix assigned to the channel in Fig. 6 is a permutation of the terms in the first row, so we have a *uniform information channel*.

Let us first discuss the case when the input from Fig. 5 corresponds to the zero-memory information source approximating NL. Thus a typical sequence emitted by the input source of Fig. 5 will be any one of the 200 data sets sampled from NL texts as in Fig. 3 and consequently the respective output of the first channel will be the *i.i.d.* data extracted from the cryptogram Y_1 like in Fig. 3.

According to the known theory (information theory and cryptography), from Fig. 5 it results the following relation between the source entropies involved in the cascaded channels

$$H(X_1) < H(Y_1) < H(Y_2) < H(Y_3) < ... < H(Y_n) < \log_2 31.$$

If we want Y_n to be statistically independent from X_1, such a result can be reached if and only if Y_n is a zero-memory information source having zero-redundancy, *i.e.* $H(Y_n) = \log_2 31$. Such a result can be achieved only asymptotically. It is clear from the channel matrix from Fig. 6 that Y_1 is strongly dependent on X_1. We arrive at an important result from the point of view of statistical independence testing between plain and cryptogram: for more than two text additions (that is Y_2, Y_3,...Y_n), one can investigate the statistical

independence between cryptogram and plain message only based on Y_n information source, namely checking if the respective Y_n complies with the uniform discrete probability law (here fair dice model with number of faces equal to 31). A fast way of testing is by using Chi-square goodness of fit test over a typical *i.i.d.* sequence emitted by the Y_n information source. The Chi-square goodness of fit test has the following test hypotheses: the null hypothesis $H_0: p(j) = 1/31$ and the alternative hypothesis $H_1: p(j) \neq 1/31$. j stands for the letter in the alphabet. The test is based on the z test value (2):

$$z = \sum_{j=1}^{31} (m_j - N/31)^2 / (N/31). \tag{2}$$

If z test value satisfies the condition $z \leq z_\alpha$ (where z_α is the α point value corresponding to the Chi-square goodness of fit having $30 = 31 - 1$ degrees of freedom), we accept the null hypothesis H_0, thus we accept the statistical independence between the input message and cryptogram. We considered $\alpha = 0.05$ significance level that leads to $z_\alpha = 43.773$.

We performed the Chi-square goodness of fit test for the five additions meaning 6 natural texts to be summed up, *i.e.* investigating five outputs (cryptograms) of the cascaded information channels in Fig. 5. The results are put together in Table 1, column 3 based on the 200 *i.i.d.* data sets. We recall that all the 200 *i.i.d.* data sets submitted to Chi-square test were sampled from the cryptogram with a 200 period of symbols (like in Fig. 3), so that the *i.i.d.* data size is $N = 22\,500$. As the 200 *i.i.d.* data sets are *a priori* equally good to convey the information about the first order probability law of the cryptogram, Table 1 presents in column 3 the proportion of accepting H_0 hypothesis of the test out of 200 data sets for each investigated cryptogram. The results from column 3 led to the conclusion that Y_4 and Y_5 may comply with a fair dice (discrete uniform probability law). We add some more information in column 4 of Table 1, namely on the maximum relative departure from 1/31 of the relative letter frequencies in cryptogram. Column 5 gives the information about how many digrams (group of two successive letters) in the cryptogram are practically equally likely within a relative error of 5% (*i.e.* relative frequency $\frac{1}{961}(1 \pm 5\%)$). In column 5 we assigned in parentheses the proportion of the equally likely digrams out of the total number of distinct possible digrams.

To conclude with Section 2

Let us recall the known results from the literature that relative redundancy for English is about 75% [2, 7]; this derives from the fact that *running-key* cipher applied to natural text in variants 1 and 2 can be broken, but not in version 3. (Such experiments have been done on printed Romanian and results indicated approximately the same interval for redundancy). These results are supported by our experiments here in this study on printed Romanian and we underline that the condition to obtain a statistical independence between the plain message and cryptogram implicitly means that the *running-key* cipher is unsolvable by the cryptanalyst. However, this condition is more severe than the one that Y_3 cannot be decomposed into the NL summed up texts (for example, Y_3 cannot be decomposed into the four natural summed up texts; however Y_3 is still dependent upon plain message). On the other hand, by looking at the cascaded information channels in Fig. 5 and Table 1, we arrive at an interesting, very important conclusion: Y_4 stands for a good key generator with practically zero redundancy. This idea can be further fructified in developing new encryption methods as we tried to illustrate in the next section of the paper.

3. EXTENDING THE RUNNING-KEY APPROACH FOR OTHER ERGODIC SOURCES

The *running-key* approach as described in Fig. 2 could be shifted from NL to chaos due to the ergodicity feature of chaotic systems and also because one can extract *i.i.d.* data sets from chaotic maps. While the ergodicity of the chaotic signals is generally assumed [12], the statistical independence seems to be in contradiction with the deterministic feature of chaotic signals and it was a challenging research in this respect [13–16]. Our intention is to extend the *running-key* approach on the chaotic maps and to see whether it is possible or not to use them as a good quality key generator in cryptography. Further on an investigation is made for image encryption based upon *running-key* approach using chaotic map.

The entire procedure to investigate the statistical independence between cryptogram and plain message based on *running-key* is resumed for logistic map, (3), as it is done for NL in Section 2.

$$z_{t+1} = R\, z_t (1 - z_t), \quad t = 0,1,2,... \tag{3}$$

The R parameter of the logistic map belongs to the $(0; 4]$ interval and the z_t values belong to the $(0; 1)$ interval [11, 12]. The illustrations in Table 2 are for $R = 4$. A typical sequence emitted by the logistic map like X_1, T_1, T_2, T_3 or T_4 is specified by an initial condition randomly chosen in the $(0; 1)$ interval. For the logistic map we consider an alphabet consisting of 32 letters to further simplify the image encryption illustration and also for a comparison with the NL. The continuous values z_t are transformed into 32 discrete values (the 32 discrete values are obtained by a partition of $(0; 1)$ interval in 32 adjacent non-overlapping equal lengths subintervals). To apply the Chi-square test in contingency tables, we organized the 32 symbols of the alphabet in four classes (following again the descending hierarchy of relative frequencies of the 32 symbols as we described for printed Romanian).

Note. In [13, 14] it was shown that for the logistic map and various *R* parameters, a sampling distance of about 30 iterations ensures the statistical independence between the extracted values. In fact for $R = 4$ the sampling distance which ensures statistical independence is about 15 iterations, but for a first evaluation by using *running-key* method we considered a sampling period of 30 symbols. We iterated the logistic map up to 4.5×10^6 times and that provided a size for the *i.i.d.* data sets equal to N=150 000. Here we are not limited to work on the experimental data as in printed Romanian. We could extend our study up to six additions (adding seven typical sequence emitted from the logistic map) meaning Y_6.

Table 2

Running-key approach on logistic map: results for the statistical independence between X_1 input and Y_n output

	Chi-square test in contingency Table pair(X_1 , Y_n)	*Chi-square goodness of fit* Test	ε_r^* for letter structure in cryptogram	No. of equally likely digrams in cryptogram (within the 5% error)
(1)	(2)	(3)	(4)	(5)
Y_1	0.00	0.000	0.5132	28 (0.0273)
Y_2	0.00	0.000	0.1025	118 (0.115)
Y_3	0.00	0.2000	0.0245	304 (0.297)
Y_4	0.73	0.8667	0.0097	560 (0.547)
Y_5	0.966	0.8667	0.0058	881 (0.860)
Y_6	0.933	0.9667	0.0043	1015 (0.991)

* $\varepsilon_r = \max\left|\hat{P}_i - 1/32\right| / (1/32)$; $\hat{P}_i$ values stand for the relative letter frequencies in Y_n sequence.

The results (Table 2) for the logistic map with $R = 4$ are similar with those obtained for NL, leading to some viewpoints of practical interest. Note that in case of chaotic maps X_1 will not act as a plain message; we only used the *running-key* approach to benefit from the results for the design of a good quality key generator for cryptographic applications. Here the independence between the X_1 input message and output message practically occurred at Y_5.

The *running-key* approach cannot be extended to images, because they do not feature ergodicity, we can neither speak about *i.i.d.* data sets (so, we cannot apply the statistical tests described above). However, some benefits can be obtained for image enciphering from the investigations presented in Table 2.

Figure 7 shows results for image encryption. The image encryption is illustrated using the variants of Fig. 2. The image stands for X_1 and the key enciphering sequences T_1, T_2, T_3, T_4 and T_5 are generated by the logistic map with $R = 4$. We considered images with 32 gray levels. Fig. 7(a) is the original image X_1. Fig. 7(b) represents cryptogram C_1 obtained as summation between X_1 and T_1. Fig. 7(c) is obtained as a summation between X_1, T_1 and T_2. Fig. 7(d) is obtained by summing up X_1 original image and T_1, T_2 and T_3 chaotic sequences. Similarly, Fig. 7(e) is obtained by summing up X_1 with 4 chaotic sequences and Fig. 7(f) is obtained by summing up X_1 with 5 chaotic sequences.

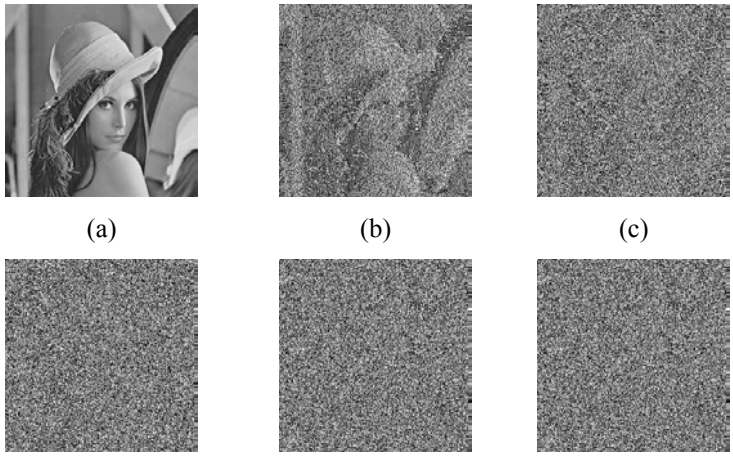


Fig. 7 – Image encryption using the extension of *running-key* on logistic map:
a) original image; b) cryptogram C_1; c) cryptogram C_2;
d) cryptogram C_3; e) cryptogram C_4; f) cryptogram C_5.

Image encryption in Fig. 7 has not the encryption procedure by itself as a main purpose, but to underline that in order to decide on cipher quality, the simple inspecting of the image cryptogram is not sufficient. For example, the image cryptogram given in Fig. 7(d) looks like a pure noise, but the key sequence (obtained by summing up 3 chaotic sequences, equivalent to Y_2 in Table 2) is still redundant and can be sensitive to a statistical attack.

4. FINAL CONCLUSIONS

The paper attempts to come with a lesson out of the *running-key* approach, so that one can resume and extend its application to ergodic sources other than NL. The paper provides a new view of *running-key* method by using the cascaded information channels to enable an easier but pertinent way to decide upon the statistical independence between the input message and the cryptogram. The study was carried out in various variants of the *running-key* method, adding numerical support to the existing results in the literature which were mainly based on cryptanalysis success. One may extend this study on various chaotic systems to benefit from the results, both for evaluating the intrinsic redundancy of the chaotic signals and also in cryptography. The extension is mainly based on the ergodicity property and on the proved possibility of generating *i.i.d.* data starting from chaotic maps. Image encryption can also benefit from these results in the sense suggested in this paper.

REFERENCES

1. GALLAGER R.G., *Claude E. Shannon: A Retrospective on His Live, Work and Impact*, IEEE Trans. on Inform. Theory, **47**, *7*, pp. 2681–2695, Nov. 2001.
2. SHANNON C.E., *Communication Theory of Secrecy Systems*, Bell Syst. Tech. J., **28**, pp. 656–715, October 1949.
3. SHANNON C.E., *A Mathematical Theory of Communication*, Bell Syst. Tech. J., **27**, pp. 379–423, 623-656 1948.
4. SHANNON C.E., *Prediction and Entropy of Printed English*, Bell Syst. Tech. J., **30**, pp. 50–64, Jan. 1951.
5. VLAD A., MITREA A., MITREA M., *Limba română scrisă ca sursă de informaţie* (Printed Romanian Language as an Information Source), Editura Paideia, Bucureşti, 2003.
6. VLAD A., MITREA A., MITREA M., *A Corpus – based Analysis of how Accurately Printed Romanian Obeys Some Universal Laws*, Chap. 15 in *A Rainbow of Corpora: Corpus Linguistics and the Languages of the World*, Wilson, Andrew/Rayson, Paul/McEnery Tony Editors, Lincom-Europa Publishing House, Munich, 2003, pp. 153–165.
7. DIFFIE W., HELLMAN M., *Privacy and Authentication: An Introduction in Cryptography*, Proc. IEEE, **67**, *3*, pp. 397–426, 1979.
8. MASSEY J. L., *Shannon and Cryptograph*y, IEEE Information Theory Society Newsletter, Special Golden Jubilee Issue, Summer 1998.
9. CRAIU V., *Verificarea Ipotezelor Statistice*, Editura Didactică şi Pedagogică, Bucureşti, 1972.

10. DEVORE J., *Probability and Statistics for Engineering and the Sciences*, 2nd ed., Brooks/Cole Publishing Company, Monterey, California, 1987.
11. ŞERBĂNESCU Al. (coord.), *Aplicaţii ale sistemelor dinamice în comunicaţii*, Editura Academiei Tehnice Militare, Bucureşti, 2004.
12. LASOTA A., MACKEY M.C., *Chaos, Fractals, and Noise. Stochastic Aspects of Dynamics*, 2nd edition, Springer, Heidelberg, New York, 1994.
13. VLAD A., LUCA A., FRUNZETE M., *Computational Measurements of the Transient Time and of the Sampling Distance That Enables Statistical Independence in the Logistic Map*, Lectures Notes in Computer Science (ICCSA 2009), **5593**, Springer-Verlag, Berlin, Heidelberg, pp. 703–718, 2009.
14. BADEA B., VLAD A., *Revealing Statistical Independence of Two Experimental Data Sets. An Improvement on Spearman's Algorithm*, Lecture Notes in Computer Science (ICCSA 2006), Springer-Heidelberg, **3980**, pp. 1166–1176, 2006.
15. LUCA A., VLAD A., *Generating Identically and Independently Distributed Samples Starting from Chaotic Signals*, Proc. Intl. Symp. on Signal, Circuits & Systems – ISSCS 2005, Iasi, pp. 227–230, July 2005.
16. GONZALEZ J.A., TRUJILLO L., *Statistical Independence of Generalized Chaotic Sequences*, Journal of the Physical Society of Japan, **75**, *2*, p. 023003, February 1–4, 2006.