NIST RANDOMNESS TESTS (IN)DEPENDENCE

Carmina GEORGESCU, Alina PETRESCU-NITA, Emil SIMION, Antonela TOMA
University Politehnica from Bucharest
Email of corresponding author: emil.simion@upb.ro

Abstract. In this paper we focus on three open questions regarding NIST SP 800-22 randomness test: the probability of false acceptance, the number of minimum sample size to achieve a given probability error and tests independence. We shall point out statistical testing assumptions, source of errors, sample constructions and a computational method for determining the probability of false acceptance and estimating the correlation between the statistical tests.

Key words: statistical testing, random bit generators.

1. INTRODUCTION

Statistical tests are an efficient tool for deciding if a set of independent observations, called measurements, belongs to a specific population or probability distribution; they are commonly used in the field of cryptography, specifically in randomness testing. Statistics can be useful in showing a proposed system is weak. Thus, one criterion in validating ciphers is that there is no efficient method for breaking it that brute force. That is, if we have a collection of cipher texts (and eventually the corresponding plain texts) all the keys have the same probability to be the correct key, thus we have uniformity in the key space. If we are analyzing the output of the cipher and find a non-uniform patterns than it can be possible to break it. However if we cannot find these non-uniform patterns no one can guarantee that there are no analytical methods in breaking it. Also statistical tests can be used for analyzing communication data and detect covert communications (steganographic systems) and anomalies in TCP flow (cyber attacks).

The paper is organized as follows. In section 2 we present the statistical tests assumptions sources of errors and sample constructions for the situation of testing the cryptographic algorithms. The statistical methods used in academic security evaluation of the AES candidates are generally based on "de facto" standard STS SP 800-22 [7], a publication of Computer Security Research Center, a division of NIST, that initially describes sixteen statistical tests (because improper evaluation of the mean and variance, the Lempel-Ziv test was dropped from the revised version). Beside the above there exist other several statistical testing procedures and tools specified in Donald Knuth's book [2], The Art of Computer Programming, Seminumerical Algorithms, the Crypt-XS suite of statistical tests developed by researchers at the Information Security Research Centre at Queensland University of Technology in Australia, the DIEHARD suite of statistical tests developed by George Marsaglia [4], TestU01, a C library for empirical testing of random number generators developed by P. L'Ecuyer and R. Simard [3]. In section 3 we discuss about STS SP 800-22, the statistical cryptographic evaluation standard used in AES candidates' evaluation.

One weak point of the statistical test suite is that does not compute second type error, which represents the probability of failing to reject the null hypothesis when it is false. Another open question in the academic world is regarding the independence of SP 800-22 statistical tests. In section [5] we propose a procedure to solve by simulation the problem of tests correlation.

Finally in section [5] we conclude.

2. STATISTICAL TESTING ASSUMPTIONS, SOURCE OF ERRORS AND SAMPLE CONSTRUCTIONS

Statistical hypothesis testing is a mathematical technique, based on sample data, used for supporting the decision making on the theoretical distribution of a population. In the case of statistical analysis of a cryptographic algorithm the sample is the output of the algorithm from different inputs for the key and plain text. Because we deal with sample data from the population, the decision process of the population's

probability distribution is prone to errors. To meet this challenge, we model the decision making-process with the aid of two statistical hypotheses: the null hypothesis denoted by H_0 - in this case, the sample does not indicate any deviation from the theoretically distribution - and the alternative hypothesis H_A - when the sample indicates a deviation from the theoretically distribution.

There can be three types of errors:

- First type error (also known as the level of significance), i.e. the probability of rejecting the null hypothesis when it is true:

$$\alpha=P(\text{reject } H_0 | H_0 \text{ is true})$$

- Second type error, which represents the probability of failing to reject the null hypothesis when it is false:

$$\beta=P(\text{accept } H_0 | H_0 \text{ is false}),$$

the complementary value of β is denoted as the test's power:

$$1-\beta=P(\text{reject } H_0 | H_0 \text{ is false})$$

- Third type error happens when we ask a wrong question and use the wrong null hypothesis. This error is less analytical and requires that we pay attention before starting our analysis.

These two errors, α and β, can't be minimized simultaneously since the risk β increases as the risk α decreases and vice-versa. From this reason one solution is to have under control the value of α and compute the probability β. Sometimes we ask a wrong question and use the wrong null hypothesis. This type of error is called *type III error*.

The strong law of large numbers is usually used for randomness testing of binary sequences. This theorem can be stated in two different ways:

- The first form is derived from *Leapunov's* theorem and states that if (f_n) is a sequence of independent random variables with the same distribution (with mean *m* and variance σ), then for "large" values $n > 30$ we have:

$$P(a < f_1 + ... + f_n < b) \approx \Phi\left(\frac{b - n\cdot m + \frac{1}{2}}{\sigma\sqrt{n}}\right) - \Phi\left(\frac{a - n\cdot m - \frac{1}{2}}{\sigma\sqrt{n}}\right).$$

- The second form is derived from *De Moivre's* theorem and formulas and states that if (f_n) is a sequence of binary independent random variables with $P(X=1)=p$ and $P(X=0)=q$, then for "large" $n > 30$ values we have:

$$P(a < f_1 + ... + f_n < b) \approx \Phi\left(\frac{b - n\cdot p + \frac{1}{2}}{\sqrt{npq}}\right) - \Phi\left(\frac{a - n\cdot p - \frac{1}{2}}{\sqrt{npq}}\right).$$

The above formulas are good estimations even in the case we have small values of *n* and *a* and *b* very close one to other.

The analysis plan of the statistical test includes decision rules for rejecting the null hypothesis. These rules can be described in two ways:

Decision based on *P*-value. In this case, we consider *f* to be the value of the test function and compare the *P*-value, defined as $P(X < f)$, with the value α and decide on the null hypothesis if *P*-value is greater than α.

The "critical region" of a statistical test is the set which causes the null hypothesis to be rejected; the complementary set is called the "acceptance region". In the acceptance region, we shall find the ideal results of the statistical test.

Because for each statistical test the rejection rate α is a probability, which is "approximated" from the sample data, we need to compute the minimum sample size necessary for achieving the desired rejection rate α. Also, the sample must be:

- independent;
- "from the same world", i.e. governed by the same distribution.

A way to construct samples for testing block ciphers is to setup the plain text and the key: $X_i=E(P_i;k_i)$ where E is the encryption function, P_i is the set of plain texts and k_i is the set of keys. For each plain text input P_i and each encryption key k_i the output from the encryption function must have a uniform distribution. To test this assumption, for AES candidates, Soto [9] constructed the samples with low/high density plain text/key (a low density text/key it is a text/key with a small number of 1s in oppositions a high density text/key which it is a text/key with a small number of 0s). As we can see using this type of construction the samples are not independent variables because they are connected by means of the encryption function E. Are the results of the statistical tests relevant if this assumption is not true? If the statistical test accepts the null hypothesis then we can say that there is not enough evidence for the non-uniformity of the sample.

If a cryptographic primitive passes a statistical test, it does not mean that the primitive is secure. For example, the predictable sequence 01010…01 is "perfect" if we analyze it with the bit frequency test. This is one of the reasons why we should be "suspicious" if we obtain perfect results. To avoid these situations, in some cases, it is indicated to include the neighborhood of the ideal result in the critical region.

NIST SP 800-90A [7] contains the specifications of four cryptographic secure PRBG for use in cryptography based on: hash functions, hash-based message authentication code, block ciphers and elliptic curve cryptography. Some problems with the later one (Dual_EC_DRBG) were discovered in 2006: the random numbers it produces have a small bias and it raise the question whether the NSA put a secret backdoor in Dual_EC_DRBG. It was proved in 2013 that (Dual_EC_DRBG) has flows. Internal memos leaked by a former NSA contractor, Edward Snowden, suggest that NSA generated a trapdoor in Dual_EC_DRBG. To restore the confidence on encryption standards, NIST reopen the public vetting process for the NIST SP 800-90A.Thus, if algorithm will fail to certain tests then it should not be used in cryptographic applications because an attacker will be able to predict the behavior of the algorithm or, even worse, may indicate the existence of certain trapdoors.

3. A VIEW ON STS SP 800-22

Pseudorandom bit generators (PRBG) are cryptographically secure if pass *next bit test*, that is there is no polynomial time algorithm which, given the first l-bits of the output, can predict l+1-bit with probability significantly greater than 0.5 and, in the situation that a part of PRBG is compromised, it should be impossible to reconstruct the stream of random bits prior to the compromising. Yao [10] proved that PRBG pass next bit test if and only if passes all polynomial time statistical tests. Because practically is not feasible to test PRBG for all polynomial statically tests we need to find a representative, polynomial time, statistical testing suite such as STS SP 800-22. Because STS SP 800-22 is a standard, we shall focus on it rather than other statistical test suites ([2], [3] or [4]). STS SP 800-22 (the revised version) consists of fifteen statistical tests, which highlight a certain fault type proper to randomness deviations. Each test is based on a computed test statistic value f, which is a function of the sample. The statistic test is used to compute a P-value=$P(f|H_0)$ that summarizes the strength of the evidence against the null hypothesis. If the P-value is greater then the null hypothesis is accepted (the sequence appears to be random). The tests are not jointly independent, making it difficult to compute an overall rejection rate (i.e. the power of the test). Recall that the tests $T_1,\ldots,T_{15}$ are jointly independent if $P(T_{i1},\ldots,T_{ik})=P(T_{i1})\cdot\cdot\cdot P(T_{ik})$ for every subset $\{i_1,\ldots,i_k\}$ of $\{1,\ldots,15\}$. Obviously, jointly independent tests are pair wise independent. The converse is not true [1]. If the statistical tests would be independent, then the overall rejection rate, would be computed using the probability of the complementary event $1-(1-\alpha)^{15}\approx 0.14$.

In table II we can see the reference distribution of NIST statistical tests:

Table II

Test	**Reference distribution**
Frequency (monobit) test	half normal
Frequency Test within a Block	$\chi^2(N)$
Runs Test	Normal
Test for the Longest Run of Ones in a Block	$\chi^2(K)$

Test	Reference distribution
Binary Matrix Rank Test	$\chi^2(2)$
Discrete Fourier Transform (Spectral) Test	Normal
Non-overlapping Template Matching Test	$\chi^2(N)$
Overlapping Template Matching Test	$\chi^2(K)$
Maurer's "Universal Statistical" Test	Normal
Linear Complexity Test	$\chi^2(K)$
Serial Test	$\chi^2(2^{m-1})+ \chi^2(2^{m-2})$
Approximate Entropy Test	$\chi^2(2^m)$
Cumulative Sums (Cusum) Test	Normal
Random Excursions Test	$\chi^2(5)$
Random Excursions Variant Test	half normal

STS SP 800-22 provides two methods for integrating the results of the tests, namely percentage of passed tests and the uniformity of *P* values. The experiments revealed that these decision rules were insufficient and therefore researchers considered their improvement would be useful. Therefore, in [6], were introduced new integration methods for these tests:

- Maximum value decision, based on the max value of independent test statistics T_i, i=1,…,*n*. In this case the maximum value of the random variables was computed; the repartition function of the max value, P(max(T_1,…,T_n)<x), being the product of the repartition functions of the random variables T_i: $\prod_{i=1}^{n} P(T_i < x)$;
- Sum of square decision, based on the sum of squares *S* of the results of the tests (which have a normal distribution). The distribution of *S*, in this case, is χ^2, the freedom degrees given by the number of partial results which are being integrated.

Weak points of STS SP 800-22:

- Fixed first order error α=0.01;
- The tests are not evaluating the second order error, which represents the probability to accept a false hypothesis.

Let us suppose that we have a binary sequence produced by a random variable X with P(X=1)=p_0 and P(X=0)=q_0=1-p_0 and test the null hypothesis H_0: p=p_0 (if p_0=0.5 then we perform a uniformity test) against the alternative hypothesis H_1: p=p_1 with $p_1 \neq p_0$.

In [6] the possibility of extending the frequency test from STS SP 800-22 to arbitrary level of significance α (and computing $\beta(p_1)$) is presented by computing, for some n>30, the second order probability:

$$\beta(p_1) = \Phi\left(\sqrt{\frac{p_0 q_0}{p_1 q_1}}\left(u_{1-\frac{\alpha}{2}} - \frac{n(p_1 - p_0)}{\sqrt{np_0 q_0}}\right)\right) - \Phi\left(\sqrt{\frac{p_0 q_0}{p_1 q_1}}\left(u_{\frac{\alpha}{2}} - \frac{n(p_1 - p_0)}{\sqrt{np_0 q_0}}\right)\right),$$

where $u_{1-\frac{\alpha}{2}}$ and $u_{\frac{\alpha}{2}}$ stand for quantiles of the normal distribution and $q_1 = 1 - p_1$.

Deriving formulas for all STS 800-22 is rather difficult, however one can estimate the value of β (as function of p_1) by using computational methods. Namely, having a validated randomness source (for example a hardware device), we propose to estimate the value of $\beta_j(p_1)$ for every j=1…15 NIST statistical test in the following way:

- set every i^{th} bit of the sample (of size *n*) to 0 and compute the number of failures;
- increment *i* and repeat the experiment until the STS 800-22 will not distingue the tested sequence from a random one;
- the value of i/n is an estimation of $\beta_j(p_1)$;

The false acceptance rate of the full NIST battery test is the $\max_j \beta_j(p_1)$.

A problem that we encountered is that of finding the minimum sample size to achieve, with error $\varepsilon>0$ and a given rejection rate α. Formally, if (f_n) is a sequence of binary independent random variables with $P(X=1)=p$ and $P(X=0)=q$, where p is unknown, we need to find n such that:

$$P(|\nu_n - p| < \varepsilon) \geq 1-\alpha,$$

where:

$$\nu_n = \frac{f_1 + ... + f_n}{n}$$

represents the sequence of relative occurrence of the symbol 1. Using the *De Moivre* form of the strong law of large numbers, the equation can sequentially be rewritten:

$$P(|f_1 + ... + f_n - pn| < \varepsilon n) \geq 1-\alpha$$

$$\Phi\left(\frac{\varepsilon}{\sigma}\sqrt{n}\right) - \Phi\left(-\frac{\varepsilon}{\sigma}\sqrt{n}\right) \geq 1-\alpha$$

$$n \geq \frac{p(1-p)}{\varepsilon^2} u^2_{1-\frac{\alpha}{2}}.$$

Finally, using the fact that $p(1-p)\leq 0.25$, we find that the minimum sample size is:

$$n_{\min} = \left\lceil \frac{1}{4\varepsilon^2} u^2_{1-\frac{\alpha}{2}} \right\rceil.$$

Numerical computation values are presented in Table III. We interpret the data as follows. The relative frequency ν_n estimate p with an error ε and this statement is valid in $100(1-\alpha)$ of the cases. In some practical situations, the sample size can be fixed. In this case, after the estimation of p, we can fix the confidence $1-\alpha$ and compute the error ε. For example, in the case of NIST STS SP 800-22 test vectors, where sample size is $n=100$ and confidence 0.9, numerical computations give $\varepsilon=0.0825$.

Table III Connection between error, confidence and minimum sample size

Error ε	**Confidence 1-α**	**Minimum sample size n_{min}**
0.01	0.90	6724
0.01	0.95	9604
0.01	0.99	16641
0.03	0.90	748
0.03	0.95	1068
0.03	0.99	1835
0.05	0.90	269
0.05	0.95	385
0.05	0.99	661

In [5] there are some comments about NIST statistical testing methodology: ambiguous hypothesis (does not specify the family of distribution and/or the alternative), error quantification (NIST does not give the size of the category-test decisions), power of the test suite, invariant test (cryptographically equivalent tests performed on the same sample do not necessary give the same result), and inadmissible tests (the existence of better tests).

After the process of evaluation of AES candidates researchers [11] reported that the test setting of Discrete Fourier Transform test (designed is to detect periodic features in the tested sequence that would indicate a deviation from the assumption of randomness) and Lempel-Ziv test (designed to see if the

sequence can be compressed and will be considered to be non-random if it can be significantly compressed) of the STS SP 800-22 are unsuitable:

- threshold value and the variance σ^2 of theoretical distribution, respectively;
- the setting of standard distribution, which has no algorithm dependence (SHA-1 for million bit sequences) and the re-definition of the uniformity of P-values (based on simulation).

Because the mean and variance of Lempel-Ziv test were evaluated using samples generated by an algorithm, in the revised version of STS SP 800-22 the Lempel-Ziv was dropped.

4. PROPOSAL FOR ESTIMATING TESTS (IN)DEPENDENCE

Because the SP 800-22 test functions have a complicated form, in order to check the independence of tests *i* and *j* we propose the following procedure:

i) implement the NIST SP 800-22 testing suite;
ii) using a "good" pseudorandom generator GPA test *N* binary samples;
iii) for each test *i* define the Bernoulli random variable T_i which gives 1 if the sample pass the test and 0 otherwise;
iv) estimate the value of $P(T_i \text{ and } T_j)- P(T_i)\ P(T_j)$. If the tests are independent then this value should be close to zero.
v) find the highest value of the above value for i and j.

5. CONCLUSIONS

In this paper we have proposed numerical methods for solving three open problems regarding the standard NIST 800-22 STS: determining the probability of accepting a false hypothesis, finding the number of minimum sample size to achieve a given probability error and the (in)dependence of statistical tests.

6. ACKNOWLEDGEMENTS

This work has been funded by University Politehnica of Bucharest, through the "Excellence Research Grants" Program, UPB – GEX. Identifier: UPB–EXCELENTA–2016 Research project title, Contract number 22/26.09.2017 (acronym: 406).

7. REFERENCES

[1] Sergei Natanovich Bernstein, *Theory of Probability*, 4th ed. (in Russian), Gostechizdat, Moscow-Leningrad, 1946.
[2] Donald Knuth, *The Art of Computer Programming, Seminumerical Algorithms*, Volume 2, 3rd edition, Addison Wesley, Reading, Massachusetts, 1998.
[3] P. L'Ecuyer and R. Simard, *TestU01: A C library for empirical testing of random number generators*, ACM Transactions on Mathematical Software, 33, 4, Article 22, 2007.
[4] George Marsaglia, *DIEHARD Statistical Tests: http://stat.fsu.edu/ geo/diehard.html*.
[5] S. Murphy, *The power of NIST's statistical testing of AES candidates*, Preprint. January 17, 2000.
[6] A. Oprina, A. Popescu, E. Simion and Gh. Simion, *Walsh-Hadamard Randomness Test and New Methods of Test Results Integratio*n, Bulletin of Transilvania University of Braşov, vol. 2(51) Series III-2009, pg. 93-106.
[7] *** NIST Special Publication 800-22, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, 2001.
[8] *** NIST standards: http://www.nist.gov/, http://www.csrc.nist.gov/.
[9] J. Soto, *Randomness Testing of the Advanced Encryption Standard Candidate Algorithms*, NIST IR 6390, September 1999.
[10] A.C. Yao, *Theory and Applications of Trapdoor Functions*, 23rd Ann. Symp. Foundations of Computer Science, pp. 80-91 1982.