Bundesamt für Sicherheit in der Informationstechnik

Application Notes and Interpretation of the Scheme (AIS)

AIS 20, Version 1

Date:	**2 December,1999**
Status:	**Mandatory**
Subject:	**Functionality classes and evaluation methodology for deterministic random number generators**
Publisher:	**Certification body of the BSI, Section II 2, as part of the certification scheme**

Offices:

No. 1: Godesberger Allee 183
Bonn-Hochkreuz
Tel.: +49 228 9582-0
Fax: +49 228 9582-400

No. 2: Mainzer Strasse 84
Bonn-Mehlem
Tel.: +49 228 9582-0
Fax: +49 228 9582-750

No. 3: Merianstrasse 100
Köln-Chorweiler
Tel.: +49 221 97959-0
Fax: +49 221 97959-250

No. 4: Gabrielweg 5
Swisttal-Heimerzheim
Tel.: +49 2254 9403-0
Fax: +49 2254 9403-40

No. 5: Kessenicher Strasse 216
Bonn-Dottendorf
Tel: +49 0228 9582-0
Fax: +49 228 9582-455

Bank account details: Bundeskasse Bonn at the Landeszentralbank Bonn (sort code 380 000 00) account no. 380 01 060 in favour of the BSI

Internet: http://www.bsi.bund.de/

1 Background to the AIS

Deterministic random number generators are incorporated within many products. In some instances they are used to generate challenges as part of an authentication process or to generate signature keys or encryption keys.

In certain cases, such as in the context of evaluation of a digital signature component, evaluation of authentication processes or the evaluation of components for key generation, analysis and assessment of the random number generator is necessary.

The evaluation manuals provide no information on this.

A thorough and uniform evaluation methodology is required for use by all bodies performing evaluations, certification or confirmation.

The document quoted in full below, "Functionality Classes and Evaluation Methodology for Deterministic Random Number Generators, version 2.0, 2 December 1999", presents an approach to describing and evaluating such a methodology.

The document incorporates the comments received from testing agencies and different sections of the BSI and has been restructured and expanded in a number of stages compared with the draft version (version 1.5) of 16 February 1999, and now constitutes the present version 2.0.

The AIS is now mandatory.

The document may be revised and extended further at a later date when more information is known or in the light of practical experience gained from using it.

Priv.-Doz. Dr. Werner Schindler
BSI, III 5

Functionality Classes and Evaluation Methodology for Deterministic Random Number Generators

Version 2.0
2 December 1999

Table of Contents

A. Rationale, Objectives and Overview of Contents

A.1 Rationale and objectives. Random numbers play an important role in many cryptographic applications. Yet there are still no uniform criteria for evaluating random number generators. This paper presents some evaluation criteria for deterministic random number generators. The basic idea is that the suitability of deterministic random number generators should be assessed with reference to the cryptographic applications in which they are used.

A.2 Overview of contents. Chapter B describes the subject of examination. Chapter C introduces four functionality classes (K1, K2, K3, K4) and explains the underlying rationale. Chapter D then describes the tasks of the evaluator, while the practical applicability of the functionality classes and the tasks of the evaluator are illustrated in Chapter E with the aid of various examples.

A.3 Note. The functionality classes defined below describe a set of hierarchical requirements which are expressed not at the abstract level of the generic headings used within ITSEC, but at the level of technical properties.

If an applicant who is aiming at a German IT security certificate uses a deterministic random number generator which cannot be assigned to any of the functionality classes K1 to K4, the application has to be agreed with the BSI.

B. Definitions and Notation

B.1 Mathematical description. A deterministic random number generator deterministically generates random numbers which depend solely on the seed (initial internal state). The 5-tuple $(S, R, \varphi, \psi, p_A)$ describes the logical structure of the generator and the seed selection process. These parameters have the following meaning:

S	the (finite) set of possible internal states of the random number generator
R	the set of possible output values (random numbers)
$\varphi: S \to S$	the state function
$\psi: S \to R$	the output function
p_A	a probability measure which describes the random distribution of the seed.

At step $n \geq 1$ the internal state is first updated using $s_n := \varphi(s_{n-1}) \in S$ and then a random number $r_n := \psi(s_n) \in R$ is calculated and output.

B.2 Note. Although, strictly speaking, the seed selection mechanism does not belong to the description of a deterministic random number generator, its probability distribution

may become extremely important when it comes to assessing the sequences of random number to be expected.

B.3 Notation and usage. The abbreviation DRNG will be used hereafter to refer to "deterministic random number generator". Where DRNG is used, it refers not to the technical realisation of the random number generator but to the defining 5-tuple (S, R, φ, ψ, p_A).

C. Functionality Classes

C.0 Rationale for introducing functionality classes. It is not possible for a deterministic random number generator to increase the total entropy of a random number sequence beyond the entropy of the seed by generating new random numbers. In this respect it contrasts with physical noise sources. Sequences of deterministically generated random numbers cannot therefore be truly "random". In the best case they may behave like truly random sequences with respect to specific criteria.

A number of attempts have been made in the literature to characterise "good" random number sequences. In the context of cryptographic applications, [FI140] (4.11.1) and [IEP] (G.4.5) deserve an especial mention. Whereas the first of these sources formulates statistical tests, the second uses the practical unpredictability of the random number sequences generated by a deterministic random number generator to characterise its suitability. It should be noted that in the last two decades statistical approaches to the evaluation of pseudorandom numbers have been closely related to stochastic simulations. However, this work has generally entailed different requirements being placed on the random numbers than those placed by cryptographic applications (see also chapter C, K2.e)).

On the other hand, different cryptographic applications also have different requirements regarding the random numbers which are necessary, and in any case deterministically generated random numbers can only behave like "truly" random numbers with respect to certain criteria. This suggests that the suitability of deterministic random number generators should be assessed according to the intended applications. Four downward-compatible functionality classes (K1, K2, K3, K4) are defined and discussed in detail below. The practical application of the criteria is demonstrated in Chapter E with the aid of six examples.

C.1 To be provided by the applicant:

(i) Statement of the intended functionality class (K1, K2, K3, K4) and the intended strength of mechanism.

(ii.a) Complete and comprehensible informal description of the deterministic random number generator.

(ii.b) Defining 5-tuple (S, R, φ, ψ, p_A)

(iii) An upper bound M for the maximum number of random numbers which can be generated with the DRNG over its entire life-cycle or until it is re-initialised with a new initial state $s_0 \in S$ selected according to p_A.

(iv) Clear description of how the seed is generated together with rationale as to how this will induce the distribution p_A.

\+ Additional information which is specified under item f) for the relevant functionality class.

C.2 Boundaries of the subject of examination DRNG

Subject of examination is the defining 5-tuple (S, R, φ, ψ, p_A). The assessment of the seed generation process, i.e. of the practical realisation of distribution p_A (statement of the applicant) is not part of the actual DRNG evaluation and is not covered in the evaluation criteria described below. Nevertheless, the applicant must explain clearly how the seed is generated (C.1(iv)) and explain why this produces distribution p_A (see also example E.7).

C.3 General note on the specification of functionality classes

(i) The strength of mechanism stated under item d) refers exclusively to logical attacks on the defining 5-tuple (S, R, φ, ψ, p_A). The strength of mechanism of the evaluation of the whole product naturally depends considerably on the technical implementation of the DRNG and its integration into the (complete) TOE (target of evaluation), as the evaluation of the whole product must also consider direct attacks on the cryptographic algorithms and protocols, the software or the operating system, as well as hardware-oriented attacks.

(ii) Item d) describes the class-specific properties. The details required for an evaluation in addition to C.1 (i)—(iv) are collected together in item f). The other items provide further information and give reasons for the selection and objectives of these requirements. Items i) and j) (see Chapter D) describe and explain the tasks of the evaluator.

(iii) The table below provides a summary of the relationship between the functionality classes, evaluation levels and the strength of the mechanism.

Class	Example (may be dependent on the choice of suitable parameters)	Minimum E level / Strength of mechanism
K1	E.1: counter E.2: linear congruential generator E.3: linear shift register E.4: recursive call of a block cipher E.5: counter with hash function E.6: RSA generator	E2 / low, medium E3 / high
K2	E.2-E.6	E2 / low, medium E3 / high
K3	E.4-E.6	E3 / medium, high
K4	E.6	E3 / medium, high

Class K1

a) Qualitative intuitive description of K1-specific requirements:

There should be a high probability that random vectors $(r_1,\ldots,r_c),(r_{c+1},\ldots,r_{2c}),\ldots,(r_{M-c+1},\ldots,r_M)$ formed from random numbers $r_1,r_2,\ldots$ are mutually different. The statistical properties of these vectors are unimportant. (The choice of parameters c and ε (and ultimately also of M) depends on the intended application.)

b) Possible applications:

--- Challenge-Response protocols (e.g. for use in a smart card-terminal authentication process).

c) Objective(s):

Protection against replay attacks

d) Requirements for K1 DRNGs:

(i) The probability that vectors $(r_1,\ldots,r_c),(r_{c+1},\ldots,r_{2c}),\ldots,(r_{M-c+1},\ldots,r_M)$ are mutually different should be at least $1-\varepsilon$.

If $\varepsilon = 0$, requirements for the strength of mechanism claim high are satiesfied. Otherwise, the following apply:

$M^2/c^2\varepsilon > 2^{52}$ and $\varepsilon < 2^{-16}$: the strength of mechanism claim high

$M^2/c^2\varepsilon > 2^{32}$ and $\varepsilon < 2^{-12}$: the strength of mechanism claim medium

$M^2/c^2\varepsilon > 2^{20}$: the strength of mechanism claim low

e) Rationale:

The set R of the random numbers capable of being generated is not identical for all DRNGs. Thus, examples E.1 to E.6 in Chapter D produce $\lceil \log_2(N) \rceil$, f, 1, (normally) 64, m or 1 bit random numbers. There can therefore be no universal criteria for the random number sequences $r_1,r_2,\ldots$ themselves, but, rather, random number vectors $(r_1,\ldots,r_c),(r_{c+1},\ldots,r_{2c}),\ldots$ must be investigated. The numeric values of c, M and ε are stated by the applicant. They are derived from the target application(s) and the bit width of the random numbers generated by the DRNG.

If one disregards the difficulties which would be encountered with a specific technical implementation, the logical effort involved in a replay attack against the defining 5-tuple (S, R, φ, ψ, p_A) depends strictly monotonically decreasing on ε, and, if c is fixed, strictly monotonically increasing on M. Specifically, for a successful replay attack in the scenario which is most favourable to him (i.e. with a sharp bound 1-ε), an adversary must on average observe 1/ε random vector sequences generated by different DRNGs and compare the individual members of each of these sequences internally with each other.

f) To be stated by the applicant in addition to C.1(i)--(iv):

(v) $c \in |N$ and $\varepsilon \in [0,1)$. (Confirmation for several parameter pairs (c, ε) is possible.)

(vi) Mathematical proof (if necessary, with plausible assumptions regarding a mathematical model - see also examples E.1-E.6), that requirement d)(i) is satisfied. (The mathematical proof is optional if $M\ |A| < 2^{32}$ with $A := \{s \in S \mid p_A(s) > 0\}$ or $10M/\varepsilon < 2^{32}$. See also K1.i) (ii.b) and (ii.c) in Chapter D.).

g) Explanations: ---

h) K1 DRNGs (examples):

E.1, E.2, E.3, E.4, E.5, E.6.

Class K2

a) Qualitative intuitive description of K2-specific requirements:

The random numbers generated possess similar statistical properties to random numbers which have been generated by an ideal random number generator.

b) Possible applications:

--- Stream ciphers which are controlled through a shift register bundle whose initial values are derived from secret long-term keys and a session key which is transmitted in plaintext at the beginning of the communication.

c) Objective(s):

Correlation attacks on cryptographic algorithms which are based on statistical weaknesses of the random numbers used (possibly as random keys) must be ruled out.

d) Requirements for K2 DRNGs:

--- The DRNG must belong to class K1 (downward compatibility).

(ii) Interpreted as a binary string, random number sequences $r_1,r_2,\ldots$ and their projections onto individual bits must pass statistical tests T1-T5 specified in Chapter F (see K2.i)).

The strength of mechanism corresponds to that of the K1-specific portion. Evaluation of the test results is independent of the strength of mechanism .

e) Rationale:

If one were to apply the tests specified in Chapter F to an ideal noise source, the probability of rejecting the null hypothesis for each individual test would be around 10^{-6}. In particular, the probability of an erroneous failure to recognise that a DRNG satisfied the K2 requirements (see K2.i), decision rule) would be less than $2.5 \cdot 10^{-6}$. Therefore halfway "reasonable" DRNGs should in practice always pass the statistical tests (see also comment D.2). Although they are not very powerful, the statistical tests should be strong enough to exclude known attacks on the cryptographic algorithms which are based on statistical weaknesses of random numbers.

"Basic components" of stochastic simulations are normally so-called standard random numbers which with respect to many of their statistical aspects behave like realisations of independent random variables uniformly distributed on the interval [0,1). In nearly all practical problem situations, only the most significant bits of the standard random numbers generated have any material effect on the simulation results. Generally this does not apply to cryptographic applications. Therefore one should strive for similar statistical behaviour of the individual bits. The tests described under K2.i (iii.b) should detect any weaknesses in individual bits (see also example E.2).

f) To be stated by the applicant in addition to C.1(i)--(iv) and K1.f)(v)-(vi): ---

g) Explanations: ---

h) K2 DRNGs (examples):

The K2-specific requirements are demonstrated through statistical tests, hence case-specific theoretical considerations are not really necessary. Nevertheless, this is

covered in Chapter E. One may assume that the examples E.2-E.6 will pass the required statistical tests, assuming a suitable choice of parameters.

Class K3

a) Qualitative intuitive description of K3-specific requirements:

It is practically impossible for an adversary to work out or guess the numbers which precede or follow a random number subsequence $r_i,r_{i+1},\ldots,r_{i+j}$ or to work out or guess an internal state. The adversary's assumed attack potential depends on the strength of mechanism.

b) Possible applications:

--- generation of pairs of signature keys

--- generation of DSS signatures (private key x or random number k; see [FI186])

--- generation of session keys for symmetric cryptographic mechanisms

--- pseudorandom padding bits (see also [RSA], section 8.1)

--- zero-knowledge proofs

c) Objective(s):

Protection against reconstruction of old random numbers and prediction of future random numbers from a known subsequence.

d) Requirements for K3 DRNGs:

--- The DRNG must belong to class K2 (downward compatibility).

(iii) For the strength of mechanism claim high, $H(p_A) \geq 80$; for strength of mechanism strength medium, $H(p_A) \geq 48$. (The entropy of p_A is given by $H(p_A) = -\Sigma_{s \in S}\ p_A(s)\ \log_2(p_A(s))$.)

(iv) It must be practically impossible for an adversary to work out predecessor r_{i-1} or successor r_{i+j+1} of a subsequence $r_i,r_{i+1},\ldots,r_{i+j}$ which is known to him ($i+j \leq M$). The adversary's assumed attack potential depends here on the strength of mechanism. Even using the most advanced know-how currently available, the probability of guessing (realised by a reasonable partial exhaustion) may at most be negligibly greater than if the subsequence were not known. It is assumed that the adversary knows the defining 5-tuple. However, he does not know any of the internal states $s_0,s_{1,\ldots,}\ s_M$.

Under the strength of mechanism claim "high", the adversary is assumed to have the most advanced know-how currently publicly available, the currently most powerful technology without limitations and a period of several years over which to perpetrate his attack. Under the strength of mechanism claim "medium" the adversary is assumed to have medium attack potential within the meaning of ITSEM, Appendix 6.C (see also

g) and example E.4). It is not possible to evaluate the K3-specific properties with the strength of mechanism claim "low".

e) Rationale:

In the definition of the attack potential to be considered, the requirements specified in d)(iv) for the strength of mechanism claim high go considerably beyond the corresponding definition in ITSEM, Appendix 6.C. The one-way properties required in d(iv) move K3 DRNGs close to cryptographic mechanisms (e.g. hash functions). A special evaluation procedure is implicitly provided for this in ITSEC 3.23 and ITSEM 6.C.34.

It is especially important in connection with digital signatures and the generation of symmetric session keys for sensitive applications that the attack potential considered exceeds the ITSEM requirements. The generation of session keys for symmetric encryption mechanisms which themselves only possess the strength of mechanism claim medium constitutes an exception here.

The length restriction of random number subsequence $r_i,\ldots,r_{i+j}$, which is presumed to be known, follows naturally from C.1 (iii). Requirement d)(iv) simultaneously guarantees the security of each predecessor and each successor r_v (with $v \leq M$) of this subsequence, as nothing is presupposed apart from $i+j \leq M$. (If the subsequence which is presumed to be known is lengthened to $r_{v+1},\ldots\ r_{i+j}$ or $r_i\ldots,r_{v-1}$, then r_v directly precedes or follows it). Moreover d)(iv) also safeguards the internal states, since, when s_t is known, random numbers $r_t,r_{t+1},\ldots$ can be calculated easily. In view of d)(iii) and the K2 property, it should be practically impossible without knowledge of a random number subsequence to guess the internal state, individual random numbers or short subsequences since, apart from pathological exceptions, relatively short random number subsequences possess virtually all of the entropy of the seed already. ("Short" depends on the bit width of the random numbers.)

Formalisation of the K3 requirements, especially as regards guessing, demands maintaining a balancing act between mathematical exactness and the practical feasibility of verifying the criteria. Many publications on "hard core bits" make use of characterisations from complexity theory (e.g. see the overview article [La]) to define terms such as bit unpredictability (see also [ACGS], 196). The crucial disadvantage for the applications we envisage is that the definitions and conclusions refer not to a single DRNG but to a whole family of DRNGs, i.e. the results are asymptotic. A quantitative determination of the computational effort required or of the probability of guessing in specific particular cases, assuming the maximum attack potential is available, should be extremely difficult if not impossible. Instead, a pragmatic approach is adopted (cf i)), whereby verification of property d) (iv) is shifted to a related problem which is generally viewed as not practically feasible with the adversary's assumed attack potential (even though this normally cannot be formally proven).

f) To be stated by the applicant in addition to C.1(i)--(iv) and K1.f)(v)-(vi):

(vii) Mathematical proof (if necessary, with plausible assumptions regarding a mathematical model) that requirement d)(iv) is satisfied.

g) Explanations:

for d)(iv), f)(vii): Demonstration of property d)(iv) can consist of showing, if necessary with plausible assumptions, that working out r_{i-1} and r_{i+j+1} or specifying a guessing strategy which utilises knowledge of $r_i,\ldots,r_{i+j}$ are at least as difficult as a problem which is generally viewed as not practically feasible with the attack potential specified in d)(iv) (see examples E.4 and E.6).

h) K3-DRNGs (examples):

The strength of mechanism claim high: E.4 (for Enc = Triple-DES, IDEA), E.5, E.6;

The strength of mechanism claim medium: E.4 (for Enc = DES).

Class K4

a) Qualitative intuitive description of K4-specific requirements:

It is practically impossible for an adversary to work out or guess predecessor random numbers or predecessor internal states from knowledge of the internal state s_i. The adversary's assumed attack potential depends on the strength of mechanism.

b) Possible applications:

--- generation of pairs of signature keys

--- generation of DSS signatures (private key x or random number k; see [FI186])

--- generation of session keys for symmetric cryptographic mechanisms

--- pseudorandom padding bits (see also [RSA], section 8.1)

c) Objective(s):

Protection against reconstruction of old random numbers from a known internal state. (Scenario: adversary obtains possession of the technical implementation of the DRNG and is able to read out the internal state.)

d) Requirements for K4 DRNGs:

--- The DRNG must belong to class K3 (downward compatibility).

(v) It must be practically impossible for an adversary to work out the predecessor random number r_{i-1} from knowledge of the internal state s_i . The adversary's assumed attack potential depends here on the strength of mechanism. Even using the most advanced know-how currently available, the probability of guessing (realised by a reasonable partial exhaustion) may at most be negligibly greater than if s_i were not known. It is assumed that the adversary knows the defining 5-tuple.

Under the strength of mechanism claim "high", the adversary is assumed to have the most advanced know-how currently publicly available, the currently most powerful technology without limitations and a period of several years over which to perpetrate his attack. Under the strength of mechanism claim "medium" the attacker is assumed to have medium attack potential within the meaning of ITSEM, Appendix 6.C. It is not possible to evaluate the K4-specific property with the strength of mechanism claim "low".

e) Rationale: analogous to K3.e). It should simply be noted that requirement d)(v) does not protect only the direct predecessor of random number r_{i-1}, but every r_v and s_v where $v<i$ (rationale analogous to K3.e), paragraph 3). Requirement d(v) is a tougher version of the "backward property" contained in d(iv), as it is easy to calculate random numbers $r_i,...,r_{i+j}$ from the internal state s_i.

f) To be stated by the applicant in addition to C.1(i)--(iv), K1.f)(v)-(vi) and K3.f)(vii):

(vii) Mathematical proof (if necessary, with plausible assumptions regarding a mathematical model) that requirement d)(v) is satisfied.

g) Explanations:

See K3.g)

h) K4 DRNGs (examples):

E.6 (the strength of mechanism claim high)

C.4 Note. It is clear that with K3 or K4 DRNGs possessing the strength of mechanism claim high one is "on the safe side". On the other hand, if the DRNG is used to generate keys for an encryption algorithm which itself only possesses the strength of mechanism claim medium, then the strength of mechanism claim medium is obviously sufficient. Subject to the intended applications, it can be highly appropriate to use a K1 or K2 generator, as these normally require less computational effort and their implementation requires less code and hence less RAM than is the case with K3 or K4 generators. These aspects can be particularly relevant when the applications involve smart cards. It should also be borne in mind that, when implementing a K3 or K4 DRNG, to be consistent it is necessary to ensure (overall evaluation!) through appropriate protective measures (hardware, software, operating system) that the internal state of the DRNG (e.g. a secret cryptographic key) is reliably protected against unauthorised retrieval. This is not necessary for K1 and K2 DRNGs.

D. Evaluation Methodology

Chapter D describes how the evaluator should test the specific properties of a given functionality class. The sub-sections are numbered beginning with i).

D.0 Connection with the evaluation of the whole product. The manufacturer specifies the security functionality requirements in the security target. Where it is already appropriate in particular cases to specify the generation and usage of random numbers at this level of abstraction, the functionality class of the random number generator is stated with reference to the security function of the (complete) TOE. (Often the deterministic random number generator forms only a part of the product to be evaluated.) Any assumptions regarding the operational environment and the secure use of the TOE must be specified (e.g. the requirement for a suitable seed generation process).

The implementation of the deterministic random number generator must be specified in the low-level design. In certain cases this can affect the documentation concerning delivery and configuration, start-up and operation and the operational documentation.

In particular, as part of the analysis of suitability, a rationale must be provided as to why the method of seed generation is suitable (see also C1 (iv) and C.2). The seed generation process should likewise be covered during penetration testing of the TOE.

D.1 Scope and sequence of evaluation work

Subject of examination is the defining 5-tuple $(S, R, \varphi, \psi, p_A)$. The generation of the seed, i.e. the practical realisation of initial state p_A, is not part of the actual DRNG evaluation and is not covered in the evaluation criteria (see C.2). However, the actual evaluation is only carried out where the evaluator is satisfied from the applicant's line of reasoning that this seed generation process does induce distribution p_A. The practical realisation then plays no further role in the evaluation itself.

- The evaluator reviews the arguments provided by the applicant in the analysis of suitability with regard to realisation of the initial distribution p_A.
- The evaluator has to compare the defining 5-tuple $(S, R, \varphi, \psi, p_A)$ with the informal description of the deterministic random number generator (C.1(ii.a)) and to check it for consistency.
- The evaluator performs class-specific tasks which are specified under item i) of the relevant functionality class and explained under j).

Class K1 (continued)

i) Tasks of the evaluator

(ii.a) Verification of any mathematical proof f)(v) provided by the applicant that the DRNG submitted belongs to class K1.

(ii.b) If no mathematical proof of the K1 property is provided and at the same time M $|A| < 2^{32}$ with $A := \{s \in S \mid p_A(s) > 0\}$, then the K1 property is demonstrated by trying out all the permitted initial values $s_0 \in A$, so that vectors $(r_1,\ldots,r_c),\ldots,(r_{M-c+1},\ldots,r_M)$ are

formed in each case and the members of the series are checked for pairwise difference. The results are weighted according to p_A.

(ii.c) If it is not possible to verify the K1 property by means of (ii.a) or (ii.b), but $10M/\varepsilon < 2^{32}$ and the strength of mechanism is at least medium, then verification is performed using a stochastic simulation. For this purpose the evaluator generates initial states $s_{0;1},\ldots,s_{0;t} \in S$ with $t = \lfloor 10/\varepsilon \rfloor$ according to p_A in a (pseudo)random way. For each of these initial states, he calculates the vectors $(r_1,\ldots,r_c),\ldots,(r_{M-c+1},\ldots,r_M)$ and checks the members of the sequence for pairwise difference.

If duplicate random vectors occur on no more than one out of all t individual simulations, then the DRNG's K1 property is confirmed (with the parameters c, M, ε).

j) Notes re i):

Re i)(ii.b) and (ii.c): the evaluator must generate a maximum of 2^{32} random numbers.

Re i)(ii.c): the evaluator interprets the results of the t individual simulations as realisations t of independent, identically B(1, p)-distributed random variables with unknown p, whereby the result "1" corresponds to multiple occurrences of vectors. When confirming the K1 property, the only item of interest is whether $p \leq \varepsilon$ holds true. For this purpose, the evaluator carries out a statistical test with the null hypothesis H_0: $p > \varepsilon$ and the alternative hypothesis H_1: $p \leq \varepsilon$, and the null hypothesis is rejected if duplicate random vectors occur on less than two individual simulations. The probability of this event depends on the unknown probability p, with $q(p) := (1+\lambda_p)e^{-\lambda_p}$ with $\lambda_p = pt$ (Poisson approximation).

If $p > \varepsilon$, then the probability of erroneous confirmation of the K1 property is less than $q(\varepsilon) = 0.0005$. For $p < \varepsilon/128$, for example, the probability of erroneously rejecting a DRNG's K1 property is less than $q(\varepsilon/128) = 0.003$. (This means in reality that the applicant has to state a significantly higher ε than the (supposed) actual value in order to get the DRNG's K1 property successfully confirmed under test requirement (ii.c) and certified.)

Class K2 (continued)

i) Tasks of the evaluator

--- Verification of K1 property (see K1.i))

Let f refer to the width of the random numbers capable of being generated by the DRNG in binary representation (normally $f = \log_2 \lceil |R| \rceil$) and π_w to the projection to the w'th component.

(iii.a): The evaluator chooses an initial state $s_0 \in S$ according to p_A, generates random numbers $r_1, r_2, \ldots$, and interprets these as bit strings of fixed length. He applies tests T1-T4 described in Chapter F to the first 20,000 bits of this sequence, using the specified critical values. He also calculates the test statistics $Z_1,\ldots,Z_{5000}$ (see test T5 in Chapter

F), determines $\max_{\tau \le 5000}\{|Z_\tau - 2500|\}$ and chooses a τ_0 (at random if there are several candidates) for which this maximum is assumed. He then applies the autocorrelation test (test T5) with shift τ_0 and the critical values specified in Chapter F to the subsequence $b'_1 := b_{10001},\ldots,b'_{10000} := b_{20000}$.

(iii.b)(w) ($1 \le w \le f.$): The evaluator chooses an initial state $s_0 \in S$ according to p_A and generates random numbers $r_1,r_2,\ldots,r_{20000}$. He then applies the statistical tests T1-T4 described in Chapter F to the sequence of projections $\pi_w(r_1),\ldots,\pi_w(r_{20.000})$, using the specified critical values. He also calculates the test statistics $Z_1,\ldots,Z_{5000}$ (see test T5 in Chapter F), determines $\max_{\tau \le 5000}\{|Z_\tau - 2500|\}$ and chooses a τ_0 (at random if there are several candidates) for which this maximum is assumed. He then applies the autocorrelation test (test T5) with shift τ_0 and the critical values specified in Chapter F to the subsequence $b'_1 := b_{10001},\tau,b'_{10000} := b_{20000}$.

Decision rule

The evaluator performs test procedures (iii.a), (iii.b)(1), (iii.b)(2),…, (iii.b)(f), (iii.a),… in sequence until a total of 257 bit sequences have been generated and tested. If the DRNG (i.e. the 5-tuple (S, R, φ, ψ, p_A)) passes all the individual tests, then the DRNG is confirmed as satisfying d)(ii). If more than one individual test results in a rejection, the DRNG is deemed to have failed to satisfy requirement d)(ii).

If exactly one individual test resulted in a rejection, then the whole test procedure must be repeated, and only if the DRNG passes all the single tests this time is it confirmed as satisfying requirement d)(ii). A second repetition is not permitted.

The 5-tuple (S, R, φ, ψ, p_A) is confirmed as satisfying the requirements for class K2 membership if it defines a K1 DRNG and passes the K2-specific tests in accordance with the decision rule from i).

j) Notes re i):

The random numbers to be tested need not be generated using the target of evaluation itself. The evaluator can use a simulation program written by him for this purpose. (This is likely to considerably speed up the process of running through test procedures (iii.a) and (iii.b)(w).) The initial distribution p_A must be simulated in an appropriate manner.

Class K3 (continued)

i) Tasks of the evaluator

--- Verification of K2 properties (see K1.i), K2.i))

(iv) Verification of d)(iii)

Verification of the mathematical proof provided by the applicant that d)(iv) is satisfied.

When confirming K3 membership with the strength of mechanism claim high (medium), not only properties d)(iii) and d)(iv) must be evaluated with mechanism strength high (medium), but also property d)(i) (see K1.d)).

j) Notes re i):

Re i)(v): see K3.g)

Class K4 (continued)

i) Tasks of the evaluator

--- Verification of K3 property (see K1.i), K2.i), K3.i))

(vi) Verification of the mathematical proof provided by the applicant that d)(v) is satisfied.

When confirming K4 membership with the strength of mechanism claim high (medium), not only property d)(v) must be evaluated with the strength of mechanism claim high (medium), but also properties d)(i) (see K1.d)), d(iii) and d)(iv) (see K3.d)).

j) Notes re i):

Re i)(vi): see K3.g)

D.2 Note. Properties d(i) (apart from the evaluation possibility K1.h)(ii.c)), d(iii), d(iv) and d(v) are verified using theoretical proofs. Naturally, proofs are reproducible, i.e. d)(i), d(iii), d(iv) and d(v) are properties of the defining 5-tuple (S, R, φ, ψ, p_A). On the other hand, verification using a stochastic simulation (cf K1.i)(ii.c)) or statistical tests (property d)(ii), cf K2.i)(iii.a) and (iii.b)) is not reliably reproducible as the seed is randomly selected according to p_A. Therefore, at least property d)(ii) is not a property of the 5-tuple (S, R, φ, ψ, p_A) itself. This awkward state is mitigated by the facts that, firstly, it is highly improbable (cf K1.i)) that a DRNG will erroneously be confirmed as satisfying the requirements for class K1 on the basis of K1.h)(ii.c) and, secondly, that the probability of failing to recognise the d)(ii) property for "reasonable" DRNGs is even lower (cf K2.e)). In this way even that portion of the results of the DRNG evaluation which is obtained empirically is "quasi-reproducible", which is essential for the reliability and trustworthiness of the evaluation procedure.

D.3 Evaluation levels. Classes K1 and K2 can be evaluated up to the strength of mechanism claim "medium" from E2 if additional information is provided by the applicant. Otherwise, at least assurance level E3 is necessary, and in this case also additional information is required from the applicant.

E. Examples

In examples E.1 to E.6 several DRNG types are investigated with regard to the requirements specified for functionality classes K1 to K4. Although K2-specific property d)(ii) is checked using statistical tests, it is briefly covered below as well.

E.7 provides an example of the required rationale C.1(iv), as to how the seed generation induces distribution p_A.

E.0 Notation. Let $Z_N := \{0,1\ldots,N-1\}$ and μ_T the uniform distribution on the finite set T.

E.1 Example (counter). The DRNG is specified via the 5-tuple $(Z_N, Z_N, \varphi, \psi, \mu_{\{0\}})$ with state function $\varphi(j) := j+1 \pmod N$ and output function $\psi(j) := j$.

If $N > M$, then even for $\varepsilon = 0$ the K1 property is satisfied for every c. Obviously, even for $p_A = \mu_{Z_N}$ the counter fails already on the K2-specific tests.

E.2 Example (linear congruential generator)

Let $N := 2^d$ and $a < N$ with $a \equiv 1 \pmod 4$. Starting from an initial state $s_0 \in Z_N$, a series $s_1,s_2,\ldots$ is calculated recursively via $s_j = \varphi(s_{j-1}) := (as_{j-1}+1) \pmod N$. In step j, the f most significant bits ($f \le d$) are output as random number r_j, i.e. $r_j = \psi(s_j) := \lfloor s_j/2^{d-f} \rfloor$. This results in the defining 5-tuple $(Z_N, \{0,1\}^f, \varphi, \psi, \mu_{Z_N})$.

For f = d, equality of two random number vectors $(r_{ic+1},\ldots,r_{(i+1)c})$ and $(r_{jc+1},\ldots,r_{(j+1)c})$ would mean in particular $s_{ic+1} = r_{ic+1} = r_{jc+1} = s_{jc+1}$, which cannot occur for $M \le 2^d$ (period length of the DRNG).

For sufficiently large d (e.g. $d \ge 48$) the statistical behaviour of the standard random numbers $s_1/2^d, s_2/2^d,\ldots$ is similar to realisations of independent random variables which are uniformly distributed on the interval [0,1). If for small f (and M small compared with period length 2^d), random number sequences $r_1,r_2,\ldots r_M$ are interpreted as realisations of independent random variables uniformly distributed on $\{0,1\}^f$, then to a good approximation $P((r_1,\ldots,r_c),\ldots,(r_{M-c+1},\ldots,r_M)$ mutually different$) \approx e^{-M/c*(M/c-1)/2^{cf+1}}$ holds true for $M/c < 2^{cf/2}$ (birthday phenomenon!). If M and ε are "reasonable", the linear congruential generator thus satisfies property d)(i) with small f also. (For $M/c = 2^{16}$ and cf = 54, for example, the right-hand term $\approx 1-2^{-23}$.) If $f \approx d$, the linear congruence generator will not pass the K2-specific test, as the k-least significant bit is 2^k periodic.

E.3 Example (linear shift register)

Let $p: \{0,1\}^d \to \{0,1\}$, $p(x) := \sum_{j=0}^{d-1} a_j x^j$ denote the primitive generator polynomial of a linear shift register of length d. Let the initial state, i.e. the initial value of the shift register, be randomly (uniformly distributed) chosen from the set of non-zero d-tuples. The defining 5-tuple is given by $(\{0,1\}^d, \{0,1\}, \varphi, \psi, \mu_{S\setminus\{0\}})$, with $\varphi(b_{n-1},\ldots,b_{n+d-2}) := (b_n,\ldots,b_{n+d-2}, \; b_{n+d-1} := \sum_{j=0}^{d-1} a_j b_{n+d-2-j})$ and $\psi(b_n,\ldots,b_{n+d-1}) := b_n$.

As p is assumed to be primitive, besides the zero state only one further cycle of length 2^d-1 exists, i.e. exactly the set of all permitted initial states. If $c \geq d$ and $M \leq 2^d$-1, then (M/c) many c-tuples $(r_1,\ldots,r_c),(r_{c+1},\ldots,r_{2c}),\ldots, (r_{M-c+1},\ldots,r_M)$ produce different pairs for every $s_0 \in S\backslash\{0\}$, since equality of two c-tuples implies in particular the equality of their projections onto the first d components, i.e. the equality of the internal states. For $c \geq d$, the linear shift register also belongs to class K1, and with the optimal value $\varepsilon = 0$. If d is sufficiently large, it should also pass the K2-specific statistical tests.

If the feedback polynomial p is known, one needs around d random numbers to reconstruct the internal state of the shift register. (If p is unknown it can be determined from a known random number subsequence of length around 2d using the Berlekamp-Massey algorithm.) Linear shift registers therefore do not belong to class K3.

E.4 Example (recursive call of a block cipher). Let Enc denote a symmetric block cipher algorithm (e.g. DES, triple-DES, IDEA) with identical plaintext and cipher text space, and let S_B and S_K denote the plaintext space and the key space, respectively. Let the initial state $s_0 := (r_0,k) \in S_B \times S_K$ be chosen randomly (uniformly distributed). Key k remains constant throughout the entire random number generation process and is kept secret.

The DRNG is described by the 5-tuple $(S_B \times S_K, S_B, \varphi, \psi, \mu_{S_B \times S_K})$ with $\varphi: S_B \times S_K \rightarrow S_B \times S_K$, $s_n = (r_n,k) = \varphi(r_{n-1},k) := (\mathrm{Enc}(r_{n-1};k),k)$ and $\psi: S_B \times S_K \rightarrow S_B$, $\psi(r_n,k) := r_n$.

Random numbers $r_1,\ldots,r_M$ are mutually different iff the initial value $r_0 \in S_B$ is in a cycle of length $\leq M$ with respect to the permutation $r \rightarrow \mathrm{Enc}(r; k)$. For Enc = DES, Enc = triple-DES or Enc = IDEA one may assume on the basis of the latest research that the random variable $k \rightarrow \mathrm{Enc}(\bullet;k)$ (random key selection!) has similar properties as a random permutation. With this model assumption, it is not difficult to see that the cycle length of r_0 (random selection!) is uniformly distributed on the set $\{1,\ldots,|S_B|\}$. Hence $P((r_1,\ldots,r_c),\ldots,(r_{M-c+1},\ldots,r_M)$ pairwise disjoint$) \geq P(r_1,\ldots,r_M$ pairwise disjoint$) \approx 1-(M-1)/|S_B|$, so that for $\varepsilon \geq (M-1)/|S_B|$ the DRNG should be considered to belong to class K1. As no statistical oddities are known of block cipher algorithms generally viewed as strong, one may assume that this DRNG also passes the K2-specific tests.

If one could determine from knowledge of a subsequence $r_i,\ldots,r_{i+j}$ its predecessor r_{i-1} or even the internal state, i.e. especially the secret key k, this would constitute a successful (special) known plaintext attack on the block cipher algorithm Enc, namely on plaintext or key, respectively. (Note: such an attack would be at least as difficult to carry out as a chosen plaintext attack against Enc.) By analogy, the task of determining the successor r_{i+j+1} constitutes a special known plaintext attack against the encryption function Enc^{-1}. For Enc = DES, triple-DES or IDEA, the DRNG fulfils the K3-specific properties d)(iii) and d)(iv), as no guessing strategies are known for guessing unknown plaintext or key bits with a probability greater than 0.5. Triple-DES and IDEA possess the strength of mechanism high, whereas simple DES has only the strength of

mechanism medium (exhaustive key search!). This DRNG type does not belong to class K4 (encryption!).

E.5 Example (counter with hash function)

Let $S = Z_N$ with $N \geq 2^{200}$, $\varphi: Z_N \rightarrow Z_N$, $\varphi(j) := j+1 \pmod N$, and let the initial value $s_0 \in S$ be random, i.e. chosen uniformly distributed on Z_N and kept secret.

In addition, let $H: \{0,1\}^{|N} \rightarrow \{0,1\}^m$ be a hash function viewed as suitable (e.g. RIPEMD-160). Then $(Z_N,\{0,1\}^m, \varphi, H, \mu_{Z_N})$ defines a DRNG.

If one interprets the hash function as a random variable over Z_N with values in $\{0,1\}^m$, and if one further assumes that sequences H(i),H(i+1),… possess statistical properties which are similar to realisations of independent random variables which are uniformly distributed on $\{0,1\}^m$, then the K1 property is verified as in E.2 for small f. ($P(r_1,r_2,\ldots,r_M$ are mutually different$) \approx e^{-M*(M-1)/2|H(S)|}$.) The DRNG satisfies the K3-specific properties (one-way property of the hash function) with the strength of mechanism claim high, although it clearly is not a K4 DRNG.

E.6 Example (RSA generator): (see also [La], 131)

Let p and q be prime numbers such that $p \neq q$, $N := pq$ and $e \in \{1,\ldots,\phi(N)\}$ with $ggT(e, \phi(N)) = 1$, whereby ϕ denotes the Euler function. The prime factors p and q are suitably chosen (e.g. in accordance with the recommendations of the catalogue of measures for the Digital Signature Act (SigG)), kept secret and deleted after calculation of N(known) and selection of e (not known). Let the initial state $s_0 = (t_0,e) \in Z_N \times B := Z_N \times \{0<y<\phi(N) \mid ggT(y,\phi(N))=1\}$ be chosen at random (uniformly distributed).

The RSA generator is described by the 5-tuple $(Z_N \times B, \{0,1\}, \varphi, \psi, \mu_{ZN \times B})$, whereby the mappings φ and ψ are given by $\varphi: Z_N \times B \rightarrow Z_N \times B$, $\varphi(t_{n-1},e) := (t_{n-1}^{\,e} \pmod N, e)$ and $\psi: Z_N \times B \rightarrow \{0,1\}$, $\psi(t_n,e) := t_n \pmod 2$.

To assess RSA generators, we use asymptotic results which are available in the literature, whereby we assume that the asymptotic behaviour takes effect at the order of magnitude of the chosen modulus N. In particular it is assumed that it is difficult, i.e. practically infeasible, to invert $x \rightarrow x^e \pmod N$ when e is known but $d := e^{-1} \pmod{\phi(N)}$ is not. (If this were possible, one would then be able to generate valid signatures solely from knowledge of the public key.) Using our terminology, it is practically impossible to work out or guess s_{i-1} from s_i.

If it were possible to guess the least significant bit in t_{i-1}, i.e. r_{i-1}, from a knowledge of the internal state (t_i,e) with a non-negligible probability in excess of 0.5 (e.g. see [La], 132 (theorem 7.1)), one would be able to determine t_{i-1} with a probabilistic polynomial-time algorithm (polynomial in $\lceil \log_2(N) \rceil$) stated in [ACGS]. If one assumes that polynomial-time algorithms are practically feasible (hypothesis!), then K4-specific property d)(v) is demonstrated: since, under the above assumption regarding the security of a RSA signature compliant with the German Digital Signature Act, it is practically impossible to determine s_{i-1} from s_i, guessing r_{i-1} from $r_i,\ldots,r_{i+j}$ can

ultimately no more likely to be successful than "blind" guessing, choosing both "0" and "1" with probability 1/2. In particular, this means that the "backward property" of K3-specific requirement d)(iv) is verified. But it is also the case that $t_v \equiv t_{v+1}^d \pmod N$ with $d \equiv e^{-1} \pmod{\phi(N)}$. The above line of argument naturally does not apply only to the initial state $s_0 := (t_0,e)$, but also to $s_0' := (t_{i+j+1},d)$. As neither e nor d is known, the K3-specific "forward property" is thus demonstrated by reason of symmetry.

On the basis of the above considerations (see also [La], 132 (theorem 7.1) and 126 (theorem 4.1)) one may assume that the c-tuples $(r_1,\ldots,r_c),(r_{c+1},\ldots,r_{2c})\ldots$ exhibit similar statistical properties as realisations of independent random vectors which are uniformly distributed on $\{0,1\}^c$. The K1 property follows as in example E.2 for small f. Likewise one may assume ([La], 126 (theorem 4.1)) that the RSA generator also passes the K2-specific tests. The probability of rejecting the null hypothesis should settle around the order of magnitude of a Type 1 error.

E.7 Seed generation

Let $p_A = \mu_S$ with $S = \{0,1\}^{128}$. The seed is calculated from keyboard entries made by the user prior to using the target of evaluation for the first time. He is allowed to enter upper and lower case letters, the numerals 0-9, "." and ":", altogether 64 characters. Every character entered is coded as a 6-bit word and succeeding 6-bit words are concatenated. Any character which is the same as its two predecessors is ignored. The bit string formed from the first 85 characters which are not ignored (510 bits) is hashed using RIPEMD-160. The seed is produced from the first 128 bits of the hash value. The user is informed in the user manual that the character string should be as "random" as possible.

Assessment: the procedure for generation of the seed is suitable for realising $p_A = \mu_S$. In fact an increase in entropy in each non-ignored character in the order of magnitude of at least 1.5 bits should be sufficient.

F. Statistical Tests

In this chapter the statistical tests which are performed in order to verify K2-specific property d)(ii) are listed. Tests T1 – T4 together with their designation and critical values are taken from [FI140] (4.11.1).

$b_1,\ldots,b_{20000}$ refers to a bit sequence of length 20000. If the sequence $b_1,\ldots,b_{20000}$ had been generated from an ideal noise source, then the probability of rejecting the null hypothesis on each individual test would be around 10^{-6}.

Test T1 (monobit test)

$X = \sum_{j=1}^{20000} b_j$

The sequence $b_1,\ldots,b_{20000}$ passes the monobit test if $9654 < X < 10346$.

Test T2 (poker test)

For j = 1,…,5000, let $c_j = 8 \cdot b_{4j-3} + 4 \cdot b_{4j-2} + 2 \cdot b_{4j-1} + b_{4j}$. Further, $f[i] := |\{j: c_j = i\}|$.

$Y = (16/5000)(\sum_{i=0}^{15} f[i]^2) - 5000$

The sequence $b_1,\ldots,b_{20000}$ passes the poker test ($=\chi^2$ goodness of fit test with 15 degrees of freedom) if $1.03 < X < 57.4$.

Test T3 (run test)

A run refers to a maximum subsequence of zeros or ones occurring in succession.

The sequence $b_1,\ldots,b_{20000}$ passes the run test if the number of occurrences of run lengths lies within the permitted intervals which are specified below. Runs of zeros and ones are evaluated separately.

Run length	Permitted interval
1	2267-2733
2	1079-1421
3	502-748
4	233-402
5	90-223
≥ 6	90-233

Test T4 (long run test)

A run of length ≥34 is deemed to be a long run.

The sequence $b_1,\ldots,b_{20000}$ passes the long run test if no long run occurs.

Test T5 (autocorrelation test)

For $\tau \in \{1,\ldots,5000\}$, $Z_\tau := \sum_{j=1}^{5000} (b_j \oplus b_{j+\tau})$

The sequence $b_1,\ldots,b_{20000}$ passes the autocorrelation test (with shift τ), if $2326 < Z_\tau < 2674$. (Note that the subsequence $b_{10001},\ldots,b_{20000}$ does not enter into the test statistic.)

G. Literature

[ACGS] W. Alexi, B. Chor, O.Goldreich and C.P. Schnorr: RSA and Rabin Functions. Certain parts are as hard as the whole. SIAM J. Comput. **17** (no. 2), April 1990, 194--209.

[FI140] FIPS PUB 140-1 (January 11, 1994), NIST, Security Requirements for Cryptographic Modules.

[FI186] FIPS PUB 186-1 (December 15, 1998), NIST, Specifications for the Digital Signature Standard (DSS).

[IEP] IEEE P 1363 Standard (August 22, 1996; Working Draft), Standard Specifications for Public Key Cryptography – Annex G: Cryptographic Random Numbers.

[La] J.C. Lagarias: Pseudorandom Number Generators in Cryptology and Number Theory. Proceedings of Symposia in Applied Mathematics **42**, 1990, 115--143.

[RSA] PKCS#1: RSA Encryption Standard. An RSA Laboratories Technical Note, Version 1.5, November 1, 1993.