applied sciences

MDPI

Article

Bit Independence Criterion Extended to Stream Ciphers

Evaristo José Madarro-Capó [1], Carlos Miguel Legón-Pérez [1], Omar Rojas [2], Guillermo Sosa-Gómez [2,*] and Raisa Socorro-Llanes [3]

1 Institute of Cryptography, University of Havana, Havana 10400, Cuba; ejmcapo@gmail.com (E.J.M.-C.); clegon58@gmail.com (C.M.L.-P.)
2 Facultad de Ciencias Económicas y Empresariales, Universidad Panamericana, Álvaro del Portillo 49, Zapopan, Jalisco 45010, Mexico; orojas@up.edu.mx
3 Faculty of Informatics, Technological University of Havana (UTH), CUJAE, Havana 19390, Cuba; raisa@ceis.cujae.edu.cu
* Correspondence: gsosag@up.edu.mx; Tel.: +52-3313682200

Received: 30 September 2020; Accepted: 26 October 2020; Published: 29 October 2020

Abstract: The bit independence criterion was proposed to evaluate the security of the S-boxes used in block ciphers. This paper proposes an algorithm that extends this criterion to evaluate the degree of independence between the bits of inputs and outputs of the stream ciphers. The effectiveness of the algorithm is experimentally confirmed in two scenarios: random outputs independent of the input, in which it does not detect dependence, and in the RC4 ciphers, where it detects significant dependencies related to some known weaknesses. The complexity of the algorithm is estimated based on the number of inputs l, and the dimensions, n and m, of the inputs and outputs, respectively.

Keywords: bit independence criterion; bit independence; RC4; stream cipher; complexity

1. Introduction

Randomness is an essential component in the security of cryptographic algorithms [1,2]. In particular, stream ciphers are composed of pseudo-random number generators and base their security on the statistical characteristics of these generators [1]. Several stream ciphers can be found in the literature whose description is based on different methods for the generation of pseudo-random numbers [3].

In practice, to determine if a generator is suitable to be used for cryptographic purposes, several statistical tests are usually applied on it to measure the randomness of its outputs [4–6]. There are numerous statistical tests to measure the randomness of the outputs of a pseudo-random number generator, among these those grouped in the batteries of NIST [7], Diehard [8], TestU01 [9], and Knuth [10], among others [2]. However, despite a large number of statistical tests being present in these batteries, none of them measure the correlation between the inputs and outputs of the stream cipher; they only measure the randomness of the outputs, which is a necessary, but not sufficient, condition to consider the generator for use in cryptography.

To consider a stream cipher secure, there must be no statistically significant correlation between the structure of its inputs and outputs. If "patterns" depending on the structure of the cipher input are generated in the output of stream ciphers, this could provide information about the input used. In the literature, there are reports of cryptanalysis based on this type of weakness [11,12]. In this way, it is essential to avoid the previous weakness and to have methods to detect it in the design and evaluation stage of the algorithm; in particular, it is necessary to have statistical tests that are capable of detecting the existence of significant statistical dependencies between the inputs and outputs of stream ciphers.

In general, there are very few statistical test reports to detect the existence of statistical dependencies between the outputs and inputs of a stream cipher. Therefore, the design of statistical tests that allow for the evaluation of them in this sense is highly important in cryptography.

The strict avalanche criterion (SAC) and the bit independence criterion (BIC) were proposed in [13] to evaluate the strength of the S-boxes used in block ciphers [14]. These two criteria measure different characteristics of the change's effect that an input bit has on the output bits; while the SAC verifies uniformity in the distribution of each output bit, the BIC measures the degree of independence between the output bits [15]. The SAC has been extended to be applied to stream ciphers [16–22]. In [22], the RC4 stream cipher [23] was evaluated through the SAC and the existence of statistical dependence between the input bits and outputs of the RC4 was detected for inputs of large size. This confirms the results obtained in [24–27], where the existence of related inputs in RC4 was reported. The idea developed in [22] was to determine the behavior of the distribution of the bits in the output by changing any bit in the input. In the design of stream ciphers, the distribution behavior of the output elements must be uniformly distributed, regardless of the bit that is being changed at the input [5]. Otherwise, the outputs could provide information on the input bits, which constitutes a weakness that, in the worst-case scenario, could lead to an attack. A discussion of attacks on stream ciphers can be found in [28]. However, the BIC has not been applied, to the best of our knowledge, to assess the degree of statistical independence between the bits of the output stream ciphers from changing a bit of the input. In this paper, we propose an algorithm that extends this criterion to evaluate the degree of independence between the input bits and the outputs of the stream ciphers. The effectiveness of the algorithm was experimentally confirmed in two scenarios: random outputs independent of the input, in which it does not detect dependence, and in the RC4 cipher, where it detects significant dependencies related to some known weaknesses [22,24–26].

2. Preliminaries

A stream cipher can be viewed as a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^m$ that transforms a binary input vector $X = (x_1, \ldots, x_n)$ of n bits into a binary output vector $Y = f(X) = (y_1, \ldots, y_m)$ of m bits, where $n, m \in \mathbb{N}$. In [13], the difference between the outputs $Y = f(X)$ and $Y^i = f(X^i)$, corresponding to the inputs X and X^i, is called the avalanche vector and denoted by $V^i = Y \oplus Y^i$, where $X^i = X \oplus e_i$, with $1 \leq i \leq n$ and e_i the unit vector with 1 in the i-th component. In $V^i = Y \oplus Y^i = (v_1^i, v_2^i, \ldots, v_m^i)$ each $v_j^i \in \mathbb{F}_2$, with $1 \leq j \leq m$, is called an avalanche variable (see Table A1, Appendix A).

Given the set $D = \{X_1, \ldots, X_l\}$ of l inputs X_r of n bits, with $1 \leq r \leq l$, a binary matrix H^i is constructed for each e_i, $1 \leq i \leq n$. To construct the matrix H^i, the avalanche vectors $V_r^i = Y_r \oplus Y_r^i = (v_{r1}^i, v_{r2}^i, \ldots, v_{rm}^i)$ are calculated, with $Y_r = f(X_r)$, $Y_r^i = f(X_r \oplus e_i)$. It is said that f satisfies the BIC if, by changing any bit i in the l inputs $X_r \in D$, it is satisfied that every pair of avalanche variables $v_{\cdot j}^i$ and $v_{\cdot k}^i$ are independent, with $1 \leq j,\ k \leq m$. The matrix H^i will be called the SAC matrix associated with the vector e_i and is shown in Table 1.

To measure the degree of independence between the pairs of avalanche variables, Webster and Tavares [13] used Pearson's correlation coefficient. In [29], the maximum value of these coefficients was used as a test statistic, denoted here by

$$BIC_{Pearson}(f) = \max_{\substack{1 \leq i \leq n \\ 1 \leq j,k \leq m \\ j \neq k}} \rho(v_{\cdot j}^i,\ v_{\cdot k}^i). \tag{1}$$

If all pairs of avalanche variables $v_{\cdot j}^i$ and $v_{\cdot k}^i$ are independent, then ideally, $BIC_{Pearson}(f) = 0$. Therefore, in practice, when $BIC_{Pearson}(f) \approx 0$, it is concluded that f satisfies the BIC.

Table 1. SAC matrix $H^i = (v^i_{rj})$ of dimension $l \times m$ for the change of bit i over the set D of l inputs.

Avalanche Vectors	Avalanche Variables							
	$\boldsymbol{v^i_{\cdot 1}}$	$\boldsymbol{v^i_{\cdot 2}}$	$\cdots$	$\boldsymbol{v^i_{\cdot j}}$	$\cdots$	$\boldsymbol{v^i_{\cdot k}}$	$\cdots$	$\boldsymbol{v^i_{\cdot m}}$
V^i_1	v^i_{11}	v^i_{12}	$\cdots$	v^i_{1j}	$\cdots$	v^i_{1k}	$\cdots$	v^i_{1m}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
V^i_r	v^i_{r1}	v^i_{r2}	$\cdots$	v^i_{rj}	$\cdots$	v^i_{rk}	$\cdots$	v^i_{rm}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
V^i_l	v^i_{l1}	v^i_{l2}	$\cdots$	v^i_{lj}	$\cdots$	v^i_{lk}	$\cdots$	v^i_{lm}

2.1. Comparison between SAC and BIC

The SAC [13] verifies whether each output bit changes approximately half of the time by changing an input bit. Using the SAC matrix H^i, it is said that f satisfies the SAC if for all i and every avalanche variable $v^i_{\cdot j}$, with $1 \leq j \leq m$ and $1 \leq i \leq n$, $HW(v^i_{\cdot j})$ is binomial distributed with parameters $n = l$ and $p = \frac{1}{2}$, i.e., $v^i_{\cdot j} \sim B\left(l, \frac{1}{2}\right)$, where $HW(\cdot)$ is the Hamming weight. On the other hand, the BIC [13] measures the degree of independence between each pair $v^i_{\cdot j}$, $v^i_{\cdot k}$ of avalanche variables. Thus, the two criteria measure a different characteristic from the effect produced on the output bits changing an input bit; the SAC verifies uniformity in the distribution of each output bit, while the BIC measures the degree of independence between the output bits.

In [30], a new method to assess the correlation between statistical randomness tests based on mutual information was presented, using some test statistics and p-values of the tests. This tool can be used to determine the degree of correlation between these two statistical tests. In [29], an assessment of the independence between these two tests through absolute correlation coefficient is given, concluding that these tests are quite uncorrelated.

2.2. Stream Ciphers and RC4

The stream ciphers perform the encryption by converting plain text into bit-by-bit cipher-text through the use of a keystream and the XOR operation. A keystream is nothing more than a sequence of numbers generated in a pseudo-random way. This is achieved by building a pseudo-random number generator. The sequence of pseudo-random numbers used must meet certain statistical properties to be considered suitable for cryptographic use. In many applications (see [4,31]), ciphers of this type have become very important tools since they are very fast and their implementation is simpler than other ciphers, e.g., a block cipher. In these types of scenarios, the problem is in the transmission of a large amount of data in communication networks in a short time.

There are a wide variety of design proposals [32] to build pseudo-random number generators. Among these, the RC4 algorithm [23] stands out from others for its wide use in different applications and protocols. The RC4 stream cipher [23] is optimized to be used in 8-bit processors, being extremely fast and exceptionally simple. It was included in network protocols such as Secure Sockets Layer (SSL), Transport Layer Security (TLS), Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA), and in various applications used in Microsoft Windows, Lotus Notes, Apple Open Collaboration Environment (AOCE), and Oracle Secure SQL [23]. In the last decade, some applications [33,34] avoided RC4 encryption given some weaknesses found [35]. However, although it is not considered very secure [36], RC4 is still one of the most widely used stream ciphers [37], and continues to motivate research nowadays [36–38]. Furthermore, this cipher is a good option to measure the effectiveness of methods that analyze weaknesses in stream ciphers related to those already known in RC4 [22,24–26], or to check the performance of hardware or software schemes that make use of cryptography [39–41].

The RC4 has two main components: the key scheduling, and the pseudo-random number generator. The key scheduling generates an internal random permutation S of values from 0 to 255, from an initial permutation, a (random) key K of l-byte length, and two pointers i and j. The maximal key length is of $l = 256$ bytes (see Algorithm 1).

Algorithm 1 RC4 key-scheduling

for $i = 0 \rightarrow 255$ **do**
    $S[i] \leftarrow i$
end for
$j \leftarrow 0$
for $i = 0 \rightarrow 255$ **do**
    $j \leftarrow (j + S[i] + K[i \mod l]) \mod n$
    Swap $S[i]$ and $S[j]$
end for

The main part of the algorithm is the pseudo-random number generator that produces one-byte output in each step. As usual, for stream ciphers, the encryption will be an XOR of the pseudo-random sequence with the message (see Algorithm 2).

Algorithm 2 RC4 pseudo-random generator

$i \leftarrow 0$
$j \leftarrow 0$
while Generating Output **do**
    $i \leftarrow (i + 1) \mod 256$
    $j \leftarrow (j + S[i]) \mod 256$
    Swap $S[i]$ and $S[j]$
    Output $S[(S[i] + S[j]) \mod 256]$
end while

The weaknesses found can be classified according to the theme they exploit, some of which are:

1. Weak keys.
2. Key recovery from the state.
3. Key recovery from the key-stream.
4. State recovery attacks.
5. Biases and distinguishes.

While the fifth point is the most studied subject in the literature, the third point is the most serious attack made to RC4. The theme that is exploited in this paper has been deeply studied—in particular, Grosul and Wallach [24] demonstrated that certain related key-pairs generate similar output bytes in RC4. Later, Matsui [25] reported colliding key pairs for RC4 for the first time, and then stronger key collisions were found in [26]. For the RC4 stream cipher, several modifications have been proposed; while some modified only certain components or some operations, others completely changed the algorithm (see [42]). It is important to note that even RC4 variants have had a lot of attention in the scientific community (see [43]).

3. BIC Algorithm in Stream Ciphers

In this section, an algorithm is proposed to extend the bit independence criterion (BIC) to stream ciphers, experimentally confirming its effectiveness. The two main differences that arise in this scenario with respect to its application in S-boxes are discussed.

Let f be the function that will be evaluated by the BIC, $D = \{X_1, \ldots, X_l\}$ the set of l inputs X_r of n bits generated randomly and m the number of bits of the outputs of f, the proposed method consists of the following steps:

Step 1. Construct the n SAC H^i, $(i = 1, \ldots, n)$ matrices of dimension $l \times m$.

1. Evaluate $Y_r = f(X_r)$, $(r = 1, \ldots, l)$, and generate the output Y_r of size m.
2. Evaluate $Y_r^i = f(X_r \oplus e_i)$, and generate the output Y_r^i of size m, where e_i is the canonical vector.
3. Build the avalanche vector $V_r^i = Y_r \oplus Y_r^i = \{v_{r1}^i, \ldots, v_{rm}^i\}$ of m avalanche variables $v_{\cdot j}^i$, $(j = 1, \ldots, m)$.

Step 2. Evaluate the independence between the avalanche variables $v_{\cdot j}^i$ and $v_{\cdot k}^i$.

1. For each pair (j, k), with $1 \leq j, k \leq m$ and $j \neq k$, measure the independence between the avalanche variables $v_{\cdot j}^i$, $v_{\cdot k}^i$ by a test statistic.
2. Set a significance level α_1 and decide, using a statistical criterion, if the observed value of the test statistic allows to reject or not the hypothesis of independence between $v_{\cdot j}^i$ and $v_{\cdot k}^i$.
3. Count the number T^i of rejections between C_2^m pairs of the matrix H^i.

Step 3. Decision on whether or not to comply with the BIC criterion:

1. Count the total number T of rejections between the n matrices H^i.
2. Set a significance level α_2
3. Decide, using a statistical criterion, whether the observed value of T allows to reject the BIC compliance.

The following sections describe each of these steps and end with the proposal of an algorithm to evaluate the BIC in stream ciphers.

3.1. Building the SAC Matrix

First difference. When evaluating the BIC in S-boxes, it is possible to go through the entire space of $l = 2^n$ inputs since n usually takes small values; however, this is impractical in stream ciphers where the dimension of the input space can be 2^{128} or greater. To solve this problem, it is proposed to use the same approach applied in the randomness assessment to the outputs of pseudo-random generators through statistical tests [2]. This approach consists of generating a sample of l inputs with $l \ll 2^n$, and to determine the strength of the cipher from the results obtained from this sample.

The l inputs are chosen randomly in the space of 2^n possible inputs. This is the main difference; while the BIC test works over all of the input space with S-boxes, the stream cipher works with a randomly selected subset of the sample space.

3.2. Test of Independence between Two Avalanche Variables $v_{\cdot j}^i$ and $v_{\cdot k}^i$

Second difference. In [13], Pearson's correlation coefficient ρ was used to measure the degree of independence between the pairs of avalanche variables. The use of such a coefficient in [13,29] has two main disadvantages: the first one is that it only detects linear correlations, and the second one is that the critical region for the rejection of the null hypothesis is not explicitly defined, i.e., a threshold is not defined below which $BIC_{Pearson}(f) \approx 0$ is decided. Thus, it can be a reason for an imprecision in the decision when dealing with small coefficient values. In order to solve the first aforementioned disadvantage, mutual information can be applied to measure the degree of independence between pairs of avalanche variables [44], but in this case, it is important to determine which estimator to use, since there are no estimators of unbiased entropy of minimal variance; the second disadvantage can be solved by defining the critical region using a transformation of the correlation coefficient of the type $t = \sqrt{(N-2)\rho^2/(1-\rho^2)}$, where t is distributed as a t-Student distribution with $N-2$ degrees of freedom [45].

Another approach is that when $v^i_{\cdot j}$ and $v^i_{\cdot k}$ are independent, then $s^i_{jk} = v^i_{\cdot j} \oplus v^i_{\cdot k}$ is balanced [46]. In this work, independence will be evaluated by measuring the adjustment $HW(s^i_{jk})$ to the binomial distribution $B(l, 1/2)$, where $HW(\cdot)$ is the Hamming weight. This allows setting a threshold for the decision criterion on independence between $v^i_{\cdot j}$ and $v^i_{\cdot k}$.

Since H^i is a binary matrix, the adjustment to the binomial distribution will be measured by the χ^2-test with 1 degree of freedom, with the test hypothesis given by:

$$H_0 : v^i_{\cdot j} \text{ and } v^i_{\cdot k} \text{ independent},$$
$$H_1 : v^i_{\cdot j} \text{ and } v^i_{\cdot k} \text{ dependent}.$$

That is,

$$H_0 : HW(s^i_{jk}) \sim B\left(l, \frac{1}{2}\right),$$
$$H_1 : HW(s^i_{jk}) \not\sim B\left(l, \frac{1}{2}\right).$$

The test statistic used is

$$\chi^2_{s^i_{jk}} = \frac{\left(HW(s^i_{jk}) - \frac{l}{2}\right)^2}{\frac{l}{4}}. \tag{2}$$

As usual [2], the value α_1 is such that

$$P\left(\chi^2_{s^i_{jk}} \leq \chi^2_{\alpha_1,1}\right) = 1 - \alpha_1. \tag{3}$$

If $\chi^2_{s^i_{jk}} > \chi^2_{\alpha_1,1}$ the null hypothesis H_0 is rejected.

It is left for future works, to compare the effectiveness of these three criteria for evaluating independence between the avalanche variables.

3.3. BIC Acceptance Test

To decide whether the stream cipher f satisfies the BIC, it is necessary to take into account the number of rejections of H_0 on the n matrices; for this, a random variable T, which counts the total number of rejections on n matrices is defined:

$$T = T(n, m, \alpha_1) = \sum_{i=1}^{n} T^i(m, \alpha_1), \tag{4}$$

where

$$T^i(m, \alpha_1) = T^i = \sum_{j=1}^{m-1} \sum_{k>j}^{m} t\left(v^i_{\cdot j}, v^i_{\cdot k}, \alpha_1\right), \tag{5}$$

and

$$t(v^i_{\cdot j}, v^i_{\cdot k}, \alpha_1) = \begin{cases} 1 & \text{If } H_0 \text{ is rejected for } v^i_{\cdot j} \text{ and } v^i_{\cdot k} \\ & \text{with significance } \alpha_1 \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The variable T^i counts the number of rejections of the null hypothesis H_0 in the matrix H^i.

Expected number of rejections of H_0. In each of the n SAC H^i matrices, C^m_2 pairs of columns are formed, thus the number of rejections T satisfies

$$0 \leq T \leq n \cdot C^m_2. \tag{7}$$

When $T = 0$, we have the ideal case for compliance with the BIC, since all the pairs of columns are independent, while as $T \gg 0$, the number of non-independent column pairs increases.

Under the hypothesis test above, with a significance level α_1, the expected number of rejections of H_0 is:

$$E\left(T^i |\ H_0\right) = \left(\alpha_1 \cdot C_2^m\right), \tag{8}$$

for each matrix H^i. In total, among the n matrices SAC are expected

$$E\left(T |\ H_0\right) = n \cdot \left(\alpha_1 \cdot C_2^m\right), \tag{9}$$

H_0 rejections.

The random variable

$$T = \sum_{i=1}^{n} \sum_{j=1}^{m-1} \sum_{k>j}^{m} t\left(v^i_{\cdot j},\ v^i_{\cdot k}, \alpha_1\right), \tag{10}$$

follows a binomial distribution $B(n \cdot C_2^m, \alpha_1)$. Taking into account that generally $\alpha_1 < 0.1$, this distribution can be approximated, in this case, to the Poisson distribution with parameter $\lambda = (\alpha_1 \cdot n \cdot C_2^m)$. Since λ is large, due to large values of $n \cdot C_2^m$, then the Poisson distribution can be approximated by the Normal distribution with mean and variance:

$$E(T|H_0) = \alpha_1 \cdot n \cdot C_2^m,\ \sigma^2(T|H_0) = \alpha_1 \cdot n \cdot C_2^m \cdot (1 - \alpha_1). \tag{11}$$

Thus

$$Z_T = \frac{T - E(T|H_0)}{\sqrt{\sigma^2(T|H_0)}} \sim N(0,1). \tag{12}$$

Decision criteria. To compare the Z_T value with the $N(0,1)$ distribution, a significance level α_2 is selected. Then, it is tested if f does not satisfy the BIC, with a significance level α_2, if $Z_T > Z_{1-\alpha_2}$. It can be seen that if $0 \leq T \leq E(T|H_0)$, then the values of Z_T decreases with respect to $Z_{1-\alpha_2}$ and $Z_T > Z_{1-\alpha_2}$ is not satisfied, so the BIC is fulfilled. On the other hand, if $T \gg E(T|H_0)$, then the values of Z_T will be greater as T increases, so $Z_T > Z_{1-\alpha_2}$ is satisfied and the BIC compliance is rejected.

Normality of the test statistic T. In the expression of T there are $n \cdot C_2^m$ Bernoulli variables $t(v^i_{\cdot j}, v^i_{\cdot k}, \alpha_1)$, whose distributions under H_0 and H_1 are different:

Under H_0, all variables $t(v^i_{\cdot j}, v^i_{\cdot k}, \alpha_1)$ are independent, identically distributed and take the value of 1 with probability $p^i_{jk} = P(t(v^i_{\cdot j}, v^i_{\cdot k}, \alpha_1) = 1) = \alpha_1$, so T follows exactly a binomial distribution $B\left(n \cdot C_2^m,\ \alpha_1\right)$. Although generally $\alpha_1 \leq 0.1$ the binomial distribution $B\left(n \cdot C_2^m,\ \alpha_1\right)$ can be approximated by the normal distribution, with mean $E\left(T|H_0\right) = \alpha_1 \cdot n \cdot C_2^m$ and variance $\sigma^2\left(T|H_0\right) = \alpha_1 \cdot n \cdot C_2^m\ (1 - \alpha_1)$, taking into account that $n \cdot C_2^m$ grows very quickly with m.

Under H_1, the variables $t(v^i_{\cdot j}, v^i_{\cdot k}, \alpha_1)$ that appear in the expression of T are not identically distributed, since the rejection of the BIC means that there are several matrices H^i for which the hypothesis H_0 of independence between $v^i_{\cdot j}$ and $v^i_{\cdot k}$ is rejected. In this case, $p^i_{jk} \neq \alpha_1$ and may be different when i, j, k varies. For this reason, a binomial does not appear directly as the distribution of T. However, it is still possible to approximate the distribution of T by the Normal distribution. For this it is sufficient to calculate the mean

$$P_{n \cdot C_2^m} = \frac{\sum_i^n \sum_j^{m-1} \sum_{k>j}^m p^i_{jk}}{n \cdot C_2^m}, \tag{13}$$

between the probabilities of all the variables $t(v^i_{\cdot j}, v^i_{\cdot k}, \alpha_1)$ and the distribution of T can be approximated by the binomial distribution $B(n \cdot C_2^m, P_{n \cdot C_2^m})$. This distribution, in turn, can be approximated by the Normal distribution, taking into account high values of $n \cdot C_2^m$. The precision of this approximation

depends on the difference between the probabilities p^i_{jk} involved in $P_{n\cdot C_2^m}$, therefore the variance value between these probabilities can be a measure of the quality of the approximation.

When comparing the distribution of T under H_0 and H_1, similarities and differences are observed. They are similar in that in both cases T follows a Normal distribution, but there are two differences, the first and most important is observed between the expected values of both distributions (it will be higher under H_1) and the second refers to the level of adjustment to this distribution (may be lower under H_1). In the rest of this work, the proposed method to evaluate the BIC in stream ciphers will be called the BIC test.

3.4. BIC Test Algorithm

Given a set $D = \{X_1, \ldots, X_l\}$ of l randomly chosen n bits inputs to the function f, constructs for each binary vector e_i $(1 \leq i \leq n)$ its associated SAC matrix H^i and for all for j, k with $j \neq k$, it is checked if $HW(s^i_{jk})$ follow the $B\left(l, \frac{1}{2}\right)$ distribution, see the proposed Algorithm 3.

Algorithm 3 BIC stream ciphers algorithm

Input: f function to evaluate, n size of the inputs of f, m size of the outputs of f, α_1 and α_2 levels of significance, D set of l inputs to the function f.
Output: If f satisfies the BIC

1: $T = 0$
2: **for** $i = 1 \rightarrow n$ **do**
3: **for** $r = 1 \rightarrow l$ **do** ▷ Matrix Construction H_i
4: Compute $V^i_r = Y_r \oplus Y^i_r$
5: **end for**
6: **for each** (j, k) **do** ▷ Independence check between $v^i_{\cdot j}$ and $v^i_{\cdot k}$
7: **if** ${\chi^2}_{s^i_{jk}} > {\chi^2}_{\alpha_1,1}$ **then**
8: $T = T + 1$ ▷ Independence is rejected between $v^i_{\cdot j}$ and $v^i_{\cdot k}$
9: **end if**
10: **end for**
11: **end for**
12: **if** $Z_T > Z_{1-\alpha_2}$ **then** f does not satisfy the BIC
13: **else** f satisfies the BIC
14: **end if**

3.4.1. Complexity of the Algorithm

In steps 3–5 of the algorithm, f is used to generate m output bits. Assuming that the stream cipher f generates each output with a constant cost, then $O(lm)$ operations are performed in these steps, since l times m output bits are generated from f. In steps 6–10 of the algorithm, $O(m^2 l)$ operations are performed due to the computation C_2^m times the Hamming weight in a sequence of l bits.

Thus the algorithm performs $O\left(n \max(l\ m,\ l\ m^2)\right) = O\left(n\ l\ m^2\right)$ operations, and the number of algorithm operations depends on the number n of input bits, the number m of output bits, and the number l of inputs used. It can be seen that the increase in the parameter m has a greater influence than n and l in increasing the number of operations of the algorithm. In the particular case $m = n = l$, $O(m^4)$ operations are performed.

3.4.2. Parameter Selection

As seen in the previous section, the number of operations of the BIC algorithm depends on three parameters, the number l of inputs, the number n of bits of each input, and the number m of bits of each output.

Selection of l such that $\widehat{p} \approx 0.5$ and $HW(s^i_{jk})$ fit to the binomial distribution $B(n, 1/2)$. The number l of entries influences the effectiveness of the χ^2-test in determining whether two columns are independent. Increasing l guarantees a greater fit of $HW(s^i_{jk})$ to the binomial distribution $B(n, 1/2)$; however,

it causes an increase in the number of operations. In practice, the idea is to obtain a cost-effectiveness ratio using a value of l such that it maintains the fit and provides a practical number of operations. Using the confidence interval for proportions [47], it is possible to obtain a value of l_0, such that prefixing $l > l_0$ achieves a good fit. This confidence interval is given by

$$P\left(-Z_{\alpha_1/2} < \frac{\widehat{p} - p}{\sqrt{\frac{pq}{l}}} < Z_{\alpha_1/2}\right) = 1 - \alpha_1. \tag{14}$$

Solving for l we get to

$$l > l_0 = \frac{Z^2_{\alpha_1/2}\, pq}{e^2}, \tag{15}$$

where $e = \widehat{p} - p$, is the deviation of $\widehat{p}$ over p, and $q = (1 - p)$.

Example 1. ***Calculation of the lower bound*** l_0 ***for*** l. *A value* l_0 *from which, with high probability, it is satisfied that* $\widehat{q} \approx \widehat{p} \approx 0.5$ *is needed. Then, substituting for a significance level* $\alpha_1 = 0.01$ *and a deviation* e *whose absolute value* $|e|$ *satisfy inequality* $|e| = |\widehat{p} - 0.5| \leq 0.03$, *we get*

$$l_0 = \frac{Z^2_{0.005} \cdot 0.25}{0.03^2} \approx 2189.$$

In this way, for the significance level α_1 *and the deviation* e *selected, it is concluded that* l *must be chosen such that* $l > l_0 = 2189$.

Example 2. ***Convergence of*** $\widehat{p}$ ***and deviation*** $\widehat{e}$. *Table 2 shows the behavior of the deviation* $\widehat{e}$ *observed for several* l, $l > l_0 = 2189$, *with* $n = 64$ *and* $m = 32$. *It can be seen how, for most of the estimated* e, *the imposed condition is met* $|\widehat{e}| \leq 0.03$.

Table 2. Values of the deviation $|\widehat{e}|$ for several l, $l > l_0 = 2189$ with $n = 64$ and $m = 32$.

l	Mean Value $\widehat{p}$	$\lvert\widehat{e}\rvert$
4096	0.5	0.05
8192	0.5	0.03
16,384	0.5	0.03
32,768	0.5	0.02

Selection of n, m *under the null hypothesis* H_0. The number n of inputs and the number m of outputs influence the sample size for the calculation of the number T of rejections of H_0. In general, we will have $d = n \cdot C_2^m$ pairs of columns to check and it is expected, with probability α_1, that $\lambda = \alpha_1 \cdot d$ pairs of columns will be rejected.

Let $\lambda_0 = \alpha_1 \cdot d_0$ be some default value of λ from which the distribution of T can be approximated to $N(0, 1)$. It is necessary to select n and m such that $d > d_0$ is satisfied and a value of λ such that $\lambda > \lambda_0$ is obtained. It is advisable to select a high value of λ_0 that avoids the use of corrections and provides a good fit.

It is known that increasing λ_0 provides better precision in the Poisson approximation to the Normal distribution. To obtain d_0, we can use the confidence interval for proportions [47], this time in an approximation to the Normal distribution with one tail. So, we have

$$P\left(\frac{\widehat{p} - p}{\sqrt{\frac{pq}{d}}} < Z_{\alpha_2}\right) = 1 - \alpha_2. \tag{16}$$

Solving for d we get to

$$d > d_0 = \frac{Z^2_{\alpha_2} pq}{e^2}. \tag{17}$$

Example 3. ***Calculation of the lower bound*** d_0 ***for*** d. *Substituting,* $p = 0.01$, $q = 0.99$, *with a significance level* $\alpha_2 = 0.001$ *and a deviation* $|e|$ *of* 0.003, *we obtain*

$$d_0 = \frac{Z^2_{0.001} \cdot 0.25}{0.003^2} \approx 10503.$$

Then, $\lambda_0 = \alpha_1 \cdot d_0 \approx 0.01 \cdot (10503) \approx 105$, *therefore, for the values of* α_1 *and e chosen, it is enough to select values of n and m such that* $\lambda > \lambda_0 \approx 105$. *In Table 3, for* $\alpha_1 = 0.01$, *some values of n and m are highlighted in italics from which* $\lambda > \lambda_0 = 105$.

Table 3. λ values for multiple values of n and m with $\alpha_1 = 0.01$. Values of n and m are highlighted in italics from which $\lambda > \lambda_0 = 105$.

n	***m***			
	8	**16**	**32**	**64**
8	2.24	9.6	39.68	*161.28*
16	4.48	19.2	79.36	*322.56*
32	8.96	38.4	*158.72*	*645.12*
64	17.92	76.8	*317.44*	*1290.24*
128	35.84	*153.6*	*634.88*	*2580.48*
256	71.68	*307.2*	*1269.76*	*5160.96*
512	*143.36*	*614.4*	*2539.52*	*10,321.92*

To select n, m and l, the trade-off between reducing computational cost and maximizing effectiveness can be taken into account. However, it is very important to be careful when selecting which values to use, since minimizing computational cost could limit the effectiveness of the BIC method and overestimate the quality of the stream cipher. It is advised to prioritize increasing effectiveness.

4. Experiments and Discussion of the Results

In this section, experiments are carried out in two different scenarios. In the first scenario, the behavior of the Z_T test statistic is investigated under the hypothesis H_0 of compliance with the BIC test, evaluating the test on random H^i matrices. The second scenario shows the behavior of the Z_T test statistic when evaluating it in a stream cipher that does not meet this criterion.

4.1. Scenario 1 (BIC in Random SAC Matrices)

It is expected that under H_0, we obtain $E(Z_T|H_0) = 0$, $\sigma^2(T|H_0) = 1$ and $Z_T \sim N(0,1)$. The experiments in this scenario were carried out under uniform and independent randomly generated SAC matrices, to evaluate compliance, under H_0, of the $N(0,1)$ distribution of Z_T.

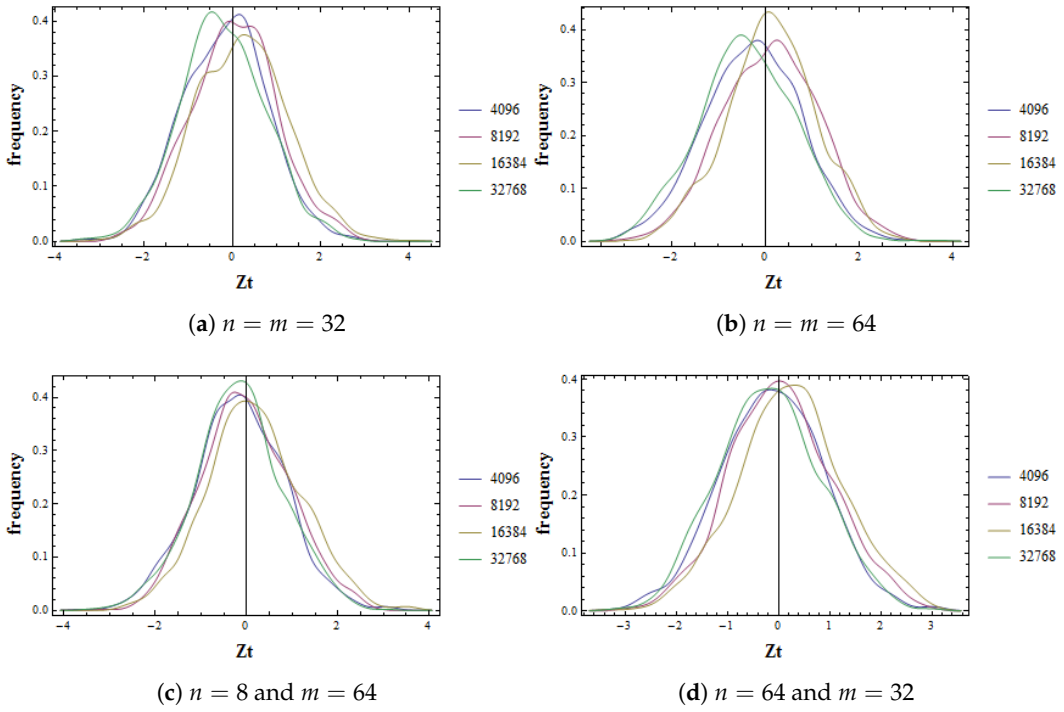
Taking into account Table 3, four sets of parameters were selected, two for $n = m$ and two for $n \neq m$:

- $n = m$:($n = m = 32$) and ($n = m = 64$)
- $n \neq m$: $(n = 64, m = 32)$ and $(n = 8, m = 64)$.

The values $l \in \{4096, 8192, 16{,}384, 32{,}678\}$ will be varied, in order to verify the influence of the variation of the parameters n, m and l in the adjustment of Z_T. The values of n and m with the lowest computational cost were selected, that is, the values of n and m that provide the lowest values of λ such that $\lambda > \lambda_0 = 105$.

The values n and m will be used as a power of two, since current ciphers work with inputs and outputs whose size has these characteristics and also l to speed up, in terms of execution time, the computation of the BIC method. However, it is important to note that the BIC method can be used for any value of n, m and l, as long as the requirements outlined in the previous section are met.

Normality of Z_T in H^i random matrices. Figure 1 corresponds to the observed distribution of 1000 values of Z_T, for each pair of parameters n and m, and each value of l.



(**a**) $n = m = 32$ (**b**) $n = m = 64$

(**c**) $n = 8$ and $m = 64$ (**d**) $n = 64$ and $m = 32$

Figure 1. Observed distribution of 1000 values of Z_T in random H^i matrices for various values of n, m, and l.

Tables 4 and 5 show the values $\widehat{E}(Z_T|H_0)$ and $\widehat{\sigma^2}(Z_T|H_0)$ respectively observed in each sample, for each value of n, m and l.

Table 4. Observed $\widehat{E}(Z_T|H_0)$ values for each selected n, m, l value.

(n, m)	l			
	4096	**8192**	**16,384**	**32,768**
(32, 32)	−0.150216	0.044674	0.214355	−0.210047
(64, 64)	−0.268244	0.110717	0.137298	−0.383549
(8, 64)	−0.154163	−0.0239	0.173869	−0.164926
(64, 32)	−0.118008	0.05765	0.236807	−0.175659

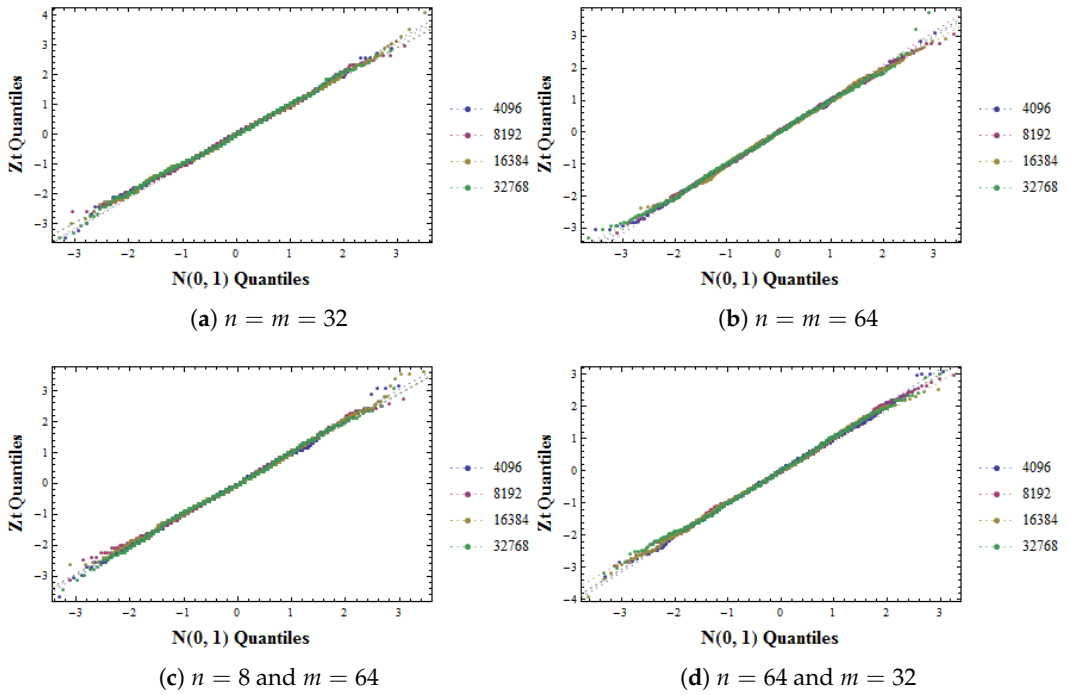
The analysis of Figure 1 and Tables 4 and 5, suggests the fulfillment of the hypothesis H_0 about the distribution of $Z_T \sim N(0, 1)$, for all the values of the parameters l, n, m selected. As can be seen in Tables 4 and 5, by varying l, n, m, the values $\widehat{E}(Z_T|H_0)$ and $\widehat{\sigma^2}(Z_T|H_0)$ of the observed distribution of

Z_T maintain the fit to the parameters $\mu = 0$ and $\sigma^2 = 1$ expected in a distribution $N(0,1)$. Figure 1 shows the bell shape and approximate symmetry of the obtained distributions.

Table 5. Observed $\widehat{\sigma^2}(Z_T|H_0)$ values for each selected n, m, l value.

(n,m)	l			
	4096	**8192**	**16,384**	**32,768**
(32, 32)	0.906793	0.938818	1.06287	0.930434
(64, 64)	1.04064	1.02569	0.97230	1.04773
(8, 64)	0.972279	0.939652	1.05472	0.923853
(64, 32)	0.994157	0.997373	1.06138	0.983795

Normality Test. The Shapiro–Wilks [48] test for normality was applied to all selected parameter sets. The results are shown in Figure 2 and Table 6.



(**a**) $n = m = 32$ (**b**) $n = m = 64$

(**c**) $n = 8$ and $m = 64$ (**d**) $n = 64$ and $m = 32$

Figure 2. Adjustment of the observed distribution from Z_T to $N(0,1)$ in H^i random matrices, which satisfy the BIC.

In Figure 2 we can see how the observed distribution of Z_t for all the values of l, n, m, fit the distribution $N(0,1)$. Table 6 shows the p-values corresponding to the Shapiro-Wilk normality test for each of the chosen parameter sets.

Table 6. p-values of the Shapiro-Wilk test of normality for samples of Z_t, in random H^i matrices, that satisfy the BIC.

l	$n = m = 32$	$n = m = 64$	$n = 8$ and $m = 64$	$n = 64$ and $m = 32$
4096	0.252382	0.724504	0.262997	0.482318
8192	0.127693	0.573267	0.161048	0.326505
16,384	0.296125	0.653315	0.141577	0.524475
32,768	0.309173	0.37739	0.210961	0.237133

It is observed that in all cases, the p-values are greater than the usual values assumed for α, such as 0.01 or 0.05 and are consistent with the assumed normality hypothesis. The higher the value of $n = m$, the higher the p-value, which corresponds to the influence of these parameters on the value of λ (see Table 3).

BIC test application on H^i random matrices. To evaluate the behavior of the BIC test in random matrices, each Z_t was compared with the critical value $Z_{1-\alpha_2}$, and the number of rejections of H_0 was counted. Tables 7 and 8 show the results for various levels of significance α_2 and $l = 16{,}384$. The observed number of rejections is expected to correspond to that expected according to the selected α_2 level, which would allow choosing α_2, to obtain zero rejections in this scenario.

Table 7. Expected $E(\#[Z_T > Z_{1-\alpha_2}|H_0]\ )$ and observed $\#[Z_T > Z_{1-\alpha_2}|H_0]$ number of rejections in samples of 1000 values of Z_t for $n = m$, in H^i random matrices.

α_2	$E(\#[Z_T > Z_{1-\alpha_2}\|H_0])$	$\#[Z_T > Z_{1-\alpha_2}\|H_0]$	
		$n = m = 32$	$n = m = 64$
0.05	50	31	43
0.01	10	8	9
0.001	1	1	1
0.0001	0	0	0

Table 8. Expected $E(\#[Z_T > Z_{1-\alpha_2}\ |H_0])$ and observed $\#[Z_T > Z_{1-\alpha_2}|H_0]$ number of rejections in samples of 1000 values of Z_t for $n \neq m$, in H^i random matrices.

α_2	$E(\#[Z_T > Z_{1-\alpha_2}\ \|H_0])$	$\#[Z_T > Z_{1-\alpha_2}\|H_0]$	
		$n = 8$ and $m = 64$	$n = 64$ and $m = 32$
0.05	50	36	42
0.01	10	7	5
0.001	1	0	1
0.0001	0	0	0

For the value of $\alpha_2 = 0.0001$ located in the last row of both tables, no statistical dependence is detected as expected in random matrices, confirming the effectiveness of the criterion and illustrating the importance of the proper selection of α_2 , according to the number $d = n \cdot C_2^m$ of pairs of columns whose independence is evaluated. For the values of $l, n, m, \alpha_1, \alpha_2$ used, such that no Type I error is made, the probability of making a Type II error must be calculated and the values that minimize it must be chosen. In this sense, experiments will be carried out in the second scenario on a stream cipher.

4.2. Scenario 2 (BIC in Stream Cipher)

For this scenario, it is convenient to apply the test to a stream cipher that violates the BIC. RC4 was chosen because there are reports of the existence of dependencies between the inputs and outputs in this cipher [22–25]. Experiments were performed setting the parameters $n = m \in \{32, 64, 128, 160, 256\}$ and 1000 sets D of $l = 16{,}384$ entries each were built. In each set, Z_T was calculated and compared with the critical value $Z_{1-\alpha_2}$, varying α_2. Figure 3 shows the distribution of the 1000 values of Z_T obtained. Table 9 show the values $\widehat{E}(Z_T)$ and $\widehat{\sigma^2}(Z_T)$ observed in each sample, for each value of n, m, and l.

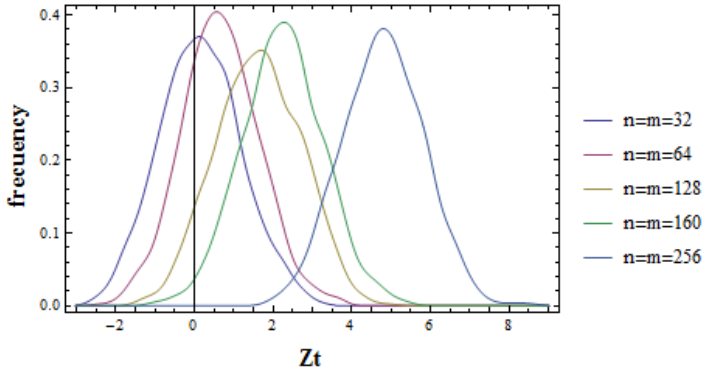


Figure 3. Distribution of the sample of 1000 values of Z_T for SAC matrices generated with RC4 with $n = m \in \{32, 64, 128, 160, 256\}$.

Table 9. Expected value $\widehat{E}(Z_T)$ and variance $\widehat{\sigma^2}(Z_T)$ of Z_T for SAC matrices generated with the RC4.

(n, m)	$\widehat{E}(Z_T)$	$\widehat{\sigma^2}(Z_T)$
(32, 32)	0.149419	1.06442
(64, 64)	0.661726	0.967951
(128, 128)	1.62968	1.15061
(160, 160)	2.24493	1.07417
(256, 256)	4.79748	1.06715

To verify the normality of the data, the Shapiro–Wilks [48] normality test was applied to all the selected parameter sets. The results are shown in Figure 4 and Table 10.

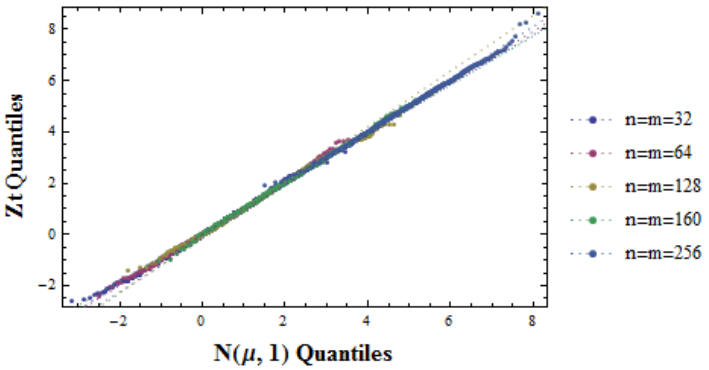


Figure 4. Normality test of the sample of 1000 values of Z_T for SAC matrices generated with the RC4 with $n = m \in \{32, 64, 128, 160, 256\}$.

Table 10. p-values of the Shapiro-Wilk test of normality on samples of Z_t for SAC matrices generated with the RC4 with $n = m \in \{32, 64, 128, 160, 256\}$.

(n, m)	p-Values
(32, 32)	0.103538
(64, 64)	0.582878
(128, 128)	0.382171
(160, 160)	0.943337
(256, 256)	0.673625

In Figure 4 we can see how by increasing the values of $m = n$ the Normal distribution $N(\mu, 1)$ of the statistician Z_t is maintained, however, the value of μ increases (see Figure 3 and Table 9).

It is observed that in all cases the p-values are greater than the usual values assumed for α, such as 0.01 or 0.05 and the samples maintain normality.

In Table 11 it is noted how in RC4 the effectiveness of the criterion increases as the values of n and m increase. That is, increasing the values $m = n$ increases the number of correct decisions to reject H_0. As mentioned, it is known that by increasing the value of n in RC4 the probability of finding very similar outputs, or even the same, increases for inputs that differ by a few bits [22,24–26].

Table 11. Expected $E(\#[Z_T > Z_{1-\alpha_2} \mid H_0])$ and observed $\#[Z_T > Z_{1-\alpha_2}]$ number of rejections in 1000 repetitions of the BIC test in SAC matrices generated with the RC4. All cases in which the observed number of rejections exceeds the expected value are indicated in italics.

α_2	$E(\#[Z_T > Z_{1-\alpha_2} \mid H_0])$	$\#[Z_T > Z_{1-\alpha_2}]$				
		$n = m$				
		32	**64**	**128**	**160**	**256**
0.05	50	*77*	*155*	*497*	*731*	*1000*
0.01	10	*22*	*44*	*272*	*462*	*993*
0.001	1	*2*	*11*	*91*	*215*	*953*
0.0001	0	0	0	*19*	*74*	*843*

This experiment confirms the effectiveness of the BIC test by detecting dependence between the inputs-outputs of RC4 and allows us to conclude that in RC4, the effectiveness is an increasing function of the value of the parameters $n = m$. All cases in which the observed number of rejections exceeds the expected value are indicated in italics.

An important feature in statistical tests is the determination of type I and type II errors [2]. Under H_0, we have that $v^i_{\cdot j}$ and $v^i_{\cdot k}$ are independent, then the type I error consists in rejecting independence when they are and therefore deciding that the cipher has a weakness when it does not have it. Meanwhile, not rejecting H_0 when there is a dependency means that it would be decided that the cipher passes the BIC, when in fact it does not pass it, and a type II error would be committed. Table 12 shows the proportion of Type I and II errors, committed by the BIC test, for some parameter sets.

Table 12. Proportion of type I and II errors made by the BIC test.

α_2	Estimation Type I Error		Estimation Type II Error	
	$n = 32, m = 32$	$n = 64, m = 64$	$n = 32, m = 32$	$n = 64, m = 64$
0.05	0.031	0.043	0.077	0.155
0.01	0.008	0.009	0.022	0.044
0.001	0.001	0.001	0.002	0.011
0.0001	0	0	0	0

It can be seen that for $\alpha_2 = 0.0001$ type I and II errors are not made.

The outputs of RC4 [23] are known to pass numerous statistical tests [49], however they do not satisfy the BIC statistical test proposed in this work. This shows that the BIC statistical test complements the classic randomness tests, therefore it constitutes a tool to consider to evaluate stream ciphers.

5. Conclusions

An algorithm was proposed to extend the application of the Bit Independence Criterion (BIC) to stream ciphers. This algorithm detects the existence of statistical dependence between the inputs and outputs of a stream cipher. The effectiveness of the BIC test was experimentally confirmed when applied on random matrices, in which it does not detect dependence, and on the RC4 cipher, detecting statistical dependencies between the inputs and outputs of this cipher that are related with previously reported.

The algorithm depends on the number n of bits of the inputs, the number m of bits of the outputs, and the number l of inputs used. These parameters determine its complexity. The results achieved confirm the importance of varying the n and m parameters to apply the BIC criteria in the evaluation of stream ciphers. For RC4 the effectiveness of the criterion is a growing function of the n and m parameters.

It is recommended to guarantee the effectiveness of the proposed BIC test by selecting the values of the parameters greater than the minimum value estimated in the article. From that minimum, increase the values depending on the available computing power, estimating the time using the complexity expressions that were presented from the algorithm.

The BIC statistical test complements the classical statistical tests of randomness as it allows expanding the evaluation of the stream ciphers, by measuring the degree of independence present between the input of the cipher and its outputs, thus measuring other statistical characteristics that are not only the evaluation of randomness of their output sequences.

In future work, it is planned to apply this test to other stream ciphers, investigate the optimal choice of the m and n parameters and compare the effectiveness of the criterion taking into account the mutual information, the Pearson's coefficient, with the transformation mentioned, and the criteria applied in this work. The behavior of the proposal will be experimentally verified when the sample size increases. It will be investigated in an implementation variant using parallelism for Algorithm 3.

Author Contributions: Conceptualization, E.J.M.-C., G.S.-G. and C.M.L.-P.; methodology, E.J.M.-C., G.S.-G. and C.M.L.-P.; software, E.J.M.-C., G.S.-G. and C.M.L.-P.; validation, E.J.M.-C., R.S.-L. and O.R.; formal analysis, E.J.M.-C., G.S.-G., O.R., R.S.-L. and C.M.L.-P.; investigation, E.J.M.-C., G.S.-G., O.R., R.S.-L. and C.M.L.-P.; writing—original draft preparation, E.J.M.-C., G.S.-G., O.R., R.S.-L. and C.M.L.-P.; writing—review and editing, E.J.M.-C., G.S.-G., O.R., R.S.-L. and C.M.L.-P.; supervision, E.J.M.-C., G.S.-G., O.R., R.S.-L. and C.M.L.-P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Notation table.

$f : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^m$	Describe the function that transforms n input bits into m output bits
$X = (x_1, \ldots, x_n)$	n-bit input binary vector
$Y = f(X) = (y_1, \ldots, y_m)$	m-bit output binary vector
e_i	Unit vector with 1 in the i-th component with $1 \leq i \leq n$
X^i	Vector resulting from the operation $X^i = X \oplus e_i$ for input X
Y^i	m-bit output binary vector corresponding to input X^i, $Y^i = f(X^i)$
$V^i = Y \oplus Y^i = (v_1^i, v_2^i, \ldots, v_m^i)$	Avalanche vector associated with vector e_i and input X
$v_j^i \in \mathbb{F}_2$	Avalanche variable associated to vector e_i and input X with $1 \leq j \leq m$
$D = \{X_1, \ldots, X_l\}$	Set of l inputs X_r, with $1 \leq r \leq l$
X_r^i	Vector resulting from the operation $X_r^i = X_r \oplus e_i$ for the input X_r

Table A1. *Cont.*

Y_r^i	Binary output vector of m bits corresponding to input X_r^i, $Y^i = f(X_r^i)$
$V_r^i = Y_r \oplus Y_r^i = (v_{r1}^i, v_{r2}^i, \ldots, v_{rm}^i)$	Avalanche vector associated with vector e_i and input X_r
$v_{rj}^i \in \mathbb{F}_2$	Avalanche variable associated to vector e_i and input X_r with $1 \leq j \leq m$

References

1. Marton, K.; Suciu, A.; Ignat, I. Randomness in digital cryptography: A survey. *Rom. J. Inf. Sci. Technol.* **2010**, *13*, 219–240.
2. Demirhan, H.; Bitirim, N. Statistical Testing of Cryptographic Randomness. *J. Stat. Stat. Actuar. Sci.* **2016**, *9*, 1–11.
3. ECRYPT Stream Cipher Project. 2011. Available online: http://cr.yp.to/streamciphers.html (accessed on 5 July 2020) [CrossRef]
4. Yerukala, N.; Kamakshi Prasad, V.; Apparao, A. Performance and statistical analysis of stream ciphers in GSM communications. *J. Commun. Softw. Syst.* **2020**, *16*, 11–18. [CrossRef]
5. Gorbenko, I.; Kuznetsov, A.; Lutsenko, M.; Ivanenko, D. The research of modern stream ciphers. In Proceedings of the 2017 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov, Ukraine, 10–13 October 2017; pp. 207–210. [CrossRef]
6. Upadhya, D.; Gandhi, S. Randomness evaluation of ZUC, SNOW and GRAIN stream ciphers. *Adv. Intell. Syst. Comput.* **2017**, *508*, 55–63. [CrossRef]
7. Rukhin, A.; Soto, J.; Nechvatal, J. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; Technical Report April; Booz-Allen and Hamilton Inc.: Mclean, VA, USA, 2010.
8. Marsaglia, G. The Marsaglia Random Number CDROM Including the Diehard Battery of Tests of Randomness. Florida State University, 1995. Available online: http://stat.fsu.edu/pub/diehard/ (accessed on 5 July 2020).
9. L'ecuyer, P.; Simard, R. TestU01: A C library for empirical testing of random number generators. *ACM Trans. Math. Softw. TOMS* **2007**, *33*. [CrossRef]
10. McClellan, M.T.; Minker, J.; Knuth, D.E. *The Art of Computer Programming, Vol. 3: Sorting and Searching*; Addison-Wesley Professional: Boston, MA, USA, 1974; Volume 28, p. 1175. [CrossRef]
11. Shi, Z.; Zhang, B.; Feng, D.; Wu, W. Improved key recovery attacks on reduced-round Salsa20 and ChaCha. *Lect. Notes Comput. Sci.* **2013**, *7839 LNCS*, 337–351. [CrossRef]
12. Maitra, S.; Paul, G. New form of permutation bias and secret key leakage in keystream bytes of RC4. In *International Workshop on Fast Software Encryption*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5086 LNCS, pp. 253–269. [CrossRef]
13. Hancock, P.A. On the Design of Time. *Ergon. Des.* **2018**, *26*, 4–9. [CrossRef]
14. Qureshi, A.; Shah, T. S-box on subgroup of Galois field based on linear fractional transformation. *Electron. Lett.* **2017**, *53*, 604–606. [CrossRef]
15. Naseer, Y.; Shah, T.; Shah, D.; Hussain, S. A Novel Algorithm of Constructing Highly Nonlinear S-p-boxes. *Cryptography* **2019**, *3*, 6. [CrossRef]
16. Turan, M.S. On Statistical Analysis of Synchronous Stream Ciphers. *arXiv* **2008**, arXiv:1011.1669v3.
17. Duta, C.L.; Mocanu, B.C.; Vladescu, F.A.; Gheorghe, L. Randomness Evaluation Framework of Cryptographic Algorithms. *Int. J. Cryptogr. Inf. Secur.* **2014**, *4*, 31–49. [CrossRef]
18. Castro, J.C.H.; Sierra, J.M.; Seznec, A.; Izquierdo, A.; Ribagorda, A. The strict avalanche criterion randomness test. *Math. Comput. Simul.* **2005**, *68*, 1–7. [CrossRef]
19. Mishra, P.R.; Gupta, I.; Pillai, N.R. Generalized avalanche test for stream cipher analysis. In Proceedings of the International Conference on Security Aspects in Information Technology, Haldia, India, 19–22 October 2011; Volume 7011 LNCS, pp. 168–180. [CrossRef]
20. Srinivasan, C.; Lakshmy, K.V.; Sethumadhavan, M. Measuring diffusion in stream ciphers using statistical testing methods. *Def. Sci. J.* **2012**, *62*, 6–10. [CrossRef]
21. Sosa-Gómez, G.; Rojas, O.; Páez-Osuna, O. Using hadamard transform for cryptanalysis of pseudo-random generators in stream ciphers. *EAI Endorsed Trans. Energy Web* **2020**, *7*. [CrossRef]

22. Madarro Capó, E.J.; Cuellar, O.J.; Legón Pérez, C.M.; Gómez, G.S. Evaluation of input—Output statistical dependence PRNGs by SAC. In Proceedings of the 2016 International Conference on Software Process Improvement (CIMPS), Aguascalientes, Mexico, 12–14 October 2016; pp. 1–6. [CrossRef]
23. Paul, G.; Maitra, S. RC4: Stream cipher and its variants. *RC4 Stream Cipher Its Var.* **2011**, 1–281. [CrossRef]
24. Grosul, A.L.; Wallach, D.S. *A Related-Key Cryptanalysis of RC4*; Rice University: Houston, TX, USA, 2000; pp. 1–13.
25. Matsui, M. Key collisions of the RC4 stream cipher. In *International Workshop on Fast Software Encryption*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5665 LNCS, pp. 38–50. [CrossRef]
26. Chen, J.; Miyaji, A. How to find short RC4 colliding key pairs. In *International Conference on Information Security*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 7001 LNCS, pp. 32–46. [CrossRef]
27. Maitra, S.; Paul, G.; Sarkar, S.; Lehmann, M.; Meier, W. New Results on Generalization of Roos-Type Biases and Related Keystreams of RC4. In *International Conference on Cryptology in Africa*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 222–239. [CrossRef]
28. Maximov, A. *Some Words on Cryptanalysis of Stream Ciphers*; Citeseer: Lund, Sweden, 2006.
29. Vergili, I.; Yücel, M.D. Avalanche and bit independence properties for the ensembles of randomly chosen n × n s-boxes. *Turk. J. Electr. Eng. Comput. Sci.* **2001**, *9*, 137–145.
30. Karell-Albo, J.A.; Legón-Pérez, C.M.; Madarro-Capó, E.J.; Rojas, O.; Sosa-Gómez, G. Measuring independence between statistical randomness tests by mutual information. *Entropy* **2020**, *22*, 741. [CrossRef]
31. Ibrahim, H.; Khurshid, K. Performance Evaluation of Stream Ciphers for Efficient and Quick Security of Satellite Images. *Int. J. Signal Process. Syst.* **2019**, *7*, 96–102. [CrossRef]
32. Gorbenko, I.; Kuznetsov, A.; Gorbenko, Y.; Vdovenko, S.; Tymchenko, V.; Lutsenko, M. Studies on statistical analysis and performance evaluation for some stream ciphers. *Int. J. Comput.* **2019**, *18*, 82–88.
33. RC4 Cipher Is No Longer Supported in Internet Explorer 11 or Microsoft Edge. Available online: https://support.microsoft.com/en-us/help/3151631/rc4-cipher-is-no-longer-supported-in-internet-explorer-11-or-microsoft (accessed on 5 July 2020).
34. SSL Configuration Required to Secure Oracle HTTP Server after Applying Security Patch Updates. Available online: https://support.oracle.com/knowledge/Middleware/2314658_1.html (accessed on 5 July 2020).
35. Satapathy, A.; Livingston, J. A Comprehensive Survey on SSL/ TLS and Their Vulnerabilities. *Int. J. Comput. Appl.* **2016**, *153*, 31–38. [CrossRef]
36. Soundararajan, E.; Kumar, N.; Sivasankar, V.; Rajeswari, S. Performance analysis of security algorithms. In *Advances in Communication Systems and Networks*; Springer: Singapore, 2020; Volume 656, pp. 465–476. [CrossRef]
37. Jindal, P.; Makkar, S. Modified RC4 variants and their performance analysis. In *Microelectronics, Electromagnetics and Telecommunications*; Springer: Singapore, 2019; Volume 521, pp. 367–374. [CrossRef]
38. Parah, S.A.; Sheikh, J.A.; Akhoon, J.A.; Loan, N.A.; Bhat, G.M. Information hiding in edges: A high capacity information hiding technique using hybrid edge detection. *Multimed. Tools Appl.* **2018**, *77*, 185–207. [CrossRef]
39. Tyagi, M.; Manoria, M.; Mishra, B. Effective data storage security with efficient computing in cloud. *Commun. Comput. Inf. Sci.* **2019**, *839*, 153–164. [CrossRef]
40. Dhiman, A.; Gupta, V.; Singh, D. Secure portable storage drive: Secure information storage. *Commun. Comput. Inf. Sci.* **2019**, 839, 308–316. [CrossRef]
41. Nita, S.; Mihailescu, M.; Pau, V. Security and Cryptographic Challenges for Authentication Based on Biometrics Data. *Cryptography* **2018**, *2*, 39. [CrossRef]
42. Zelenoritskaya, A.V.; Ivanov, M.A.; Salikov, E.A. Possible Modifications of RC4 Stream Cipher. *Mech. Mach. Sci.* **2020**, *80*, 335–341. [CrossRef]
43. Jindal, P.; Singh, B. Optimization of the Security-Performance Tradeoff in RC4 Encryption Algorithm. *Wirel. Pers. Commun.* **2017**, *92*, 1221–1250. [CrossRef]
44. Verdú, S. Empirical estimation of information measures: A literature guide. *Entropy* **2019**, *21*, 720. [CrossRef]
45. Hutson, A.D. A robust Pearson correlation test for a general point null using a surrogate bootstrap distribution. *PLoS ONE* **2019**, *14*. [CrossRef]
46. Liu, F.; Dong, Q.; Xiao, G. Probabilistic analysis methods of S-boxes and their applications. *Chin. J. Electron.* **2009**, *18*, 504–508.

47. Walpole, R.E.; Myers, R.H. *Probability & Statistics for Engineers & Scientists*; Pearson Education Limited: London, UK, 2012.
48. Siraj-Ud-Doulah, M. A Comparison among Twenty-Seven Normality Tests. *Res. Rev. J. Stat.* **2019**, *8*, 41–59.
49. Riad, A.M.; Shehat, A.R.; Hamdy, E.K.; Abou-Alsouad, M.H.; Ibrahim, T.R. Evaluation of the RC4 algorithm as a solution for converged networks. *J. Electr. Eng.* **2009**, *60*, 155–160.