Efficient Online Tests for True Random Number Generators

Werner Schindler

Bundesamt für Sicherheit in der Informationstechnik (BSI)
Godesberger Allee 183, 53175 Bonn, Germany
Werner.Schindler@bsi.bund.de

Abstract. General problems and difficulties are discussed which have to be considered when testing true random numbers. Requirements are formulated which appropriate online tests should fulfill. Then we propose an online test procedure which meets these requirements.

Keywords: True random number generator, statistical test, online test.

1 Introduction

Random numbers play an important role in many cryptographic applications. Random numbers are used, for instance, to generate random session keys, signature parameters and challenges for challenge-response protocols and zero knowledge proofs. Roughly speaking, the class of random number generators can be divided in three subclasses. First, there are true (physical) random number generators. Usually, an analog signal generated by a physical noise source is digitalized after uniform time intervals, e.g. by a comparator. Many true random number generators use a mathematical follow-up-treatment, i.e. an algorithm applied on the digitalized analog signals. The goal of a mathematical follow-up treatment is to reduce or at least to mask weaknesses of the digitalized analog signals. In contrast, pseudorandom number generators derive (pseudo-)random numbers deterministically from a randomly chosen seed. Pseudorandom number generators are very cheap as they merely require some additional lines of code. Their drawback is that the whole entropy is "contained" in the seed. Finally, there are "mixed" generators which derive random numbers from user's interaction (mouse movement or key strokes) or register values of the used PC.

In the following we restrict our attention to **t**rue **r**andom **n**umber **g**enerators (TRNGs). We denote the digitalized analog signals as *das-random numbers* and the values after the mathematical follow-up treatment has been applied on as *internal random numbers*. Upon an external call the TRNG outputs internal random numbers.

Obviously, non-appropriate random numbers can weaken strong cryptographic mechanisms considerably. To assess a TRNG a mathematical model of the physical noise source should be evaluated and analyzed, and suitable (that is, suitable with respect to the mathematical model) statistical tests should be applied on

Ç.K. Koç, D. Naccache, and C. Paar (Eds.): CHES 2001, LNCS 2162, pp. 103–117, 2001.

the das-random numbers generated by some TRNGs prototypes. A lot of research work has been devoted to the generation of good physical noise sources and the determination of suitable statistical tests ([2], [10], [12] etc.).

Tolerances of the components of the random noise sources may cause that a particular TRNG produces worse das-random numbers than the carefully investigated prototypes. Further, ageing of these components may also affect the statistical quality of the generated das-random numbers. As a consequence statistical tests ("online tests") have to be executed while the TRNG is in operation to ensure that the generated random numbers are appropriate. Especially, if the TRNG is integrated in a smart card online tests should run fast, require only few lines of additional code and little memory.

Section 2 considers the question whether the das-random numbers or the internal random numbers should be tested and in Sect. 3 general demands are formulated which online tests should fulfill. In Sect. 4 we briefly discuss the drawbacks of a widely used online test. In Sects. 5 – 9 a new online test procedure is described, analyzed and illustrated at two examples. The paper ends with final remarks.

2 Which Random Numbers Should Be Tested?

If there is no mathematical follow-up treatment the das-random numbers coincide with the internal random numbers. Otherwise, the online tests can be applied on the das-random numbers or, alternatively, to the internal random numbers.

For many cryptographic applications it is inevitable that random numbers cannot be determined or guessed with a reasonable probability, even if predecessors or successors are known. Pseudorandom number generators rely on the complexity of their algorithms which shall ensure *practical security* (see, e.g. [1]). For TRNGs the situation is much more comfortable as the total entropy of a das-random number sequence increases per generated das-random number. If the increase of entropy is sufficiently large, this ensures *theoretical security*. (Clearly, a lucky attacker could guess a randomly chosen session key, for instance, but if the key length is sufficiently large his success probability is negligible.)

Hence it is desirable to control the increase of entropy. Unfortunately, entropy is not a function of random numbers but of random variables. In the following we will interpret das-random numbers as values assumed by random variables whose distribution usually is at least not exactly known. We use statistical tests to compare das-random number sequences with sequences generated by ideal random number generators.

Remark 1. In [3] a variant of Maurer's "universal" statistical test (cf. [11], [4]) is introduced. Its test value is closely related to the entropy per bit block *provided that the bits were generated by a stationary binary random source with finite memory*. If this is not the case, however, as for Maurer's test the test value need not yield a reliable estimate of the entropy. For pseudorandom bits generated by a linear feedback shift register, for example, the increase of entropy per bit

equals zero whereas the test value "suggests" a considerable amount of entropy per bit block. Moreover, as Maurer's test it requires a lot of memory and gigantic sample sizes. Hence both tests are not suitable as online tests but may be used for the investigation of TRNG prototypes.

Example 1. Let the TRNG produce binary das-random numbers and let a linear feedback shift register of length 63 with primitive feedback polynomial be synchronized with the digitalization of the analog noise signal. In each time step the feedback shift register delivers an internal random number (a single bit). The actually generated das-random number is XOR-ed to the feedback value and the sum is fed back into the shift register.

For each initial value of the feedback shift register this mathematical follow-up treatment is a one-to-one mapping and thus cannot increase the average entropy per bit. Statistical weaknesses of the das-random numbers are not reduced but only transferred into others. If, for example, the das-random numbers are independent but not equidistributed (i.e., if the probability for "0" is not 0.5) the internal random numbers are essentially equidistributed but dependent. Unless its linear complexity profile is tested applying statistical tests on the internal random number sequence will presumably even not detect the worst case, i.e. if the physical noise source has totally broken down. In fact, after this moment the das-random numbers are constant and the internal random numbers are generated deterministically.

This brief analysis of Example 1 has revealed an important fact: The internal random numbers may pass certain statistical tests which the das-random numbers do not. However, this does not necessarily imply that the mathematical follow-up treatment reduces statistical weaknesses of the das-random numbers. Maybe they are merely masked and transformed into others. An increase of the entropy per bit can only achieved by a data compression which in turn lowers the bit rate. (In the simplest case non-overlapping bits are XORed.) Of course, the das-random numbers may not be equidistributed and there may exist dependencies on preceding das-random numbers but in contrast to the internal random numbers there will not exist complicated algebraic dependencies. Consequently, the das-random numbers but not the internal random numbers should be tested, especially if the TRNG is used for sensitive applications.

3 Which Requirements Should Online Tests Fulfill?

As motivated in the previous section online tests should be applied on das-random numbers which usually (but not necessarily) are single bits.

Definition 1. A *realization* of a random variable X is a value assumed by X. The term *iid* abbreviates "independent and identically distributed". We call a random variable *binary* if it only assumes the values 0 and 1. A random variable X is called *equidistributed* on a finite set $\Omega := \{\omega_1, \ldots, \omega_k\}$ if $\mathrm{Prob(X} = \omega_\mathrm{j}) = 1/\mathrm{k}$ holds for all $j \leq k$. Applying a statistical test on a sample delivers a numerical value called *test value* or *test statistic*.

Mathematical Model and Definitions. Das-random numbers assume values in Ω_{das} (usually, but not necessarily, $\Omega_{\text{das}} = \{0, 1\}$). We assume that the das-random numbers are realizations of random variables $B_1, B_2, \ldots$ so that the test value $\widetilde{T}$ itself may be interpreted as a realization of a random variable T, the so-called *test variable*. The distribution of $B_1, B_2, \ldots$ and that of T clearly depend on the particular TRNG. For an *ideal random number generator* (a fiction!), of course, the random variables $B_1, B_2, \ldots$ are iid and equidistributed on Ω_{das}. To avoid clumsy formulations we call bit sequences generated by an ideal random number generator *ideal sequences*. A χ^2-distribution with k degrees of freedom is denoted with χ^2_k.

We will use statistical tests to compare das-random numbers with ideal sequences. To a certain degree statistical deviations of the das-bits from ideal sequences, however, are tolerable. (Assume, for example, that the das-bits were realizations of iid random variables $B_1, B_2, \ldots$ with $\text{Prob}(\text{B}_\text{j} = 1) = 0.49$. Then the average entropy per das-bit was about 0.9997.) Clearly, "tolerable" essentially depends on the intended applications for which the random numbers shall used and, to a certain degree, on the mathematical follow-up treatment which may increase the entropy per random number (cf. Sect. 2). If the statistical properties of the das-random number sequence deviate too much from that of ideal sequences the online test should generate a noise alarm. The preceding considerations suggest various requirements which online tests should fulfill. Recall that due to tolerances of components or ageing effects a TRNG may produce worse das-random number sequences than the carefully investigated TRNG prototypes. Clearly, also ideal sequences would occasionally fail statistical tests.

Requirements for online tests.
(R1) An online test has to detect a total breakdown of the noise source very soon.
(R2) An online test should detect non-tolerable statistical weaknesses of the das-random numbers.
(R3) The probability for a noise alarm should be small if the deviation of the statistical properties of the das-random numbers from that of ideal sequences is tolerable.
(R4) An online test should run fast and require only a few lines of code and little memory.

4 Drawbacks of a Widely Used Online Test and General Difficulties

In this section we discuss briefly problems of a near-at-hand online test procedure which is widely used in practice.

Example 2. Let the TRNG generate binary das-random numbers. A FIFO stores internal random numbers. If the FIFO is full the generated internal random numbers are neither used nor stored. Upon external request, the FIFO outputs internal random numbers. Periodically every minute and whenever the FIFO has

to be refilled $n = 320$ consecutive das-bits are segmented into 80 non-overlapping four-bit blocks $\widetilde{W}_1 := (\widetilde{B}_1, \ldots, \widetilde{B}_4), \ldots, \widetilde{W}_{80} := (\widetilde{B}_{317}, \ldots, \widetilde{B}_{320})$. On each sample a χ^2-test (or more precisely, a χ^2-test for goodness of fit ([7], 69)) is applied. For this, the $\widetilde{W}_j$ are interpreted as binary representations of four-digit numbers. For $i = 0, \ldots, 15$ the frequencies $fr(i) := |\{j \leq 80 \mid \widetilde{W}_j = i\}|$ are determined and finally

$$\widetilde{T} := \sum_{i=0}^{15} \frac{(fr(i) - 5)^2}{5} \quad . \tag{1}$$

The null hypothesis, i.e. that the tested sample was generated by an ideal noise source, is rejected if the test value $\widetilde{T}$ exceeds 65.0. A rejection of the null hypothesis causes a noise alarm which puts the TRNG out of service. The TRNG has to pass extensive investigations before it can manually be restarted by an authorized person.

The system administrator has laid down that there should not occur more than 0.027 noise alarms per TRNG and year in average if the TRNG produces tolerable das-random numbers. (The numerical value 0.027 clearly depends on the concrete application. For other applications, however, smaller values may be appropriate.) To reach this goal the designer of this online test has chosen the rejection area $(65.0, \infty)$. His considerations were the following: It is reasonable to assume that each TRNG executes about 530000 χ^2-tests per year. If the das-random numbers were generated by an ideal random number generator the test variable T was approximately χ^2_{15}-distributed ([7], 69) and thus Prob(T > 65.0) $\approx 3.4 \cdot 10^{-8}$. This yields an expected number of 0.018 noise alarms per year. As the das-random numbers generated by the investigated TRNG prototypes did not reveal serious statistical weaknesses the online test designer expects that the average number of noise alarms per TRNG and year will not exceed the given upper bound $0.027 = 1.5 \cdot 0.018$.

However, this argumentation is not quite correct. Even for ideal sequences the test variable T is not exactly but merely *approximately* χ^2_{15}-distributed. In fact, the 4-tuples are multinomially distributed and the exact probability Prob(T > 65.0) is about $3.8 \cdot 10^{-7}$. If X denotes a χ^2_{15}-distributed random variable the *absolute error* |Prob(T > 65.0) − Prob(X > 65.0)| is surely small but not the *relative error*

$$\frac{|\text{Prob(T} > 65.0) - \text{Prob(X} > 65.0)|}{\text{Prob(X} > 65.0)} \approx 10.1 \quad . \tag{2}$$

(Note that we use Prob(X > 65.0) as denominator but not Prob(T > 65.0) because the predictions of the test designer are based on the approximate probability.) For the scenario described in Example 2 this means that even ideal random number generators would cause about $11 \cdot 0.018 \approx 0.2$ noise alarms per year in average; the TRNGs maybe even more. Anyway, this exceeds the upper bound 0.027 considerably. To avoid this, of course, the test parameter 65.0 could be increased (e.g. to 75.0). However, as the amount of increase is not based on a solid computational basis but more or less arbitrary it may happen that even

serious statistical weaknesses will not be detected. In particular, a strong (data-compressing and hence throughput-reducing) mathematical follow-up treatment is absolutely inevitable.

The χ^2_{15}-approximation may seem to be terribly bad. However, this is not the case: Note that $\mathrm{Prob}(\mathrm{T} > 30.6) = 0.01025$ and $\mathrm{Prob}(\mathrm{X} > 30.6) = 0.00993$, for example, with an relative error of about 0.03. The relative error is maximal at the tail of the χ^2_{15}-distribution, i.e. for large rejection boundaries z. Increasing the sample size (here: 320 bits) reduces the relative error for each fixed value z (here: for $z = 65.0$) since the exact distributions converge to the χ^2_{15}-distribution as the sample sizes tend to infinity.

However, these considerations point to a serious general problem. In particular, especially if the sample size of a statistical test is small, one has to be very careful when using an approximate distribution of the respective test variable at its tail. To obtain the exact rejection probability for ideal sequences in Example 2 we just had to count the tuples in $(\{0,1\}^4)^{80}$ for which the χ^2-test value is ≤ 65.0. However, although symmetries were exploited the computational effort was considerable. For non-ideal sequences it was *not practically feasible* to decide whether demand (R2) or (R3), resp., is fulfilled. However, online tests with such unpleasant properties are widely used in practice. In many cases even for ideal sequences the exact distribution of the test variable cannot be determined.

5 A New Online Test Procedure

In Sect. 5 we describe a new online test procedure. We will show later that it meets all requirements formulated in Sect. 3. In Sect. 8 it will be illustrated at two examples.

Step 1: First, the statistician has to choose a statistical test, the so-called "basis test", and to fix its sample size n. This may be a χ^2-test or any tests from [7], for example, provided that the needed memory, the lines of code and the execution time are acceptable for the used device and the intended applications. Ideally, the basis test should be chosen with respect to the mathematical model of the TRNG, or more precisely, of the random variables $B_1, B_2, \ldots$. (Of course, this mathematical model should have been confirmed by extensive statistical investigations of some TRNG prototypes. As the choice of the basis test does not affect the general principle of our online test procedure we will not pursue this topic in the remainder.) In the following $\mathrm{E}_0(T)$ denotes the mean of the test variable T under the null hypothesis, that is, if the random variables $B_1, B_2, \ldots$ were iid and equidistributed on Ω_{das}.

Step 2: With respect to the intended applications minimal requirements on the distribution of the random variables $B_1, B_2, \ldots$ have to be specified. (Example: Based on the mathematical model of the noise source and the evaluation of TRNG prototypes the test designer concludes that the binary random variables $B_1, B_2, \ldots$ are Markovian. For the intended applications $\mathrm{Prob}(\mathrm{B_j} = 1) \in$

$[0.48, 0.52]$ and $|\mathrm{Prob}(\mathrm{B_{j+1}} = 1 \mid \mathrm{B_j} = 0) + \mathrm{Prob}(\mathrm{B_{j+1}} = 0 \mid \mathrm{B_j} = 1) - 1| \leq 0.01$ are viewed as sufficient. In particular, it may be reasonable to choose a basis test which considers the one-step transition frequencies.) Time intervals or events have to be specified after those a basis test has to be executed, e.g.: always, one basis test per second, one basis test after an external call for random numbers, basis tests within the idle time of the device (if the TRNG is part of a larger cryptographic system) etc. With regard to the intended applications a reasonable upper bound for the average number of noise alarms within a time interval (year, month etc.) has to be specified. This upper bound should not be exceeded for any distribution of the random variables $B_1, B_2, \ldots$ which meets the minimal requirements specified before. Moreover, the consequences of a noise alarm have to be laid down, e.g.: The TRNG is put out of service, no further random numbers are produced till a check of the noise source and / or a manual restart of the TRNG, or something like that.

Step 3: A *test suite* consists of at most N basis tests. The basis test values are denoted with $\widetilde{T}_1, \widetilde{T}_2, \ldots$ while $\widetilde{H}_0 := \mathrm{E}_0(T)$. In step $j \geq 1$ a basis test is performed, and the basis test value $\widetilde{T}_j$ is determined. Then $\widetilde{H}_j := (1-\beta)\widetilde{H}_{j-1} + \beta\widetilde{T}_j$ is computed ($\beta \ll 1$) and rounded to a multiple of 2^{-c} where c is a fixed integer. Moreover, the following decision rules have to be considered:

$$\text{(A): if } \widetilde{T}_j \notin [r, s] \quad \Rightarrow \quad \text{noise alarm} \tag{3}$$

$$\text{(B): if } \widetilde{T}_{j-k+1}, \ldots, \widetilde{T}_j \notin [t, u] \quad \Rightarrow \quad \text{stop the test suite} \tag{4}$$

$$\text{(C): if } \widetilde{H}_j \notin [v, w] \quad \Rightarrow \quad \text{stop the test suite} \tag{5}$$

The parameter r and s should be chosen that a violation of decision criterion (A) is absolutely unlikely unless the random noise source has totally broken down. Consequently, a violation of decision rule (A) causes a noise alarm. Alternatively, if x consecutive test suites have been stopped due to (B) or (C) this also causes a noise alarm. Otherwise, after a test suite has been finished (due to a stop or because N basis tests have been executed) the next test suite begins.

The choice of the parameters $n, N, \beta, c, r, s, k, t, u, v, w$ and x should consider the goals formulated in Step 2. Without loss of generality we may assume that the parameters v and w are multiples of 2^{-c}. We point out that storing and updating the "history value" $\widetilde{H}_j$ needs no more than integer arithmetic. For this, we set $\beta := 2^{-b}$ for a suitable integer b. Before the test suite begins $\widetilde{H}_0 := \mathrm{E}_0(T)$ is rounded to a multiple of 2^{-c}. To update the history value $\widetilde{H}_{j-1}$ the actual basis test value $\widetilde{T}_j$ is rounded to a multiple of 2^{-c} and then

$$\widetilde{H}_j := ((2^b - 1)\widetilde{H}_{j-1} + \widetilde{T}_j + 2^{b-1}) >> b \tag{6}$$

is calculated. (As usually, "$>> b$" denotes a right shift of b bits.) We point out that if $\widetilde{H}_j$ is calculated with a floating point arithmetic the basis test value $\widetilde{T}_j$ need not to be rounded before (cf. Remark 2 in Sect. 7).

6 Rationale and Advantages of the New Online Test Procedure

Roughly speaking, statistical results get more reliable the more tests are performed. The value $\widetilde{H}_j$ "contains" the history of the actual test suite up to step j without storing the test values $\widetilde{T}_1, \widetilde{T}_2, \ldots, \widetilde{T}_j$ explicitely. Decision rules (A) and (B) shall detect a total breakdown of the noise source or if the statistical quality of the das-random numbers has rapidly become worse, resp. The main task of decision rule (C), however, is to detect weaknesses in the long-term behaviour.

If the basis test values can be calculated with integer arithmetic (which should exist anyway) then the whole online test procedure needs no more than integer arithmetic (cf. Sect. 5). As the evaluation of the decision rules (A), (B) and (C) requires only little running time, a few extra lines of code and little extra memory our online test procedure perfectly meets demand (R4). In the worst case decision rule (A) requires only a few das-random numbers more than the sample size n of one basis test to detect a total breakdown of the noise source. Hence requirement (R1) is also fulfilled. In Example 2 we described an online test which is widely used in practice. Even for ideal random number generators it required enourmous computational power to determine the expected number of noise alarms within a particular time interval. For (non-ideal) TRNGs however, the system designer had almost no control what is going on. For the online test procedure described in Sect. 5, however, for each parameter set *and for each assumed distribution* of the $B_1, B_2, \ldots$ we can at least approximately determine the expected number of noise alarms within a time interval. That is, we also have control on the effects of the test procedure if it is applied on non-ideal das-random number sequences. A suitably chosen parameter set n, N, b ($\beta := 2^{-b}$), c, r, s, k, t, u, v, w and x supports the goals formulated in Step 2 of Sect. 5. Especially, it meets the requirements (R2) and (R3).

7 Mathematical Background

In this section we determine the average number of test suites until a noise alarm occurs. As each basis test requires a large number of das-random numbers we may assume that the test variables

$$T_1, T_2, \ldots \qquad \text{are iid,} \tag{7}$$

regardless of the distribution of the random variables $B_1, B_2, \ldots$ (which may be dependent!) and the test strategy, i.e whether all das-random numbers are tested or not (see Sect. 5, Step 2). The *distribution* of the test variables $T_1, T_2, \ldots$, of course, depends essentially on that of $B_1, B_2, \ldots, B_n$.

The only task of decision rule (A) is to detect an eventual total breakdown of the noise source. The probability that (A) causes a noise alarm is absolutely negligible unless the random noise source has indeed totally broken down. We hence restrict our attention to decision rules (B) and (C). First, we first derive

a formula to calculate the probability p_{st} for a stop of a test suite. Recall that the history values $\widetilde{H}_1, \widetilde{H}_2, \ldots$ and the parameters v and w are multiples of 2^{-c}.

Let $\widetilde{Y}_j$ denote the largest integer for which $\widetilde{T}_{j-\widetilde{Y}_j+1}, \widetilde{T}_{j-\widetilde{Y}_j+2}, \ldots, \widetilde{T}_j \notin [t,u]$. (Especially, $\widetilde{Y}_j := 0$ if $T_j \in [t,u]$.) We interpret the numbers $\widetilde{Y}_1, \widetilde{Y}_2, \ldots$ as realizations of random variables $Y_1, Y_2, \ldots$. Due to (7) the random vectors $(H_0, Y_0 := 0), (H_1, Y_1), \ldots$ form a homogeneous Markov chain on the infinite state space $\{\ldots, -2^{-c}, 0, 2^{-c}, 2 \cdot 2^{-c} \ldots\} \times \{0, 1, \ldots\}$. Therefrom we derive a homogeneous Markov chain $Z_0, Z_1, \ldots$ on the *finite* state space

$$\Omega = \big\{(h,y) \mid h \in [v,w], h \text{ is a multiple of } 2^{-c}, 0 \le y < k\big\} \cup \{\infty\}. \tag{8}$$

In particular, Z_j attains the state ∞ if $(1-\beta)H_{m-1} + \beta T_m \notin [v,w]$ or $Y_m = k$ for any $m \le j$ whereas $Z_j := (H_j, Y_j)$ else. That is, Z_j attains the state ∞ if the test suite has been stopped till time step j. Especially, ∞ is an absorbing state. (For the mathematical background of finite Markov chains the interested reader is referred to [8].)

Next, we determine the transition matrix $Q = (q_{\omega_1,\omega_2})_{\omega_1,\omega_2 \in \Omega}$ We point out that $(\widetilde{H}_{j-1}, \widetilde{H}_j) = (h,h')$ for $h,h' \in [v,w]$ iff $(1-\beta)h + \beta\widetilde{T}_j \in [h' - 2^{-c-1}, h' + 2^{-c-1})$, or equivalently, iff $\widetilde{T}_j \in \beta^{-1}\big[h' - 2^{-c-1} - (1-\beta)h, h' + 2^{-c-1} - (1-\beta)h\big)$. Elementary but careful considerations yield the transition matrix

$$\begin{array}{l} Q = (q_{\omega_1,\omega_2})_{\omega_1,\omega_2 \in \Omega} \quad \text{with transition probabilities} \quad q_{\omega_1,\omega_2} = \\ \left\{\begin{array}{ll} \mathrm{Prob}\,(\mathrm{T_j} \in \mathrm{A_{h,h'}} \cap (\mathbb{R} \setminus \mathrm{C})) & \text{if } \omega_1 = (h,y), \omega_2 = (h', y+1) \text{ and } y < k-1 \\ \mathrm{Prob}\,(\mathrm{T_j} \in \mathrm{A_{h,h'}} \cap \mathrm{C}) & \text{if } \omega_1 = (h,y), \omega_2 = (h',0) \\ \mathrm{Prob}\,(\mathrm{T_j} \in \mathbb{R} \setminus \mathrm{D_h}) & \text{if } \omega_1 = (h,y), \omega_2 = \infty \text{ and } y < k-1 \\ \mathrm{Prob}\,(\mathrm{T_j} \in (\mathbb{R} \setminus \mathrm{D_h}) \cup (\mathbb{R} \setminus \mathrm{C})) & \text{if } \omega_1 = (h,k-1), \omega_2 = \infty \\ 1 & \text{if } \omega_1 = \omega_2 = \infty \\ 0 & \text{else} \end{array}\right. \end{array} \tag{9}$$

where we used the abbreviations

$$\begin{aligned} A_{h,h'} &:= \beta^{-1}[h' - 2^{-c-1} - (1-\beta)h, h' + 2^{-c-1} - (1-\beta)h)\ , \\ C &:= [t - 2^{-c-1}, u + 2^{-c-1}) \quad \text{and} \\ D_h &:= \beta^{-1}[v - 2^{-c-1} - (1-\beta)h, w + 2^{-c-1} - (1-\beta)h)\ . \end{aligned}$$

Remark 2. If the basis test value $\widetilde{T}_j$ is rounded to a multiple of 2^{-c} before it is "mixed" with $\widetilde{H}_{j-1}$ (cf. end of Sect. 5) in (9) the terms "Prob($\mathrm{T_j} \in \ldots$)" should read "Prob($round(T_j) \in \ldots$)" where $round(\cdot)$ temporarily denotes the round-off function.

Now let $\boldsymbol{v}_{(\omega)}$ denote the column vector with $|\Omega|$ components which are all zero except the component indexed by ω which equals 1. As $Z_0 = (H_0, Y_0) = (E_0(T), 0)$ we obtain

$$p_{\mathrm{st}} := \mathrm{Prob}(\text{test suite is stopped}) = \mathrm{Prob}(\mathrm{Z_N} = \infty) = \boldsymbol{v}^{\mathrm{t}}_{(\mathrm{E_0(T)},0)} \mathrm{Q^N} \boldsymbol{v}_{(\infty)}. \tag{10}$$

The probability that x particular test suites are stopped is $1-\sum_{i=1}^{x}(1-p_{\mathrm{st}})p_{\mathrm{st}}^{i-1} = 1-(1-p_{\mathrm{st}}^{x}) = p_{\mathrm{st}}^{x}$. Wald's equation ([5], 50) hence implies

$$E(\#\textit{test suites per noise alarm}) = \frac{\sum_{i=1}^{x} i(1-p_{\mathrm{st}})p_{\mathrm{st}}^{i-1} + xp_{\mathrm{st}}^{x}}{p_{\mathrm{st}}^{x}} \qquad (11)$$

In the following we denote the distribution of $B_1, B_2, \ldots, B_n$ with $\nu[n]$ while $F_{\nu[n]}(\cdot)$ denotes the cumulative distribution function of the test variable T_i if the basis test is applied on $B_1, \ldots, B_n$. Especially, $\mu[n]$ means that the B_j are iid and equidistributed on Ω_{das} (null hypothesis).

To initialize the transition matrix Q we have to know the cumulative distribution function of T_i. What's the difference to the situation in Example 2? Also there the knowledge of $F_{\nu[n]}(65.0)$ would have solved the main problems. In particular, one could check whether (R2) and (R3) are fulfilled. As already mentioned in Sect. 4 even for $\mu[n]$ the relative error between the exact cumulative distribution $F_{\mu[n]}(\cdot)$ of T_i and that of the χ^2_{15}-distribution is very large at the (extreme) tails of both distributions. In the online test procedure described in Sect. 5, however, the factor β is small and thus even an extremely large single basis test value $\widetilde{T}_j$ will not influence the history variable $\widetilde{H}_j$ considerably unless a total breakdown of the noise source has just occurred. For decision rule (C) the totality of all basis test values up to this moment is essential while the probability that decision rule (B) stops the actual test suite depends essentially on $1-(F_{\nu[n]}(u)-F_{\nu[n]}(t))$ and, of course, on k. We recommend to choose t and u that for the tolerable distributions $\nu[n]$ this probability is $\geq 10^{-3}$. In particular, unlike as in Example 2, for $\nu[n]=\mu[n]$ the deviation the approximate cumulative distribution function of T_i from $F_{\mu[n]}$ will not influence p_{st} considerably.

For general $\nu[n]$ we may approximate $F_{\nu[n]}$ by an empirical cumulative distribution function $F_{\nu[n]emp}$ which we derive with a stochastic simulation (see, e.g. [9], [5]). For this, we need a fast pseudorandom number generator with good statistical properties. (Unpredictability of the pseudorandom numbers is irrelevant in this context.) A sound candidate is, for example, the recursive algorithm

$$x_{n+1} \equiv ax_n + 1 \pmod{2^{64}}, \qquad \text{with} \quad a \equiv 1 \pmod 4, \quad a > 2^{48}, \qquad (12)$$

a so-called *linear congruential generator*. Setting $v_j := x_j 2^{-64}$ yields a sequence of standard random numbers. The standard random numbers behave similarly as realizations of iid random variables $V_1, V_2, \ldots$ which are equidistributed on the unit interval $[0,1)$. From the standard random numbers one derives a sequence $\widetilde{B}'_1, \widetilde{B}'_2, \ldots$ which is viewed as a realization of $B_1, B_2, \ldots$. (Example: Let the B_j be iid binary random variables with $\mathrm{Prob}(\mathrm{B_j}=1)=0.48$. Then we set $\widetilde{B}'_j := 1$ if $v_j \leq 0.48$ and $\widetilde{B}'_j := 0$ else.) We apply the basis test to $\widetilde{B}'_1, \ldots, \widetilde{B}'_n$ and compute the respective basis test value $\widetilde{T}'_1$. Repeating this process K times ($K \geq 10^5$) we obtain an empirical cumulative distribution function $F_{\nu[n]emp}$ which we use for the initialization of the transition matrix Q. Due to Glivenko-Cantelli's theorem ([6], 145) the absolute value $\sup_{x\in\mathbb{R}} |F_{\nu[n]}(x) - F_{\nu[n]emp}(x)|$ should be small

which is essential for decision criterion (C). Concerning decision rule (B), the relative error $|(1 - F_{\nu[n]}(u) + F_{\nu[n]}(t)) - (1 - F_{\nu[n]emp}(u) + F_{\nu[n]emp}(t))|/(1 - F_{\nu[n]emp}(u) + F_{\nu[n]emp}(t))$ should be small as $1 - F_{\mu[n]}(u) - F_{\mu[n]}(t) \geq 10^{-3}$ for typical choices of t and u. To obtain a reliable approximation of $1-F_{\nu[n]}(65.0)$ for the χ^2-square test in Example 2, however, the parameter K had to be gigantic.

Remark 3. (i) Of course, $F_{\nu[n]emp}$ is not exact. In principle, this could cause a bad approximation of the exact probability p_{st}. We gave reasons why this should not be the case. Moreover, stochastic simulations support this opinion. If the empirical distribution was derived twice ($K = 10^6$) for the same distribution $\nu[n]$ the respective p_{st}-values usually differed less than 1 per cent from their arithmetical mean. This shows that the derived results are stable. In (ii) we give a formal argumentation that decision rule (B) amplifies small relative errors no more than by factor k.
(ii) The probability that at least k consecutive test values $\widetilde{T}_1, \widetilde{T}_2, \ldots, \widetilde{T}_N$ lie outside $[t, u]$ is about $1 - (1 - p^k)^{N(1-p)}$ where p temporarily stands for the probability $\mathrm{Prob}(\mathrm{T_i} \notin [\mathrm{t}, \mathrm{u}])$ (or $\mathrm{Prob}(round(T_i) \notin [t, u])$, resp.; cf. Remark 2) If p' denotes an approximation of p the relative error equals $|(1-(1-p^k)^{N(1-p)}) - (1-(1-p'^k)^{N(1-p')})|/(1-(1-p'^k)^{N(1-p')})$. If $(Np^k)^2, (Np'^k)^2 \ll 1$ the relative error is about $|(1-p)p^k - (1-p')p'^k|/(1-p')p'^k$. If additionally $p \approx p'$ (which is likely, for example, if $p \geq 10^{-3}$ and $K \geq 10^5$) this term further simplifies to $k|p - p'|/p'$.

8 Examples

In Sect. 8 we discuss two examples. Especially, Example 3 provides an appropriate solution for Example 2. The effect of the particular parameters is explained in Sect. 9.

Example 3. We consider the same situation as in Example 2 (cf. Sect. 3). Due to the construction of the noise source it may be assumed that the random variables $B_1, B_2, \ldots$ are iid but not necessarily equidistributed (cf. Remark 4(iii)). Extensive statistical investigations of TRNG prototypes have confirmed this hypothesis. For the intended applications it is absolutely acceptable ("tolerable") if $\mathrm{Prob}(\mathrm{B_j} = 1) \in [0.49, 0.51]$. Otherwise, a noise alarm should occur sooner or later, depending on the "degree" of the statistical weaknesses of the das-random numbers. If $\mathrm{Prob}(\mathrm{B_j} = 1) < 0.475$ or $\mathrm{Prob}(\mathrm{B_j} = 1) > 0.525$, however, a noise alarm should occur soon. Recall that per TRNG about 530000 basis tests are performed per year (cf. Example 2).

Proposed solution: Basis test: χ^2-test on 128 four-bit blocks (i.e. $n = 512$). Further, we use the parameter set $N = 512$, $\beta = 1/64$ (i.e. $b = 6$), $c = 5$, $r = 0.0$, $s = 200.0$, $k = 3$, $t = 0.0$, $u = 26.75$, $v = 13.0$, $w = 17.0$, $x = 3$.

The values in Table 1 were derived on basis of empirical distribution functions as described in Sect. 7 ($K = 10^6$). The right-hand column of Table 1 gives the expected number of noise alarms per year. In particular, if $\mathrm{Prob}(\mathrm{B_j} = 1) \notin [0.475, 0.525]$ a noise alarm will occur after a few test suites.

Table 1. Example 3: Expected number of noise alarms per year

$\mathrm{Prob}(\mathrm{B_j} = 1)$	p_{st}	$E\left(\frac{\#\text{ noise alarms}}{\text{year}}\right)$
0.500	0.0162	0.004
0.495	0.0184	0.006
0.490	0.0289	0.024
0.485	0.0745	0.396
0.480	0.2790	16.6
0.475	0.7470	

Table 2. Example 4: Expected number of noise alarms per year

$\mathrm{Prob}(\mathrm{B_j} = 1)$	p_{st}	$E\left(\frac{\#\text{ noise alarms}}{\text{year}}\right)$
0.500	0.0151	0.00005
0.495	0.0180	0.00011
0.490	0.0349	0.0015
0.485	0.1096	0.1332
0.480	0.3866	14.5
0.475	0.8501	

Remark 4. (i) The calculation of the basis test values requires no more than integer multiplication and addition and, finally, a division by $8 = 2^3$.
(ii) For $c = 6$ instead of $c = 5$ the p_{st}-values are some percent larger as the Markov process $Z_0, Z_1, \ldots$ is less "inert" (cf. Sect. 9).
(iii) For simplicity, in Examples 3 and 4 we assume that the random variables $B_1, B_2, \ldots$ are iid. This, however, need not be the case for all types of random noise sources. We point out that dependent random variables B_j can be handled in the same way as iid ones. (Numerical example: Let the random variables $B_1, B_2, \ldots$ be Markovian with $\mathrm{Prob}(\mathrm{B_{j+1}} = 1 \mid \mathrm{B_j} = 0) = 0.490$ and $\mathrm{Prob}(\mathrm{B_{j+1}} = 0 \mid \mathrm{B_j} = 1) = 0.490$. Using the same parameters as in Example 3 yields $p_{\mathrm{st}} \approx 0.0243$, and hence about 0.014 noise alarms are expected per year.) Of course, if we drop the assumption that the B_j are iid we usually have to consider much more distributions than listed in Table 1.

Example 4. We consider the same situation as in Example 3. However, due to the intended application the expected number of noise alarms per year must not larger than 0.0015 if the TRNG produces appropriate das random numbers. (For example, the noise source could be part of a smart card which is used by customers to execute e-commerce applications. If the TRNG causes a noise alarm the smart card denies further service and has to be replaced by a new one.)

Proposed solution: Basis test: χ^2-test on 128 four-bit blocks (i.e. $n = 512$). Further, we use the parameter set $N = 512$, $\beta = 1/64$ (i.e. $b = 6$), $c = 5$, $r = 0.0$, $s = 200.0$, $k = 4$, $t = 0.0$, $u = 24.0$, $v = 13.125$, $w = 16.875$, $x = 4$.

Remark 5. Our online test procedure enables "controlled" testing. If "sound" TRNGs generate das random numbers which themselves are appropriate for the intended applications and if the mathematical model of the physical noise source and thus that of the das random numbers is reliable (i.e., if we can be sure that the (eventually TRNG-specific) distribution of the das random numbers is contained in the assumed class of distributions) this usually makes a strong, data-compressing (throughput-reducing!) mathematical follow-up treatment dispensable. If also the reduced data-rate is sufficiently large for the intended applications, however, a data-compressing follow-up treatment may be used as an additional security mechanism, even if this may not actually be necessary.

9 Fine-Tuning of the Parameter Set

If the basis test and the parameter set have been chosen suitably the online test procedure should perfectly meet the particular requirements of the intended applications. In Sect. 7 we described how to compute p_{st} and the expected number of test suites until a noise alarm occurs. However, as the computation of p_{st} is time-consuming we cannot try thousands of randomly chosen parameter sets. Below, we briefly describe the effect of particular parameters.

n: Unless it is too small the sample size n usually does not influence the distribution of the basis test variables T_j if $\nu[n] = \mu[n]$. If $\nu[n] \neq \mu[n]$, however, increasing n often implies higher rejection rates. Example: Let $\widetilde{T} := 2(\widetilde{B}_1 + \cdots + \widetilde{B}_n - 0.5n)/\sqrt{n}$ which merely considers the number of ones within the sample. If the B_j are iid with $\mathrm{Prob}(\mathrm{B_j} = 1) = \mathrm{p}$ the central limit theorem implies $\mathrm{Prob}(|\mathrm{T}| > \alpha) = 1 - \Phi\left(\alpha/\sqrt{4\mathrm{p}(1-\mathrm{p})} - \sqrt{\mathrm{n}}(\mathrm{p}-0.5)/\sqrt{\mathrm{p}(1-\mathrm{p})}\right) + \Phi\left(-\alpha/\sqrt{4\mathrm{p}(1-\mathrm{p})} - \sqrt{\mathrm{n}}(\mathrm{p}-0.5)/\sqrt{\mathrm{p}(1-\mathrm{p})}\right)$ where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution. If $\alpha = 2.575$ and $p = 0.48$, for example, for $n = 128$ (resp., for $n = 512$) we obtain the probability 0.018 (resp., 0.048). If $p = 0.5$, however, this probability equals 0.01 for both, $n = 128$ and $n = 512$.

N: Due to (10) it is reasonable to choose a power of 2 as this minimizes the number of matrix multiplications and thus the computation time. Furthe, it avoids unnecessary round-off errors when computing the probability p_{st}.

x: For small p_{st} equation (11) is essentially determined by the term p_{st}^{-x}. Thus, for different p_{st}-values increasing the parameter x amplifies the ratio of the expected number of test suites till the first noise alarm occurs. (Example: If $p_{\mathrm{st};1} = 2p_{\mathrm{st};2}$ it is $p_{\mathrm{st};1}^{-3} = p_{\mathrm{st};2}^{-3}/8$ but $p_{\mathrm{st};1}^{-4} = p_{\mathrm{st};2}^{-4}/16$.) This effect can be used to "separate" the tolerable from the non-tolerable distributions.

β: The smaller $\beta := 2^{-b}$ the smaller is the influence of single basis test values on the history values $\widetilde{H}_1, \widetilde{H}_2, \ldots$.

c: The history variables $H_0, H_1, \ldots$ may be interpreted as a "weighted" random walk on $[v, w] \cap \{\ldots, -2^{-c}, 0, 2^{-c}, 2{\cdot}2^{-c}, \ldots\}$ with absorbing state ∞. The smaller c the more "inert" is this random walk and hence the smaller is p_{st}. In particular, $\widetilde{H}_j \neq \widetilde{H}_{j-1}$ iff $(1-\beta)\widetilde{H}_{j-1} + \beta\widetilde{T}_j \notin \widetilde{H}_{j-1} + [-2^{-c-1}, 2^{-c-1})$ iff $\widetilde{T}_j \notin \widetilde{H}_{j-1} + \beta^{-1}[-2^{-c-1}, 2^{-c-1})$. We recommend to choose $b, c \in \{5, 6\}$. (Recall, however, that the transition matrix Q has $|\Omega|^2 = [k(2^c(v-w)+1)+1]^2$ entries.).

10 Conclusions and Final Remarks

In this paper we proposed a new online test, or more precisely, a new online test procedure for which it is *practically feasible* to determine the expected number of noise alarms within a time interval, even if the tested random numbers are not independent and equidistributed. The system designer can vary a whole parameter set and hence can fit the test to the very special requirements of the intended applications. In particular, this makes data-compressing (i.e. throughput-reducing) mathematical follow-up treatments in many cases dispensable. Compared with "ordinary" online tests the proposed online test procedure does only need a little more memory, some additional lines of code and slightly more running time.

References

1. AIS 20: Functionality Classes and Evaluation Methodology for Deterministic Random Number Generators. (English Translation, mandatory if a German security certificate is applied for). (December 1999). www.bsi.bund.de/aufgaben/ii/zert/jil_ais/ais20e.pdf
2. V. Bagini and M. Bucci: A Design of Reliable True Number Generators for Cryptographic Applications. In: Ç.K. Koç and C. Paar (eds.): Cryptographic Hardware and Embedded Systems. First International Workshop, CHES '99. Springer, Lecture Notes in Computer Science, Vol. **1717**, Berlin (1999), 204–218.
3. J.-S. Coron: On the Security of Random Sources. In: H. Imai and Y. Zheng (eds.): Public Key Cryptography. Second International Workshop on Practice and Theory in Public Key Cryptography, PKC '99. Springer, Lecture Notes in Computer Science, Vol. **1560**, Berlin (1999), 29–42.
4. J.-C. Coron, D. Naccache: An Accurate Evaluation of Maurer's Universal Test. In: S. Tavares and H. Meijer (eds.): Selected Areas in Cryptography '98, SAC '98. Springer, Lecture Notes in Computer Science, Vol. **1556**, Berlin (1999), 57–71.
5. L. Devroye: Non-Uniform Random Variate Generation. Springer, New York (1986).
6. P. Gänssler und W. Stute: Wahrscheinlichkeitstheorie. Springer, Berlin (1977).
7. G.K. Kanji: 100 Statistical Tests. Sage Publications, London (1995).
8. J.G. Kemeny and J.L. Snell: Finite Markov Chains. D. Van Nostrand, New York (1960).
9. D.E. Knuth: The Art of Computer Programming. Vol. 2, Addison-Wesley, London (1981).

10. D.P. Maher and R.J. Rance: Random Number Generators founded on Signal and Information Theory. In: Ç.K. Koç and C. Paar (eds.): Cryptographic Hardware and Embedded Systems. First International Workshop, CHES '99. Springer, Lecture Notes in Computer Science, Vol. **1717**, Berlin (1999), 219–230.
11. U. Maurer: A Universal Statistical Test for Random Bit Generators. J. Crypt. **5** (1992), 89–105.
12. NIST Special Publication 800–22: A Statistical Test Suite for Random and Pseudorandom Numbers. (December 2000).