INTERNATIONAL STANDARD

ISO/IEC 20543

First edition
2019-10

Information technology — Security techniques — Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408

Technologies de l'information — Techniques de sécurité — Méthodes d'essai et d'analyse des générateurs de bits aléatoires dans l'ISO/IEC 19790 et l'ISO/IEC 15408

Reference number
ISO/IEC 20543:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT security techniques*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Cryptographic applications need random numbers for a wide range of tasks. A strong cryptographic random bit generator that is suitable for general cryptographic applications is expected to provide output bit strings that cannot be distinguished with any potentially practical computational effort and any potentially practical sample sizes from bit strings of the same length drawn uniformly at random. Furthermore, such an RBG is expected to offer enhanced backward secrecy and enhanced forward secrecy.

Information technology — Security techniques — Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408

1 Scope

This document specifies a methodology for the evaluation of non-deterministic or deterministic random bit generators intended to be used for cryptographic applications. The provisions given in this document enable the vendor of an RBG to submit well-defined claims of security to an evaluation authority and shall enable an evaluator or a tester, for instance a validation authority, to evaluate, test, certify or reject these claims.

This document is implementation-agnostic. Hence, it offers no specific guidance on design and implementation decisions for random bit generators. However, design and implementation issues influence the evaluation of an RBG in this document, for instance because it requires the use of a stochastic model of the random source and because any such model is supported by technical arguments pertaining to the design of the device at hand.

Random bit generators as evaluated in this document aim to output bit strings that appear evenly distributed. Depending on the distribution of random numbers required by the consuming application, however, it is worth noting that additional steps can be necessary (and can well be critical to security) for the consuming application to transform the random bit strings produced by the RBG into random numbers of a distribution suitable to the application requirements. Such subsequent transformations are outside the scope of evaluations performed in this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 15408 (all parts), *Information technology — Security techniques — Evaluation criteria for IT security*

ISO/IEC 17825, *Information technology — Security techniques — Testing methods for the mitigation of non-invasive attack classes against cryptographic modules*

ISO/IEC 18031:2011, *Information technology — Security techniques — Random bit generation*

ISO/IEC 19790, *Information technology — Security techniques — Security requirements for cryptographic modules*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
backward secrecy
assurance that previous RBG output values cannot be determined from knowledge of current or subsequent output values

3.2
bit stream
continuous output of bits from a device or mechanism

[SOURCE: ISO/IEC 18031:2011, 3.4]

3.3
black box
idealized mechanism that accepts inputs and produces outputs, but is designed such that an observer cannot see inside the box or determine exactly what is happening inside that box

Note 1 to entry: This term can be contrasted with *glass box* (3.13).

[SOURCE: ISO/IEC 18031:2011, 3.6]

3.4
conformance-tester
tester
individual assigned to perform test activities in accordance with a given conformance testing standard and associated testing methodology

EXAMPLE An example of such a standard is ISO/IEC 19790 and the testing methodology specified in ISO/IEC 24759.

[SOURCE: ISO/IEC 19896-1:2018, 3.2, modified — The term "tester" has been added as an admitted term.]

3.5
deterministic random bit generator
DRBG
random bit generator that produces a random-appearing sequence of bits by applying a deterministic algorithm to a suitably random initial value called a seed and, possibly, some secondary inputs

Note 1 to entry: Non-deterministic sources can also form part of these secondary inputs.

Note 2 to entry: The security of a deterministic random bit generator rests primarily on the strength of its cryptographic algorithms and on the randomness contained in the seed value. In a deterministic random bit generator that is suitable for cryptographic use, at least forward and backward secrecy shall be assured without invoking secondary inputs to the RBG or reseeding."

3.6
enhanced backward secrecy
assurance that the knowledge of the current internal state of a random bit generator does not allow an adversary to derive with practical computational effort knowledge about previous output values

Note 1 to entry: The notion of enhanced backward secrecy is trivial for memoryless RBGs. Therefore, it is only a useful notion for deterministic and hybrid RBGs, the security of which rests at least in part on cryptographic properties of the state transition function and the output generation function of the random bit generator.

3.7
enhanced forward secrecy
assurance that knowing the current internal state of the random bit generator does not yield practically relevant constraints on subsequent (future) output values

Note 1 to entry: Deterministic random bit generators are unable to achieve enhanced forward secrecy. Unlike forward and backward secrecy as well as enhanced backward secrecy, enhanced forward secrecy rests entirely on the ability of a continuous reseeding process to supply as much entropy as is required to make the prediction of future outputs infeasible.

Note 2 to entry: It is possible for a random bit generator to have enhanced forward secrecy but still expand entropy, i.e. output a bit-string that can in principle be significantly compressed". For instance, one can consider an RBG design with a random source which produces at each invocation a 128 bit random string *R* with an estimated 120 bits of min entropy, with a 512 bit internal state *S(n)*, a state transition function giving *S(n+1) := SHA3-512(S(n)||R)*, and an output generation function applying SHAKE-256 on *S(n)||R* with up to 1024 bits of output per invocation.

Note 3 to entry: Another term often found in the literature that is interchangeable with enhanced forward secrecy is prediction resistance.

3.8
entropy
measure of the expected amount of information contained in a bit string given knowledge of how the bit string was generated

Note 1 to entry: There are various notions of entropy that play a role in cryptography. Worth mentioning among them are Shannon entropy, min entropy, collision entropy, guessing entropy, algorithmic entropy and Renyi entropy (the latter notion containing as special cases among others Shannon entropy, Min entropy and Collision entropy).

Note 2 to entry: The amount of entropy contained in an unknown bit string is always relative to an observer. RBG evaluations establish entropy estimates in face of an attacker with detailed knowledge about the entropy source and also consider her abilities to observe or influence the state of the entropy source.

Note 3 to entry: Irrespective of the chosen kind of entropy, the term "full entropy" always means the same, namely uniformly distributed and independent random numbers, that is, ideal randomness.

Note 4 to entry: An algorithmic entropy is a logarithm to the base 2 of the length of the shortest encoding in some given formal language. Its measure is based on the notion of optimal compression. The algorithmic entropy of a bit-string is dependent on the underlying formal language and even given a well-defined formal language, is in general incomputable unless the language is very restricted. However, related notions are of relevance in a cryptographic context. For instance, one can ask how much the sequence of raw random numbers derived from some physical noise source can be compressed using some fixed computationally efficient compression strategy that is informed by a precise understanding of the physical noise source and of the process that converts the output of the noise source into the raw random numbers.

3.9
entropy source
mechanism or device which produces intrinsically unpredictable output

Note 1 to entry: In the context of purely deterministic random bit generators, entropy generation can be performed just once, and in this case, it is possible for the RBG device not to contain an entropy source. The source of the entropy used by such an RBG nevertheless needs to be evaluated to the same standards that would otherwise be required.

Note 2 to entry: In some circumstances, it can be admissible for a deterministic RBG to be seeded with externally generated entropy instead of containing hardware that produces entropy within its own perimeter. In that case, the externally generated entropy shall only be available to the RBG instance it is intended for.

3.10
evaluator
individual assigned to perform evaluations in accordance with a given evaluation standard and associated evaluation methodology

Note 1 to entry: An example of an evaluation standard is ISO/IEC 15408 (all parts) with the associated evaluation methodology given in ISO/IEC 18045.

[SOURCE: ISO/IEC 19896-1:2018, 3.5]

3.11
forward secrecy
assurance that the knowledge of subsequent (future) values cannot be determined from current or previous values

[SOURCE: ISO/IEC 18031:2011, 3.13]

3.12
glass box
idealized mechanism that accepts inputs and produces outputs and is designed such that an observer can see inside and determine exactly what is going on

Note 1 to entry: This term can be contrasted with *black box* (3.3).

[SOURCE: ISO/IEC 18367:2016, 3.12]

3.13
health test
online test and total failure test
any mechanism (statistical test or otherwise) which detects at least one of the following two scenarios:

a) a transient or permanent total failure of the entropy source, i.e. a drastic decrease in entropy which usually manifests itself in a small number of easily detectable symptoms

b) smaller deviations from the normal behaviour of the entropy source, but nevertheless intolerable which undermine security claims made by the vendor. In contrast to a total failure, it usually requires a slightly larger sample size until these deviations are reliably detected

3.14
independent and identically distributed
IID
property of a family of random variables stating that they share the same distribution and are mutually independent

3.15
laboratory
organization with a management system providing evaluation and or testing work in accordance with a defined set of policies and procedures and utilizing a defined methodology for testing or evaluating the security functionality of IT products

Note 1 to entry: These organizations are often given alternative names by various approval authorities. For example, IT Security Evaluation Facility (ITSEF), Common Criteria Testing Laboratory (CCTL), Commercial Evaluation Facility (CLEF).

[SOURCE: ISO/IEC 19896-1:2018, 3.8]

3.16
min entropy
the min entropy of a finite random variable *X is –log2(p_max)* where *p_max* denotes the probability of the most likely outcome. That is, *p_max >=p_x* for all *x*

3.17
guessing entropy
guess work
<of *X*> expected number of guesses an adversary following an optimal guessing strategy needs to submit in order to guess the value of *x*[19], with *X*, a random finite variable and *x*, the value of a realization of *X* (i.e. a corresponding random variate)

Note 1 to entry: The formula for the guessing entropy is *\sum_i=1^n ip_i* where the *p_i* are ordered *p_1 >= p_2 >= ...* (that is, the optimal guessing strategy is to guess the most likely outcomes first).

3.18
non-dedicated non-deterministic random bit generator
NNRBG
non-deterministic random bit generator the security of which is not based on randomness generated by hardware that was designed explicitly to generate randomness

Note 1 to entry: TNRBG und NNRBG stand for true dedicated NRBG and non-dedicated NRBG, respectively.

3.19
non-deterministic random bit generator
NRBG
random bit generator that continuously samples multiple entropy sources and, if operating correctly, has an output that is expected to be unpredictable for attackers with unbounded computational capabilities over short timescales

3.20
perfect forward secrecy
property of a cryptographic protocol whereby an attacker cannot compromise past runs of the protocol by learning the long-term secrets of the participants

3.21
physical entropy source
entropy source based on the use of a dedicated physical effect (e.g. noisy diode, nuclear decay, etc.)

3.22
noise source
element of a technical system or its environment which produces partially unpredictable output. In this document, “noise source” and “entropy source” are taken to be entropy sources

3.23
non-physical entropy source
entropy source not based on a dedicated physical system but on unpredictable parts of the environment or technical components that were not originally designed for random bit generation

Note 1 to entry: Examples can be user input or the collection of various difficult to predict system data (e.g. hard drive access times, noise from a sensor device, system interrupts) in a standard computer.

3.24
post-processing
part of a random bit generator which processes the output of a random source with the aim of removing dependencies between random bits or biases. Is often also referred as a conditioning component

3.25
random bit generator
RBG
device or algorithm designed to produce bits that appear statistically independent and unbiased

Note 1 to entry: In case of purely physical random bit generators, the existence of very small entropy defects can be permitted. Deterministic RBG constructions, on the other hand, shall offer output that is computationally indistinguishable in practice from ideally distributed data. In addition, it is worth noting that hybrid designs have advantages over both purely deterministic and purely physical designs by combining the true entropy guarantees of physical RBGs with the near-ideal output distribution of deterministic RBGs and resilience properties, for instance with regards to noise source failure.

3.26
raw random numbers
bit sequence produced internally within a random bit generator by digitization of the random noise source or detection of unpredictable events within the machine in question, before any post-processing beyond the digitization has been performed

Note 1 to entry: It should be noted that although the raw random numbers represent an early stage in random bit generation, they can already contain complicated inherent pseudo-random patterns. For instance, part of the randomness in hard drive seek times is commonly associated to chaotic turbulent air flow patterns inside the hard drive; even if one abstracts away all other features of a hard drive, it seems difficult to argue that an RBG based on this effect does not have significant internal memory. However, sources with large internal memory are notoriously difficult to properly characterise by statistical tests with realistic sample sizes. The extent to which pseudorandom patterns are exhibited by a raw random source therefore depends on the design of the entropy source and shall be considered when analysing it. Generic statistical tests can mistake pseudo-randomness for actual randomness and thus overestimate the entropy of the raw random numbers. It is for this reason primarily that it is important to understand the design of the mechanism producing the raw random numbers. This comprises influences of the digitization mechanisms itself, e.g. resolution and non-linearity of A/D converters or noise produced by amplification circuits.

3.27
security strength
largest natural number, *n*, such that a computationally unbounded attacker cannot distinguish with more than negligible advantage an *n*-bit value produced by the RBG from an *n*-bit value drawn uniformly at random, when given the true prior distribution of internal RBG states

Note 1 to entry: If no such number *n* exists, the security strength is said to be infinite.

Note 2 to entry: Only hybrid or physical random bit generators can have infinite maximal supported security strength, as deterministic random bit generators always rely on an initial seed value. It is worth noting, however, that the output of pure physical random bit generators can often be distinguished from random data in practice if the design of any conditioning steps that can be performed is known to the attacker.

3.28
Shannon entropy
<of a finite random variable *X*> expected value of *–log2(px),* where *px* is the probability of observing the realization *X=x*

Note 1 to entry: In other words, for a finite random variable *X* with range *S* that the Shannon entropy *H(X)* is given by the formula $H(X) = -\sum x \in S \cdot px \cdot log2(px)$, where for the purposes of calculating the expected value one adopts the convention that *0*log2(0) = 0*.

3.29
stationarity
property of a stochastic process whereby the joint distribution of subsequent instances of the process is time-invariant

3.30
stochastic model
partial mathematical description of a random bit generator based on at least a qualitative understanding of the entropy source which, together with possibly some data gathered empirically for parameter estimation, allows the derivation of entropy claims

Note 1 to entry: In the context of evaluating random bit generators, it is recommended but not required that the stochastic model describe the behaviour of the raw random bits. Subsequent post-processing can make it more difficult to make a convincing case that the stochastic model is in sufficient correspondence with the workings of the device to be modelled to support the entropy claims to be shown. For instance, a stochastic model applied to the output random numbers of a deterministic random bit generator will be essentially untestable statistically insofar as strong cryptographic post-processing can render even very low entropy data indistinguishable from random noise at realistic sample sizes, at least from the point of view of any adversary lacking a stochastic model of the raw random numbers.

3.31
TNRBG
non-deterministic random bit generator the security of which is based on a hardware component that has been designed explicitly to generate randomness

Note 1 to entry: TNRBG und NNRBG stand for true dedicated NRBG and non-dedicated NRBG, respectively.

3.32
validation authority
entity that will validate the testing results for conformance to ISO/IEC 19790

[SOURCE: ISO/IEC 19790:2012, 3.132, modified — In the definition, "this International Standard" has been changed to "ISO/IEC 19790".]

3.33
vendor
entity, group or association that submits the cryptographic module for testing and validation

Note 1 to entry: The vendor has access to all relevant documentation and design evidence regardless if they did or did not design or develop the cryptographic module.

[SOURCE: ISO/IEC 19790:2012, 3.133]

4 Symbols and abbreviated terms

CCTL	Common Criteria Testing Laboratory
CLEF	Commercial Evaluation Facility
ITSEF	IT Security Evaluation Facility
LFSR	Linear Feedback Shift Register
OS	Operating System
SHA	Secure Hash Algorithm (SHA-256 and SHA3-512 referred to in this document)

5 Structure of this document

This document is divided into five clauses after the current clause: overview of non-deterministic random bit generators, conformance testing of NRBG, overview of deterministic random bit generator, conformance testing of DRBG and testing methodology. Each clause focuses on testing and evaluation activities for random bit generators for a conformance scheme using ISO/IEC 19790 and an evaluation scheme using the ISO/IEC 15408 series.

6 Overview of non-deterministic random bit generators

6.1 Introductory remarks on random bit generation

The current clause intends to demonstrate the problems of evaluating random bit generators and the security goals that are to be achieved by looking at the well-known setting of coin-tossing. One side of the coin is called "a head" (H) and the other is called "a tail" (T). Randomness is generated by tossing the coin into the air and noting which side is up when it lands.

Flipping a coin multiple times produces an ordered series of coin flip results denoted as a series of $H(s)$ and $T(s)$. For example, the sequence "*HTTHT*" (reading left to right) indicates a head followed by a tail, followed by a tail, followed by a head, followed by a tail. This coin flip sequence can be translated into

a binary string in a straightforward manner by assigning *H* to a binary one ("1") and *T* to a binary zero ("0"); the resulting example bit string is "10010".

The required properties of randomness can be examined using the example of the coin toss experiment described above. The result of each coin flip, from the point of view of using the output in cryptographic applications, is:

— *unpredictable*: Before the flip, it is unknown whether the coin will land showing a head or a tail. This is, in the case of a coin flip, contingent on not knowing with sufficient precision the initial physical parameters of the coin flip such as initial speed, height above ground, physical properties of the ground on which the coin is going to come to rest and rotation rate of the coin. If there is sufficiently low relevant entropy in the initial conditions of the flip, then the experiment becomes predictable[8]; what entropy is relevant for can only be determined by examining a physical model of the coin flipping process. But if initial conditions contain sufficient relevant entropy, the result is kept secret, and if initial conditions are not repeated too closely in a predictable manner on subsequent trials, it is not possible to determine what the result of flipping the coin was, given knowledge of any subsequent or previous outcome. The unpredictability after the flip depends also on whether the adversary can observe the outcome of the coin flip or not. The notion of entropy quantifies the amount of unpredictability or uncertainty relative to an observer and is discussed more thoroughly later in this document;

— *unbiased*: That is, each potential outcome has the same chance of occurring. The extent to which this is true depends on the same factors as listed above. Being unbiased in this sense means that each instance of the coin tossing experiment follows a uniform distribution (over the two possible outcomes *H* and *T*) and therefore that the sequence of coin tossing experiments is identically distributed as each experiment has the same probability distribution; and

— *independent*: The coin flip is memoryless; whatever happened before the current flip does not influence it. Whether this is true for a real coin toss experiment depends on whether the randomness entering the experiment via the initial conditions is memoryless and possibly on whether the coin itself changes, e.g. due to wear and tear over repeated experimental runs.

Simulating an idealized coin flipping experiment – i.e. a random source emitting a stream of bits that is unbiased, independent and identically distributed – is what cryptographic applications can generally aim for. The reason for this is that, while some cryptographic applications can tolerate significant deviations from ideal randomness (e.g. an AES-256 key is not brute-forceable if its bits are IID with 60 percent zeroes), others start leaking information even in the presence of small biases (for instance, secret sharing schemes) or can even get broken when a small amount of information about cryptographic secrets leaks (e.g. ECDSA nonces[21]). Also, the theoretically claimed security level of any cryptographic mechanism is often only reached if keys are ideally distributed. RBGs to be evaluated in this document simulate a series of idealized coin flips, even under strong assumptions on the abilities of any adversaries.

To evaluate whether a random bit generator supplies sufficient randomness, one needs to analyse the working principles of the device in question to arrive at a stochastic model ideally of the raw random numbers generated within the device. Based on this stochastic model, statistical tests can then be selected which enables the evaluator to derive estimates of the entropy contained in the raw random numbers.

6.2 Modelling of random sources

6.2.1 Stochastic models

6.2.1.1 General

Subclause 6.2 introduces the methods that are to be used in the modelling of random sources for evaluation in this document and define documentation requirements and evaluator actions related to that step in the evaluation of a random bit generator wherein it is checked that the stochastic behaviour of the entropy source is sufficiently well understood to proceed. Therefore, 6.2 does not itself define

minimum quality standards on the random source. Instead, such quality standards are defined by requirements imposed on the security claims to be submitted by the vendor. One abstract way to model a process that generates a random signal is by means of a stochastic model. As per 3.26, a stochastic model is a partial mathematical description of the system in question as a mathematical random process. A stochastic model is explicitly or implicitly a claim that the output of some circuit follows a probability distribution from a certain family of distributions.

The purpose of introducing a stochastic model into the evaluation of a random bit generator is fourfold:

a) Having a stochastic model of a randomness generating component transforms the generally intractable problem of ascertaining by black box testing whether the output of the device contains the desired amount of entropy into the possibly tractable problem to determine whether statistical testing yields results compatible with the hypothesis that the mechanism samples from one of the distributions covered by the stochastic model. Based on the stochastic model, it is then possible to test for the amount of entropy generated by the mechanism.

b) The stochastic model contains output distributions that correspond to defective states of the randomness producing device and statistical testing can then be used to determine that one is in one of these regimes. This is necessary as without a hypothesis about the behaviour of defective states, it is practically impossible to test for them.

c) The stochastic model can and shall be supported using technical arguments derived from the design of the randomness producing device that the stochastic model purports to model. Thereby, a connection is made between the technical properties of the device under evaluation and the claimed security properties of the core of the random bit generation mechanism.

d) Examining the stochastic model of an early stage of random bit generation and the supporting technical rationale allows the evaluator to confirm that technical arguments predict the general shape of the distribution of random output at a point where this output can still clearly be distinguished from ideal output. In contrast, many RBG constructions lead to output at the end-stage of random bit generation which is indistinguishable from ideally distributed output almost irrespective of the amount of true entropy contained therein.

The stochastic model needs to cover all technically plausible modes of failure or performance degradation.

For instance, a stochastic model for the output of a randomness producing device can claim that the output is identically and independently distributed for independent calls to the mechanism in normal operational mode and zeroes-only in the only technically plausible failure mode. The probability of spontaneously (without adversarial intervention) entering a mode with performance less than the security claims for the device may be claimed to be some low value per call to the mechanism.

Usually, a stochastic model encompasses some assertion of stationarity: that under suitable technical conditions the process in question is modelled by one member of the family of probability distributions and that over short time scales, the relevant distribution parameters are not expected to change greatly.

A stationarity claim of this type is not in contradiction to the notion that the device can experience effects of ageing, transient effects during start-up, or that it can fail. In the first case, the change in distribution parameters is too slow to affect sampling appreciably over short time spans; in the transient response case, the RBG has presumably not reached its operational state yet and cannot in fact yet be used; and in case of failure, the output distribution can change drastically, but this happens with low likelihood and the likelihood of it happening from an operational starting state does not change significantly with time.

Note that the question whether a process is stationary or not depends in part on the description of the process that is being used. For instance, a standard random walk is a classic example of a non-stationary process (the range of values that is being taken gets wider over time), but if the states reached in the random walk are used as a source of randomness, it can (depending on further processing steps used) be equivalent to instead consider the step-wise differences as entropy input, which yields an independent and identically distributed Bernoulli process.

Sources that are not amenable to being modelled by an underlying stationary process are harder to characterize than approximately stationary sources, because distribution parameters that a statistical test can attempt to estimate can in this case change over the course of sampling, shortly thereafter, or shortly before.

6.2.1.2 Requirements

A claim of (approximate) stationarity shall always be substantiated by technical arguments.

Therefore, in general a stochastic model shall:

— be a partial mathematical description of a stochastic process;

— describe precisely the stage of random bit generation in the device under study that is claimed as being modelled;

— allow for the efficient derivation of entropy claims for the distribution of the targeted stage of random bit generation from test data;

— cover technically plausible defective states of the mechanism targeted in the modelling;

— be supported by technical arguments based on the design of the targeted mechanism.

Furthermore, the description of technically plausible defective states of the random source that is contained in the stochastic model shall allow for the construction of statistical tests ("online health tests") that detect efficiently an intolerable deterioration of the quality of the source.

For example, the stochastic model can specify a parametrized statistical distribution and an allowed region in which the parameters of all devices lie to satisfy the security claims. An online health test can now apply a tailored statistical test to check whether a device's parameter still lies within that region.

6.2.2 Heuristic analysis of entropy sources

6.2.2.1 General

In some contexts, it can be impossible to constrain the distribution of digitized noise data by a stochastic model in the above sense. It can be difficult to find strong technical grounds for assuming certain characteristics of the underlying distribution. This is generally the case when randomness or at least the appearance of unpredictable behaviour is produced by a complex physical system, such as a human user or a computer. In this case, the large physical system cannot usually be understood at a sufficient level of detail to support the kind of technical argument backing the stochastic model required in 6.2.1; even basic properties like stationarity are often not given.

6.2.2.2 Requirements

The vendor shall then provide a heuristic analysis of the entropy source. The aim of heuristic analysis is to lower-bound the amount of entropy collected and to identify any plausible conditions that can lead these entropy claims to fail. In contrast, a stochastic model aims to derive a conservative, but ultimately realistic, estimate of the amount of entropy acquired by a mechanism, including a quantitative characterization of the distribution in weak states of the RBG.

The distribution of the random source output shall still be constrained to a family of distributions and a rationale shall be submitted that explains why the true distribution of the values under study can be expected to be contained in the claimed distribution family even though the processes producing the targeted intermediate output are only incompletely understood on a technical level. Naturally, such heuristic analysis shall always aim to err on the side of caution, i.e. underestimate the entropy provided. This includes assuming the greatest technically plausible ability for an adversary to influence or observe entropy generation.

6.2.3 Physical and non-physical sources

In the context of this document, intuitively, a random source is a component of an RBG which generates an at least partially unpredictable bit stream and for which the vendor has submitted either a stochastic model as outlined in 6.2.1 or a characterization of the produced entropy based on heuristic reasoning as outlined in 6.2.2. The function of a random source within an RBG is, thus, to produce genuinely non-deterministic behaviour. In some cases, random sources can be quite complex: think, for instance, of a system that draws entropy from user activities on a standard computer. In this case, the "random source" is the software component inside the computer which extracts information on the system state and writes it into a suitable buffer for post-processing together with the user who performs actions that are to some degree genuinely unpredictable.

On the other hand, dedicated random sources based on physical effects can be relatively simple: for instance, consider a small circuit that switches between two states in response to single photons hitting a small photo sensor and the state of which is read out at intervals dictated by an external clock.

This distinction between simple and complex random sources is of some importance in their modelling and subsequent evaluation. While, for dedicated physical sources, a solid understanding of the behaviour of the source on grounds of technical analysis can be gained, this is generally infeasible for complex, non-physical random sources. Therefore, this document distinguishes between non-deterministic random bit generators based on a true, dedicated physical source (TNRBGs) and random bit generators based on a non-dedicated source (NNRBGs).

NOTE 1 It can depend on the data supplied by the vendor whether a NRBG is considered a TNRBG or an NNRBG. For instance, an RBG can be considered a TNRBG if it draws entropy from both a hardware component and user interaction if the security claims for the RBG rest only on the claimed properties of the hardware component and if a stochastic model supporting the security claims for the hardware component has been submitted and made plausible by the vendor. On the other hand, the same RBG can be considered an NNRBG if the same security claims for the overall construction are supported at least in part by heuristic reasoning regarding entropy derived from user interaction.

NOTE 2 It is conceivable that with very conservative entropy claims, the random source of an NNRBG can be evaluated to a similar level of assurance and indeed using similar methodology as would be done for a dedicated physical source. In that case, one would for instance in a user interface device not focus on the entropy supplied by user interaction but on sensor noise present in the device that captures user interaction. However, in practice this is difficult as one usually does not have access to the raw data captured by such devices (e.g. an unmodified optical mouse does not allow one to access raw data from its optical sensor). Also, even if the raw data in question were accessible, the digitizing mechanism itself can be realized with much less complexity in a dedicated RBG than is the case in a device originally designed for another purpose.

6.2.4 Overview of the evaluation of the random source of a TNRBG

In the evaluation of a TNRBG, the study of its random source focuses on showing that, at the level of the random source output (the raw random bits), there is sufficient entropy and more generally that the raw random bits follow a distribution that is suitable in terms of supporting the security claims made for the whole construction. The random source shall be described by a stochastic model that includes all technically plausible failure cases and which is supported by technical arguments based on the source's design. The evaluator shall check that the technical arguments based on the design of the source supporting the assumption that the source behaves in accordance with the stochastic model of the source are plausible. The actual entropy output and other security relevant features of the output distribution of the real device shall be derived using statistical tests as described in Annex A, under the assumption that the stochastic model holds. In addition, the evaluator shall check whether test data indicates that the stochastic model holds. The vendor shall supply the stochastic model and all supporting documentation thereof and clearly indicate what stage of random number generation within the whole design the stochastic model targets (see Annex A).

The vendor shall also clearly indicate what effects if any are expected on the random source by ageing, a change in operating conditions (for all operating conditions specified by the vendor, e.g. temperature), or long-term use.

Finally, the evaluator shall verify that the claimed security properties of the random source are sufficient to support the security claims made by the vendor for the RBG-level output if any post-processing steps that can be applied are correctly implemented according to the mathematical description of such steps as provided by the vendor. The evaluator shall also check that the stochastic model in conjunction with online health and total failure tests and test data submitted as part of the evaluation documentation support the security claims made for the source. Standard cryptographic hardness assumptions (e.g. on the one-way properties of hash functions or block ciphers) can be used in the latter argument.

6.2.5 Overview of the evaluation of the random source of an NNRBG

In the case of NNRBGs, a dedicated physical random source does not exist, and the random source used is generally too complex to be completely understood by the evaluator. Any technical argument supporting the claim that the output of the random source follows a distribution from a particular family of distributions shall therefore remain incomplete. Instead, the vendor shall supply heuristic reasoning pursuant to 6.2.2.2 to show that the output at the random source level stays inside some, possibly broadly defined, family of probability distributions. The evaluator shall check the plausibility of the heuristic analysis so given. The vendor shall further supply experimental evidence that the heuristic reasoning is valid, and this experimental evidence shall also be verified by the evaluator. Finally, the evaluator shall verify that the claimed security properties of the random source, to the extent that they are backed up by the given heuristic reasoning, are sufficient to support the security claims made by the vendor for the RBG-level output if any post-processing steps that can be applied are correctly implemented according to the mathematical description of such steps as provided by the vendor. Standard cryptographic hardness assumptions can be used.

6.3 General design template and taxonomy for non-deterministic random bit generators

6.3.1 Overview

In the design of an NRBG, existing standards and best practice documents generally leave the designer more degrees of freedom than with most other widely used cryptographic constructions. Dedicated random bit generators inherently involve the design of specialized hardware, where very low-level details of hardware design have an impact on the security and the performance of the overall construction. On the other hand, NRBGs without a dedicated source shall use various low-entropy sources to generate unpredictability. The success of this depends ultimately on the amount of entropy that is available from any of these sources in a given situation. The problem of developing a random bit generator therefore is much less suited to a solution that fits all use cases than is, for instance, the problem of building a secure block cipher or indeed of a purely deterministic RBG.

Nonetheless, in terms of high-level design, templates underlying most practical NRBG constructions can be identified, and choices in high-level design have a direct impact on the documentation requirements and evaluation steps that are necessary to assess the quality of an NRBG. Subclause 6.3 has the following aims in that context:

— introduce a general design template for NRBGs to be evaluated in this document;

— introduce the taxonomy of non-deterministic random bit generators that is to be used in this document and link it to some extent to the relevant parts of ISO/IEC 18031;

— outline best practices with regards to NRBG high level design;

— explain the evaluation steps that are influenced by the high-level design of the RBG;

— define documentation requirements related to the high-level design of the RBG.

6.3.2 Functional model of a NRBG

In general, an NRBG takes as input a stream of unpredictable input bits produced by an entropy source and possibly some additional, can be predictable, input and generates from this a stream of

output random numbers. In an efficient NRBG construction, the memory used up by this shall be fixed and the functions that compute the next internal state and the next output shall be efficient. These considerations effectively lead to Figure 1 as a general NRBG template.

For the purpose of this design template, the primary entropy source is the entropy source on which the security claims made by the vendor rely. The vendor shall, in the documentation submitted by them to the evaluator, unambiguously identify the elements that correspond to the components of Figure 1 in their design.

The activities that need to be performed and the documentation that vendors should submit in the course of the evaluation of a non-deterministic random bit generator depend on the type of random bit generator. For this reason, the following taxonomy of random bit generators is used by this document.

NRBGs can be specified into sub-classes along at least three different axes:

a) NRBGs can be classified depending on the nature of their entropy source. Depending on the nature of the primary entropy source, the NRBG will be either a TNRBG or an NNRBG. This corresponds exactly to the distinction between physical and non-physical entropy sources in ISO/IEC 18031. In terms of minimal requirements on a DRBG, the evaluator shall verify that a TNRBG design meets at a minimum the requirements of ISO/IEC 18031:2011, 8.3.1.2 and 8.3.2.2 on RBGs with physical sources and that an NNRBG meets the requirements of ISO/IEC 18031:2011, 8.3.1.2 that apply to RBGs with non-physical entropy sources. Beyond this impact on minimal security requirements, the choice between a TNRBG and an NNRBG also influences documentation. The evaluation of a TNRBG shall always use a stochastic model of the primary entropy source. The vendor of an NNRBG can submit heuristic reasoning pursuant to 6.2.2 instead.

b) NRBGs can be classified depending on whether they are hybrid NRBGs or pure NRBGs. In a hybrid NRBG, a significant part of the security assurance of the NRBG comes from cryptographic processes similar to those employed in a strong deterministic random bit generator, but entropy is in addition gathered and mixed into the internal state on a regular basis from a true entropy source. In a pure NRBG, in contrast, the output of the NRBG is derived from the output of the random source by a transformation which cannot hide at least a total entropy loss of the source from an adversary: typically, non-cryptographic means, i.e. by much simpler conditioning, compression or mixing mechanisms such as von Neumann correction or linear feedback shift registers used as entropy accumulators. Hybrid NRBGs can, if designed properly, achieve security properties that neither pure non-deterministic nor deterministic RBGs can achieve: unlike deterministic RBGs, they can achieve enhanced forward secrecy, whereas unlike pure non-deterministic RBGs, a hybrid RBG can be able to continue producing output suitable for cryptographic applications even if the noise source fails in a way that is not detected by health tests. Also, achieving undifferentiability from an ideal source under all computationally feasible statistical tests is at the least very difficult in pure non-deterministic RBGs, but conceptually easy in DRBGs and hybrid RBGs. Hence, whether an NRBG is pure or hybrid needs to be considered when certifying certain security claims. For instance suppose that one security claim on the output of a pure NRBG is that the output cannot at realistic sample sizes be distinguished from ideally distributed data. Since in a pure NRBG the output generation and state update functions are not cryptographically strong, then very strong arguments is needed to support such a security claim. Unprocessed data from a physical entropy source (“raw random numbers") typically exhibit at least some miniscule deviation from ideal random numbers which can be detected when analysing large amounts of data.

c) NRBGs can be classified according to whether they are entropy compressing or entropy expanding. An example of an RBG which can reasonably be called non-deterministic, but which can under circumstances expand a smaller amount of entropy into a longer sequence of (cryptographically strong) pseudo randomness is the */dev/urandom* generator in UNIX-like operating systems.: over short enough time scales, */dev/urandom* behave exactly as a deterministic random bit generator; but reseeding is still continuous, i.e. the RBG tries to retrieve as much entropy as possible from the entropy sources present in the system, and assuming proper functioning of the entropy gathering process, there are some guarantees on the information-theoretic unrelatedness of outputs which are generated at sufficient (but not excessively large) temporal separation. Whether an NRBG is entropy compressing or entropy expanding is important mostly in terms of certifying security

claims related to the entropy of the output bits. For instance, an RBG that creates 256 bits of output for any 128 bits of entropy is generally only be able to support cryptographic applications up to the security level of 128 bits. Therefore, the vendor shall explicitly state how much entropy is claimed for each call to the RBG, depending on the parameters with which the RBG is called. The vendor shall indicate whether the data returned by different calls to the RBG can reasonably be treated as independent random data in an information theoretic sense. If, in the previous example, the RBG were to buffer the 256 bits of output containing 128 bits of entropy and deliver the first and the second half of the buffer to different callers, the security claim of 128 bits would not hold in an information theoretic sense (though it can still be secure in practice).Entropy compressing NRBG on the other hand aim to produce bit strings containing an amount of entropy equal or at least very close to the length of the output. To achieve this, they usually use compression, e.g. hash functions, to increase the entropy per output bit too close to 1.

As is the case with any kind of taxonomy, the boundaries between these classes of random bit generators cannot always be clear: for instance, a random bit generator can draw on both physical and non-physical entropy sources and it can then depend on the level of assurance that the evaluator can gain on either kind of source whether it is to be viewed as a physical or a non-physical NRBG; likewise, some constructions can have cryptographic post-processing in the form of e.g. a hash function that compresses the raw random numbers, but not cryptographic post-processing that in itself has the properties of a strong deterministic RBG, as required in the definition of hybrid RBGs above.

In general, the strongest security assurances can be obtained for hybrid RBGs with a well-understood physical source which do not expand entropy, and which have a cryptographic post-processing that imparts strong security properties on the output even in the case of an undetected failure of the random source. The requirements on the functionality class PTG.3 in [7] can serve as informative guidance in the evaluation of such generators. The core requirement on the entropy source is that there shall be a claim founded on a thorough understanding of the source's working principles about the minimum amount of entropy produced that one can expect to gather per readout. This claim shall consider the extent to which an adversary can be able to monitor the internal state of the entropy source. Note that resilience against hard to detect brief, transient failures of the entropy source can help in defending against fault attacks by some external attacker.

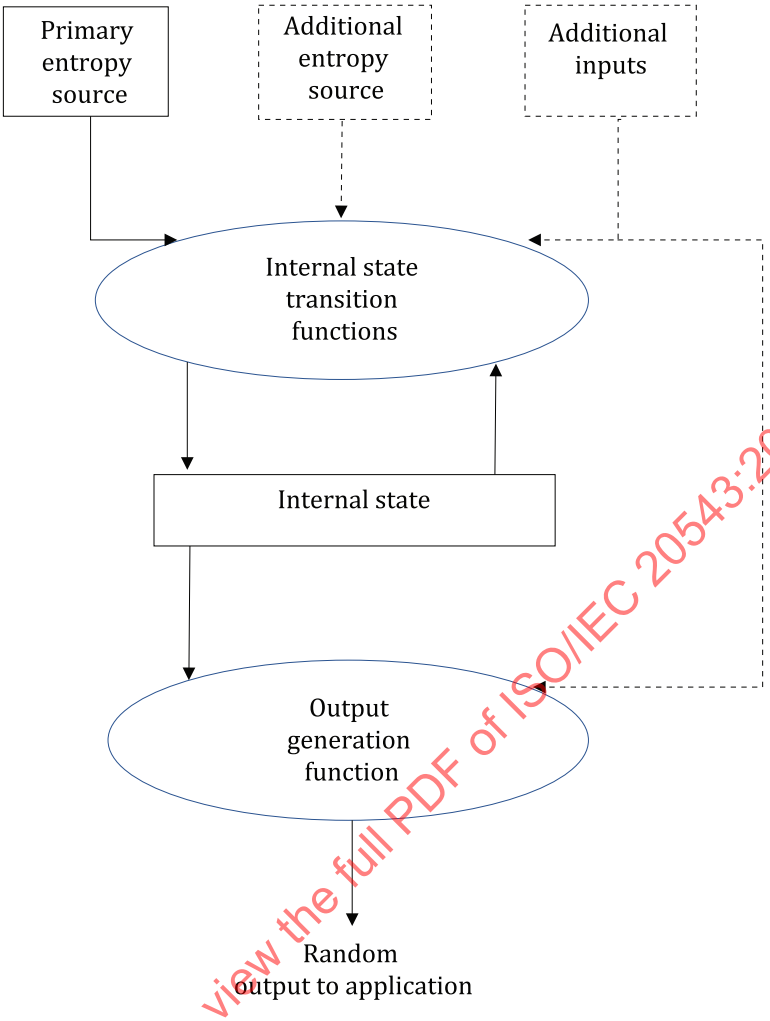

Figure 1 — Block Diagram of NRBG

6.3.3 Components of a NRBG

6.3.3.1 General

The following subclauses provide an overview of the evaluation steps required for each block illustrated in Figure 1. The focus of this subclause is on the technical content of the evaluation steps for each component. Detailed documentation requirements are treated in Clause 7.

6.3.3.2 Primary entropy source

This component serves as the source of unpredictability in the NRBG by providing data to be processed by the internal state transition function. In a NRBG, unpredictability is based on the use of one or more sources of entropy. These sources are known as entropy sources and can be further classified as either physical or non-physical.

The vendor shall provide a stochastic model of the primary entropy source respectively of the raw random numbers in the case where the primary entropy source is a dedicated device (i.e. if the NRBG is a TNRBG). If a failure is detected, the RBG shall not output random bits for which any claimed security

properties (e.g. amount of entropy) have been affected. Another stage of random bit generation than the raw random numbers can be targeted by the stochastic model, but it shall be clear which part of the random bit generation process is being modelled. This stochastic model shall be based on at least a qualitative understanding of the processes that give rise to the generation of entropy within the primary entropy source. If several entropy sources are used by a random bit generator, the source on which the vendor's security claims are mainly based shall be regarded as the primary source for evaluation. Stochastic model submitted by the vendor shall satisfy the requirements of 6.2.1.

If the primary entropy source is not a device originally designed to provide entropy, it can be necessary to submit in place of a stochastic model a heuristic analysis of the entropy collecting process reliably lower-bounding the entropy collected. This heuristic analysis shall meet the requirements of 6.2.2. In this case, evaluation testing and online health testing with the aim of verifying entropy lower bounds respectively assuring that the RBG has not entered some technically plausible state in which the claimed entropy lower bounds fail to hold shall be performed on the digitized raw entropy data before it enters post-processing.

Furthermore, the evaluator shall verify that health and total failure tests are performed at start-up of the generator and when drawing entropy from it to be injected into the internal state. Note that it can be useful to perform total failure tests on the entropy source after drawing data and then waiting for the test results before using the data. This avoids a situation where the entropy source breaks down after health testing, but before or while gathering data (possibly due to an external attacker). These tests shall be able to detect all realistic failure modes of the noise source that can lead to an intolerable deterioration of the quality of the raw random numbers. The proper functioning of the state transition function and the output generation function is more properly tested by known-answer tests. If known-answer tests of these functions are performed online, special care should be taken that no interfaces to these test functions are accessible from outside the NRBG perimeter. In particular, such test interfaces shall not be exposed to the consuming application.

It is expected that health testing is easiest to perform on the raw random numbers, as statistical defects is most easily detectable at the earliest stages of random bit generation. However, other approaches can be entirely valid if there is strong technical support for the conclusion that intolerable deviations from the desired distribution is detected. Additionally, it is worth mentioning that health testing can partially be covered by approaches that do not test the random numbers at all, but which e.g. monitor the physical state of the noise source (e.g. voltages in a noise diode) to detect tampering or accidental failure. The primary entropy source can be complemented in some design by additional entropy sources to improve the resulted entropy.

6.3.3.3 Internal state transition functions

The internal state transition functions control all the operations that alter the internal state. These include the mandatory functions that place the output of the entropy source into the working state, and that present part of the internal state to the output generation function.

The internal state transition function of an RBG will generally play an important role in mixing the output of the entropy source into the internal state and in ensuring that the final RBG output is indistinguishable from an ideal source.

In principle, the internal state transition function of a NRBG can be a non-cryptographic construction, such as a conditioning function intended to increase bitwise entropy and remove statistical dependencies in the output by discarding in a suitable way some of the information contained in the raw random numbers. For instance, von Neumann correction, if applied to an independent and identically distributed but biased source, will yield independent, identically distributed, and unbiased output. However, to obtain strong assurances about the output produced by such methods, one generally needs to understand the distribution of the raw random numbers, including any dependencies between them, quite well and in a quantitatively accurate way.

The required constraints on the behaviour of the raw random source cannot be obtained entirely by empirical study of the source: for instance, if one tries to assume generically that the random source is a simple machine with just a small number of internal states and some limited ability to be influenced by

its environment, a probabilistic finite state machine can appear like a natural mathematical model of that situation. The machine will produce outputs (bit strings) and receive inputs (changing environmental conditions such as temperature or voltage, which need to be discretized in this model). But the sample sizes needed to detect potential long-term dependencies that can severely lower the entropy of at least some bits of the output grow in the general case exponentially with the number of assumed internal states already in this model (note that this holds both for dependencies of outputs on changing inputs from the environment and dependencies between subsequent output bits with no input).

Generally, assumptions on the stochastic behaviour of the source will be needed to obtain strong entropy guarantees, such as the absence of practically exploitable multi-step dependencies or deviations from stationarity. These assumptions shall be justified by means other than statistical testing. Therefore, to apply statistical testing methods to the problem of ensuring that a raw random number source produces sufficient entropy for cryptographic applications, it is necessary to derive from the design of the source a partial mathematical description, i.e. a stochastic model, that places determining the entropy output of the source into a class of efficiently computationally tractable problems.

For instance, this can be the case if the vendor can provide solid evidence that the design of the source implies that the source should emit an independent and identically distributed byte stream. In this case, testing can on the one hand establish the min entropy of the byte stream based on the IID assumption, and on the other hand test the IID assumption in ways that will be based on knowing plausible failure modes of the device under examination.

In terms of resilience and general security properties, it is advantageous for the state transition and output functions to have strong cryptographic properties. Ideally, it should be possible to show forward secrecy, backwards secrecy, and enhanced backwards secrecy assuming a high-entropy internal state even if the noise source fails totally after initial seeding.

The evaluator of an RBG construction shall check that any post-processing and conditioning steps that are taken to transform the raw random numbers into the output random ensure under standard cryptographic hardness assumptions near-ideal properties of the output random numbers given the stochastic model of the raw random numbers. These evaluation steps shall consider the properties of the raw random numbers as given by the relevant stochastic model.

6.3.3.4 Internal state

This component consists of information that is carried over between calls to the NRBG, and all the information that is processed during a request. For this reason, an internal state is a mandatory component. However, it is not compulsory that any portion of the internal state depend on previous states, i.e., there is no mandatory requirement for any portion of the internal state to be carried over to the next NRBG call (e.g., in the coin flip example, no internal state is carried over from one-coin flip experiment to the next).

In such cases, the internal state of a NRBG is totally dependent on the output of the entropy source at the time that the NRBG is used, unless there exist additional non-mandatory inputs.

6.3.3.5 Output generation function

This component provides random output to the requesting application by processing all or a subset of the bits in the current internal state and any subset of the optional additional inputs.

Depending on the properties of the entropy source and the state transition function, the output generation function will generally serve as an important component in obtaining backward and particularly forward secrecy. The component can, if properly designed, prevent in this context the

random output from revealing information about the previous or current values of the internal state, entropy source inputs, or other random outputs.

NOTE Theoretically, it is possible to achieve nearly perfect behaviour in an RBG with trivial state transition and output functions: one "just" uses a near-perfect entropy source. However, designing a physical entropy source to generate, without a need for cryptographic post-processing, unbiased, identically distributed, independent random bits is hard; designing such a source which can be shown to a high level of confidence to have these properties is even harder. Also, such an RBG will still fail quickly if the noise source deteriorates in a way that is not detected by the health and failure tests. On the other hand, the design of deterministic random bit generators is well understood. Therefore, it is highly advisable for the state transition and output generation function to work together in such a way that the RBG will even with predictable input from the entropy source after some initial good seeding produce output that cannot be distinguished from random data by attackers not knowing the internal state.

6.3.3.6 Other considerations

The vendor of an RBG shall perform a taxonomic classification of the RBG according to suitable security properties: as a minimum, it shall be noted whether the RBG is non-deterministic or deterministic, entropy expanding or entropy compressing, whether the output is expected based on cryptographic standard assumptions to achieve indistinguishability from an ideal source as well as whether backward secrecy, forward secrecy, and enhanced backward and forward secrecy are achieved and on what basis this assurance is arrived at. Additionally, the vendor shall point out any properties of the RBG known to them that deviate significantly from a generic member of this class.

For instance, suppose that a vendor claims that their RBG is a hybrid, entropy expanding RBG which is backward secure, forward secure and enhanced backward secure based on standard cryptographic assumptions as well as enhanced forward secure based on being reseeded with 120 bits of entropy from a strong physical random source at each invocation. Generically, one would then expect that the vendor's security claim is that there exists no algorithm which can break, for instance, forward security with feasible data, time or memory requirements. But it is possible to imagine that with the design in question, this is not strictly speaking true and that instead the vendor's claim is just that forward security can only be broken by a memory- and time-efficient algorithm after the attacker has performed a one-time large precomputation to find the attack algorithm or if the attacker has themselves generated certain parameters to be used in the algorithm. Such deviations from security expectations based on claimed standard security properties of the RBG shall be clearly mentioned in the documentation submitted to the evaluator and shall be carefully considered by the evaluator when arriving at an evaluation result. If a security property of an RBG is found which greatly deviates from expectations based on the taxonomic classification of the RBG and which is not mentioned in the design rationale submitted, the security claims affected by the unexpected property should normally be rejected by the evaluator.

7 Conformance testing of NRBG

7.1 Overview

The goal of conformance testing in the context of RBGs is, in general terms, to show that the security claims made by the vendor are delivered by the implemented RBG and that the security claims made by the vendor at a minimum include the security requirements on the appropriate type of RBG as given in 6.3. With regards to the entropy source employed, this means that the vendor shall submit well-defined security claims in the form of entropy estimates derived from a stochastic model of the stage of random bit generation targeted by the stochastic model, usually the raw random numbers together with evidence that the stochastic model has plausibility given the physical details of the source's design. The tester then shall check that the reasoning linking the stochastic model with the design of the source and with the security claims is valid and that testing of the raw random numbers does not yield results contradicting the stochastic model.

Additionally, conformance testing shall ensure that the deterministic components of the RBG construction such as the state transition, output generation, health, and failure test functions are implemented according to their specification. The specifications of these components itself shall be

checked in a separate testing step for being suitable in terms of the entire RBG construction reliably providing a random bit stream indistinguishable from that emitted by an ideal source.

Conformance testing of a NRBG can be performed by a laboratory for a validation authority. The NRBG can be a component of a cryptographic module whose requirements are described in ISO/IEC 19790. ISO/IEC 18031 describes an overview and requirements for a NRBG but cannot describe a specific implementation or design. The implementation or design will be dependent on the sources of entropy and the requirements of the consuming application and can either be implemented in hardware, software or a combination thereof.

Therefore, for the purposes of this document, conformance testing shall determine if the design and implementation of an RBG supports the security claims made by the vendor. The evaluator shall determine whether the security claims made by the vendor conform at a minimum to the requirements on the appropriate type of random bit generator as given in 6.3. Any security property (such as forward security, backward security, enhanced backward security) shall at a minimum be claimed at a 100-bit security level. Security properties that have a known polynomial time quantum attack using only classical queries to the RBG shall not be accepted in this document.

7.2 Testing

7.2.1 Design documentation

The vendor shall provide complete documentation of the NRBG design and implementation to the testing laboratory. The documentation shall include a detailed logical diagram that illustrates all of the components, sources and mechanisms that constitute the primary entropy source. If multiple entropy sources are used, it shall be shown that any entropy sources that are not described cannot (under standard cryptographic hardness assumptions and technically reasonable assumptions e.g. on the ability of the other components to observe or to influence the primary source) reduce the entropy of the RBG output.

These components can include LFSRs, noisy diodes, thermal sampling, analogue to digital converters, entropy service calls from other components or modules, clock readings, memory cache hits, as well as various human-induced measurements, such as the time intervals between keystrokes, mouse movements, etc.

This documentation shall provide the vendor's rationale on determining the relationship between the amount of gathered entropy and the claimed randomness of the raw random numbers.

The documentation shall provide a stochastic model of an appropriate stage of random bit generation (typically, of the raw random numbers) or strong heuristic arguments lower-bounding the entropy collected if it is not feasible (for non-physical RNGs) to fulfil the documentation requirements for a stochastic model of some stage of random bit generation as laid out in 6.2.1.

NOTE Usually, a heuristically augmented lower entropy bound will not be tight, i.e. it will require a large security margin in terms of the ratio of internally generated entropy versus number of random bits emitted by the RBG (in other words, the post-processing will have a high "compression factor"). Lower entropy bounds derived from stochastic models will be tighter and thus allow for more efficient post-processing constructions.

7.2.2 Analysing entropy

7.2.2.1 Overview

The vendor shall provide an analysis of entropy. This means that the vendor shall provide arguments that will allow a random number generator expert who is knowledgeable about the principal components of the proposed random bit generation device to arrive at a reliable judgement as to whether the entropy claims made by the vendor are to be accepted. The guiding criterion in making this determination shall be whether it is possible to imagine operational circumstances under which the entropy claims made by the vendor fail. To this end, the vendor's heuristic analysis shall include assumptions on the ability of an attacker to observe or influence the device during operation.

Normally, this analysis should be based on a stochastic model of an appropriate stage of random bit generation. If it is infeasible to understand entropy production at the level of detail that is required to obtain a useful stochastic model, then in place of an analysis by use of a stochastic model a heuristic analysis deriving a reliable lower bound to the amount of entropy collected can be submitted. In this case, the vendor shall explain why the construction of a stochastic model is infeasible.

7.2.2.2 Requirements

The vendor **shall** provide the following documentation:

— a stochastic model of an intermediate stage of random bit generation (normally the raw random numbers) that can be used to derive an entropy claim for the output random numbers if free parameters are fixed to specific values. See 6.2.1 for detailed documentation requirements on this item. If several sources of entropy are used, stochastic models shall be provided for each source which is claimed to contribute to overall entropy generation. If it is not feasible to obtain a stochastic model of the random source and to support it with strong technical arguments based on the design of the source, heuristic arguments in accordance to 6.2.2 which reliably lower-bound the entropy acquired can be submitted in place of a stochastic model;

— a derivation of entropy claims for the output random numbers based on a stochastic model or heuristic analysis of the stage of random number generation modelled in the previous step;

— an explanation of any post-processing steps that transform the raw random numbers into the output random numbers. The exposition shall be detailed enough to allow independent re-implementation of the post-processing steps. Test vectors shall be given for all non-trivial post-processing steps or a standard shall be referenced that contains such test-vectors;

— an explanation of health-tests (all total failure tests and online tests) done during start-up and operation of the device to ensure that the raw random numbers produced at the random source are of sufficient quality. The health testing requirements for random bit generators of functionality class PTG.2 according to Reference [7] can serve as informative guidance if a dedicated random source is being used. The health tests performed shall cover all realistic failure modes of the device. The determination of what realistic failure modes can look like shall be based on an understanding of the physical processes used in the entropy source; and

— a document explaining unambiguously the claimed security properties of the overall construction.

In the case where randomness is supplied by a complex natural system such as a human user, the stochastic model of derived random events can be replaced by heuristic bounds on the min entropy gathered from the relevant events. These heuristic bounds shall be carefully argued for based on an adversary's ability to observe the relevant environmental inputs and on their observed variability. The worst plausible case shall be used.

If multiple entropy sources are used, it is admissible to add their contributions to a joint entropy estimate if a stochastic model of their joint distribution backed by strong technical arguments allows this or, failing that, if strong heuristic arguments can be given for assuming them independent.

7.2.2.3 Examples

7.2.2.3.1 Overview

In the sequel, some examples are given of the heuristic analysis of entropy that the vendor or testing laboratory shall provide. These are meant to outline the general lines of argument that heuristic analysis can follow; if one says, for instance, that three bits of entropy can be extracted from timing the intervals between user-induced keystrokes, this is not meant to imply that this is a generally acceptable estimate. Whether such an estimate is justified depends on the particulars of the situation at hand, not least on the extent to which an adversary is able to gain information about the timing of said key strokes.

7.2.2.3.2 Human-driven entropy generation

Human-driven entropy generation means using data on events triggered by a human user to generate entropy. In this case, it is ultimately unknown to what degree the noise source can be predictable, and therefore deriving a model of the distribution of the entropy containing raw data that can be backed up by technical arguments will be impossible. Therefore, entropy claims will in this case not be derived by examining a stochastic model of the entropy generating process but by lower-bounding the entropy collected by heuristic arguments.

In the case of human-driven entropy generation, the vendor can for instance say that three bits of entropy are gathered by measuring the time intervals between subsequent keystroke events. For example, the vendor can argue that based on the operating conditions of the device, an adversary can estimate the time at which the human user starts entering data on the keyboard at most with precision one second and that similar restrictions apply for acquiring timing data on any subsequent keystrokes; further, the vendor can argue that speed typing records suggest that even highly trained humans are unable to accurately type more than about twenty letters per second, indicating that the standard deviation in keystroke timing in a very highly skilled typist can be of the order of 10 ms. If the time measurement has millisecond precision, one can then assume that a sequence of keystrokes will yield 1000 evenly distributed possibilities for the timing of the first keystroke and three bits per subsequent keystroke. Hence, a heuristic estimate of the amount of entropy gathered with n keystrokes in this setting can be [10 + (n − 1) * 3] bits. The vendor shall, in such a case, provide measurements of the actual distribution of keystroke timings using a real input device as would be used in a deployed system and provide reasoning to rule out effects that can otherwise be expected to invalidate or to partially invalidate the assumptions underlying the estimate of gathered entropy given; for instance, in the setting of the example here under discussion, the sampling frequency of the input device itself instead of the system timer used can provide the real lower limit to the temporal resolution of the timings, thereby invalidating the assumption that the lower bits of the timings contain entropy at all.

If during the review of the entropy generated by any kind of entropy source it becomes apparent that the amount of entropy gathered by a certain process depends on variable external factors (as for instance in the preceding example, on the sampling frequency of an attached hardware device), the worst identifiable case that does not involve preventable adversarial action shall be assumed. The vendor shall explain the worst-case scenario they identified and how they identified it.

7.2.2.3.3 Entropy generation by dedicated physical devices

If the primary entropy source is a dedicated physical device, the vendor shall submit a stochastic model of a suitable stage of output generation of said device.

— If the entropy is generated by a physical device such as a sample of a radioactive isotope coupled with a detection device for nuclear decay events, such that the average rate of detected decay events is known and the random value is the number of atoms that have decayed in a particular time period, the vendor or laboratory shall state some known facts about the mean rate of the decay and the measuring device and also about either the distribution or at least about the variance of the number of the decaying atoms and give a rough estimate of the generated entropy. The vendor shall provide a stochastic model of the raw random number sequence and reasoning to support that model. The vendor shall further derive an estimate of the entropy gathered from that model. Note also that in this scenario, not all outcomes (numbers of decayed atoms) are equally likely. Instead, the distribution is centred on its mean value. Therefore, the vendor should either use the min-entropy estimate or come up with another reasonable and statistically sound lower bound on the generated uncertainty.

— If the entropy is generated by oscillating rings, the vendor or laboratory shall explain the design of the random noise generator. The design description in Reference [3] can serve as an example. However, to complete the description of the entropy source from the referenced presentation, the vendor or laboratory shall provide an explanation, at least heuristically, how the jitters are measured, how these measurements are used to generate the raw random number for the NRBG and how much entropy the raw random number carries. Again, the design description by the vendor

shall lead to a stochastic model of the raw random numbers produced by the noise source which can be tested, and which can be used to derive a reliable lower bound to gathered entropy.

— If the RBG is reseeded frequently, an estimate of the pool's entropy can be increased accordingly until limits given by the design of the entropy pool and the output generation function are met if the stochastic model of the entropy shows that treating subsequent entropy as independent is reasonable. This can for instance be the case if the physical entropy source is a sample of a radioactive isotope, which continues decaying independently (in some sense, and after adjusting by the number of the remaining atoms and changes in the reliability of detection equipment) of its history and therefore in this case the entropy values can be added without providing any further justification beyond considering the effects mentioned. Note, however, that if a claim of forward secrecy is made, an estimate of the entropy contained in the entropy pool that is appropriate to support claimed security levels must be justifiable even directly after a full state compromise.

7.2.2.3.4 Entropy gathering from the operational environment

If entropy is gathered from the operational environment of a module, it will often be impossible to single out a single, specific source of entropy that can be captured in a stochastic model. Evaluation will therefore in such a case focus on obtaining very conservative estimates on the amount of entropy collected and on making sure that unsafe operational states are reliably avoided.

— If the entropy is coming from an operational environment of the module a careful analysis shall be made of the source of entropy and of the ability of realistic attackers to observe or influence the relevant portions of the operational environment. If this source is an entropy provider in an operating system (e.g. *getrandom()* or */dev/random* in a Linux-based OS), a careful analysis of the generated entropy (possibly provided by the vendor of the OS) is required. The vendor or laboratory can refer to an independently published analysis of */dev/random* and *dev/urandom* such as [9]. Care should be taken to ensure that entropy claims are made only for versions of the dev/random or *dev/urandom* generators that have received peer review; whether this is the case in any situation depends on the version of the Linux RBG under consideration and the state of published scientific review (see Reference [22]). When relying on results about the quality of an operating system entropy provider, it is also important to check whether the operating conditions under which the algorithm was found to provide sufficient entropy are met. For example, an analysis can discover that the algorithm only provides entropy if the hardware is x86 and if the OS does not run in a virtual machine [see following reference].

— For certain versions of the Linux kernel and if the necessary operating conditions are satisfied, the following line of arguments can possibly be applied unless careful analysis finds otherwise the *dev/random* justification is the easier of the two. The OS satisfies a request for a random value only when it collects "enough" entropy; that is, when its own estimate of the collected entropy is such that a module's request can be met. For example, if the module needs to generate a 256-bit symmetric key and therefore the module requests 256 bits of entropy then the OS returns not the *dev/random* call until it is able to generate this much entropy. Until then, the module cannot generate the aforementioned symmetric key. However, the detailed setting in terms of used RBG version, entropy sources available in a particular device, and other relevant factors shall be taken into account when using any particular instantiation of the Linux RBG. For instance, it is not obvious that the assumptions made by the entropy estimator inside the operating system hold if it runs on a virtual machine.

In case of the *dev/urandom* request, the OS always sends an immediate reply back to the module. This reply can or cannot possess the desired amount of entropy. The non-blocking behaviour of */dev/urandom* implies that the consuming application has no guarantees about the amount of entropy it receives.

To meet the requirements, the vendor shall first demonstrate that the initial call (that is, the first call after the module has been powered up or instantiated) to *dev/urandom* returns the claimed amount of entropy. A possible way to achieve this is to analyse the sequence of events that precedes this initial call. If, for example, this sequence includes several restarts of the module and if each of these restarts includes several events that are measured and that provide the desired uncertainty, then a heuristic claim about the entropy in the initial call can be made. These events can include the times between

the restarts, the measurements of an operator activity during the restarts (mouse clicks, etc.), the values stored in certain memory locations that are known to be unpredictable during the restarts. This argument has a good chance of succeeding for stand-alone modules; embedded modules normally do not require multiple restarts so the use of *dev/urandom* in such modules is harder to justify. The same caveats as explained in the */dev/random* example apply.

If the vendor can justify having 100 bits of entropy returned on the first call to */dev/urandom*, then the vendor can often continue claiming that at least a 100 bit security strength is achieved on each subsequent call. The basic reason for this is that many versions of */dev/random* are designed to behave like a strong PRNG after being initially seeded even if there is no additional entropy input. Hence, from the point of view of an adversary, the uncertainty about any subset of the output sequence of length at least approximately equal to the amount of entropy in the seed should equal their uncertainty about the state of the seed.

NOTE Applications that aim at providing Perfect Forward Secrecy needs a random number generator that provides at least enhanced backwards secrecy, as otherwise the RBG state can be viewed as a long-term secret knowing of which allows an attacker to compromise past protocol runs. Likewise, if in this setting an RBG is used that fails to provide enhanced forward secrecy (i.e. that is not often enough reseeded or reseeded with insufficient entropy) then an attacker learning the internal state of the RBG can compromise future protocol executions. In general, these example considerations imply that care needs to be taken to review the requirements of the consuming application before endorsing a specific RBG for use, unless the RBG is a hybrid random number generator with frequent reseeding, forward secrecy even in case of failure of the noise source, enhanced backward secrecy and ideally also enhanced forward secrecy.

7.2.2.3.5 Other considerations

In any case, the design description shall include information on how ageing or accidental damage are expected to affect the performance of the noise source. For instance, in the case of a random bit generator based on a radioactive source, ageing will manifest itself in a reduction of the remaining number of unstable atoms over time as well as possibly in the mechanism detecting decay events becoming less reliable. The latter effect is much harder to understand well in this example than the former and therefore deserves deeper scrutiny.

For semiconductor-based sources ageing characteristics can be based on models.

7.2.3 Min entropy

Computing the Shannon entropy of a noise source can be difficult; furthermore, Shannon entropy can deviate significantly from guessing entropy, i.e. it does not (in the general case of not nearly uniform distributions) allow a reliable prediction of the expected amount of work an adversary has to perform in order to guess for instance a cryptographic key that was produced by a certain random bit generator. However, these deviations are largely benign from a cryptographic point of view: while no upper-bound on the guessing entropy of a distribution can be derived from knowing its Shannon entropy, and while Shannon entropy itself is not a lower bound to guessing entropy, such a lower bound can be derived from it [10]. Essentially, guessing entropy is never in excess of two bits lower than Shannon entropy and for distributions with large (more than a few bits) entropy, it is very close to one bit lower than Shannon entropy at worst.

But guessing entropy itself, besides being very difficult to estimate accurately for real-world random experiments, does not fully answer the questions that are of interest to the evaluator of a cryptographic random bit generator. It is possible that the expected workload of guessing a bit string produced by some random process is very high, but that the likelihood of guessing correctly very early in the execution of the optimal guessing strategy (on the first try, for instance) is also very high. In principle, one would like to have some guarantee of the form that an attack that succeeds with probability p has to do guess work of order at least $N*p$, for some large N (say, $N > 2^{100}$).

The simplest notion of entropy that allows one to obtain guarantees of this kind is min entropy. It is obvious that min entropy lower-bounds Shannon entropy and it can therefore be used (with possibly about two bits' loss) to lower-bound guessing entropy. Shannon entropy, on the other hand, can have some practical advantages over min entropy in some situations, e.g. the existence of a simple entropy

chain rule for dependent random variables. For distributions that can be expected to deviate only in minor ways from an ideal distribution, the Shannon entropy of the distribution will be very similar to the min entropy and can sometimes be easier to calculate.

An estimate of the generated min entropy that is a simplification of one of the methods proposed in Reference [2] is as follows.

This method would only apply if noise sources (and any conditioning components, if applicable) are IID (independent and identically distributed random variables). See Reference [2] (section 9.1.1) or any statistics textbook for an explanation of this notion. It is not necessary for the sources to produce a uniform distribution of the outcomes: the probabilities of different outcomes can be different. However, the probability distributions are identical between the sources (or between the different consequent readings of each source's random output) and these probabilities do not depend on the outcomes of other events generated by these sources.

Find the probability of the most common outcome among all the possible events generated by the noise source. If this probability is already known, then it can be used. The vendor or laboratory shall give a justification to why this probability is what it is claimed to be. If is not known, then, following Reference [2], take a dataset with N samples and count the occurrences of the most common value in the dataset. Again, following Reference [2], count the number of occurrences of this most common value in the dataset and denote the result.

Reference [2] presents a method to compute min entropy of a noise source by computing a confidence interval around the observed frequency of the empirically most likely outcome of the sampled random experiment. The upper bound of this confidence interval is then in Reference [2] used as a conservative estimate of min entropy.

The IID assumption for the raw random numbers shall be separately argued for based on the design of the raw random source. For distributions with small support, the IID assumption itself can then already be satisfactory as a stochastic model of the raw random numbers.

7.2.4 Statistical tests

The vendor shall run statistical tests on the stage of random bit generation targeted by their stochastic model. In this context, it shall be emphasized that the evidence that innocuous outcomes of statistical tests supply towards establishing trust in a given random source depends on various factors:

— Any statistical evidence supporting the entropy claims presented by the vendor shall be gathered separately from any statistical testing of components that was done during the development process; at the point in time when statistical evidence is collected, the design shall already be fixed. This rule is meant to prevent the following situation: a weak design is being tweaked to make it stronger. The tweaks that are being tried do not improve the design and statistical tests are repeatedly failed. But after a few iterations, the test used falls below the designated level of significance by chance, without the design under study having become appreciably stronger, and the test is judged to have been passed.

— A situation shall be avoided wherein a random source is iteratively "tuned" to pass certain statistical tests. The largest concern here is that such a development process can lead to a construction which hides the weaknesses of the initial design from the statistical test suite used without removing them; see e.g. the RBG presented in Reference [11] and the subsequent break of it in Reference [12] for a real-world example of the dangers of using a suite of statistical tests as a primary benchmark for a cryptographic RBG.

— The testing shall target the early stage of random bit generation that is described by the stochastic model". Depending on what post-processing is performed to obtain the output random numbers from the raw random numbers, the utility of performing statistical tests on the output of an RBG ranges from completely useless to somewhat dubious, at least when the goal is to judge the suitability of the underlying random source. Tests on the output random numbers can be useful too, but in general only for assuring that the transformation from raw random numbers to output random numbers has been implemented satisfactorily.

The vendor shall further demonstrate that the test result supports the vendor provided rationale. Typically, it takes several statistical tests to obtain a reasonable estimate of entropy. Some tests establish the degree of confidence in the independence of the observed values. Other tests can examine the short and long runs of bits and again, check the behaviours of these runs for their consistency with the claimed properties of the tested source. References [1], [2] and [7] can be used as informative guidance. The rationale shall be mathematically sound and consistent with vendor claims of randomness. The choice of tests shall be suitable to refute the stochastic model of the random source provided by the vendor if it fails in ways that appear reasonably plausible on inspection of the source's construction.

The vendor shall provide documentation on the development process of the random source along with the design rationale. The evaluator shall then check that the development process avoids concerns about the design having been tuned to pass a given statistical benchmark.

Annex B provides tests files for exemplary statistical tests. Reference [2] shows a sequence of statistical tests that would allow a vendor to test if the noise sources are IID. These tests can be used to test the IID assumption in the sense that they can have a chance to refute it. However, the IID assumption shall also be supported by a qualitative understanding of the noise source. Statistical tests alone are insufficient to support a claim of independent and identical distribution of the raw random bits. The test result shall be collected at representative environmental conditions inside the normal operating range (e.g. 25 °C, 0 °C, +100 °C for temperature). To the extent that the device itself is not capable of detecting excursions from the normal operating range, it shall be ensured by operational guidance that the device is not subjected to conditions outside the regime so indicated.

NOTE There are many ways in which a random source can fail to produce IID output, but a finite set of statistical tests can in practice only detect a limited number of specific defects. For example, an entropy source with complicated dependencies between its output bits or whose related output bits are not close to each other is not IID and can have exploitable weaknesses. But it is conceivable that such a source can still reliably pass statistical blackbox tests for IID.

7.3 Evaluation

7.3.1 General

Evaluation of a NRBG can be performed by a laboratory in accordance with a given evaluation standard and associated evaluation methodology. The NRBG can be a component of a cryptographic application or appliance whose requirements are described in ISO/IEC 15408 (all parts).

7.3.2 Vendor input to conformance testing

7.3.2.1 General

There are certain elements in the process of validating the design of a random bit generator for which the evaluator shall rely on vendor-provided information primarily. This clause will give some detail on the requirements on vendor input to the evaluation process.

The vendor shall provide all necessary material to demonstrate evidence of the claimed performance:

— a well-defined security target;

— a technical description of the entropy source;

— a stochastic model of an appropriate stage of random bit generation (or heuristic reasoning lower-bounding the entropy collected if obtaining a stochastic model is infeasible);

— test data obtained from the device and a statistical evaluation of the obtained test data including a justification based on the stochastic model of why the chosen statistical tests are suitable to back up the security claims pertaining to the random source made in the security target;

— a specification of any health and total failure testing steps;

— a justification of why the chosen health and total failure tests are appropriate;

— a specification of any post-processing or conditioning steps that are performed to obtain the output from the raw random numbers;

— a justification based on the properties of the random source and of any post-processing steps justifying all security claims made in the security target.

7.3.2.2 Security target

The vendor shall supply a security target for the RBG. The security target shall unambiguously define the min-entropy expected to be provided by each call to the NRBG under the least favourable operating conditions that are still within the operational envelope of the device. An estimate of Shannon entropy is acceptable instead of an estimate of min-entropy if the entropy defect is very low.

The entropy claim will be expressed quantitatively, i.e. as a number of bits of entropy per call of the RBG.

The security target shall also outline unambiguously the range of operating conditions the device is meant to function under and any additional assumptions that are needed to support secure operation. It shall include a claim as to the basic properties of the random bit generator. This claim shall be backed up by suitable design documentation and should include whether the RBG is deterministic or non-deterministic, whether seeding is based on dedicated physical effects or not, whether its security can be backed up by cryptographic arguments under standard assumptions, and whether forward secrecy, backwards secrecy, and the enhanced versions thereof are supported.

The security target shall outline any properties of the RBG that the vendor is aware of and which deviate from generic expectations on the RBG behaviour based on the rest of the security claims made by the vendor.

7.3.2.3 Technical description of the entropy source

The vendor shall supply a complete technical description of the entropy source. If a dedicated entropy source is used, this can, for example, take the form of the complete circuit diagram of an electronic circuit that has as its output the raw random numbers. The description shall, in this case, contain an explanation of what underlying physical effect is exploited to obtain randomness.

If a non-dedicated source is used, the description shall include a list of all hardware components that are expected to be present in the system and shall explain how these hardware components are being queried to obtain entropy. If subsequent security claims rely on physical effects happening within some of the hardware components, the description shall contain an explanation of why these physical effects are expected to produce randomness

7.3.2.4 Stochastic model

The vendor shall supply a stochastic model of the raw random numbers, or of another appropriate early stage of random bit generation, as part of the documentation supporting their entropy claim. See 6.2.1 for details on the requirements regarding the stochastic model. If it is infeasible to construct a stochastic model, heuristic reasoning pursuant to the requirements of 6.2.2 can be submitted in its place.

7.3.2.5 Vendor-defined tests

The vendor shall supply evidence based on statistical testing backing up their entropy claim. The vendor shall provide a rationale showing that the statistical tests employed together with the stochastic model of the random source can constrain the amount of entropy emitted by the random source to a level at or above the entropy claim with a high level of confidence (see Annex A).

7.3.2.6 Health testing

The vendor shall supply documentation on any health and total failure testing performed by the source. This documentation shall be precise enough to allow for a reimplementation of health and total failure testing. It shall also precisely specify the stage of random bit generation which is being tested. The vendor shall show that deviations from expected performance causing a loss of entropy below the security claim made by the vendor will be detected by the health tests chosen with high probability.

7.3.2.7 Conditioning components

If a conditioning component is used, the vendor shall provide the specification of the conditioning component. The vendor provided specification shall include mathematical analysis why the entropy claim made is being met at the output of the conditioning component. It shall also include rigorous reasoning on whether the component interferes with any health testing measures employed and how the component itself can fail.

8 Overview of deterministic random bit generators

8.1 General remarks

The dividing line between hybrid and purely deterministic random bit generators is by nature not sharply delineated; a DRBG can reseed regularly. For the purposes of this document, a random bit generator is called deterministic if reseeding is not performed continuously. The key property of a deterministic RBG to be evaluated is that under a wide range of attack scenarios it provides output that cannot be distinguished from ideally distributed random data by a computationally bounded adversary.

In this setting, there are in principle the following areas that need to be checked during the evaluation of a DRBG:

— *Initial seeding*: The vendor shall show that the initial seeding process produces a seed distribution that cannot be exploited by an adversary with full information about the DRBG design to break any claimed security properties of the DRBG substantially faster than is implied by the security strength claimed for the DRBG. In arguing that the initial seeding has this property, the vendor can use generally accepted cryptographic hardness assumptions, but they shall make these assumptions explicit.

NOTE A high seed entropy even together with a state transition function and output generation function that work together well is not in general enough to ensure good properties at the level of RBG output. For instance, it is not difficult to construct a DRBG which is forward secure, backward secure and enhanced backward secure if the seed state is picked uniformly at random from the set of all possible states, but for which the state transition function also has a high number of efficiently computable fixed points. In such a construction, it is possible to imagine the seed process producing only fixed points of the state transition function while remaining high-entropy, thereby voiding all security assurances of the DRBG.

— *Reseeding*: If some form of prediction resistance is being claimed in the security target supplied by the vendor, then reseeding shall use a high-quality source of entropy. The entropy gathered during reseedings shall be quantified and the reseed process shall be subject to the same evaluation steps as the initial seeding. The same criteria shall apply whenever the reseed process can potentially reduce the entropy of the DRBG internal state (e.g. if part of the internal state is overwritten with freshly generated random data). In all other circumstances, a high-quality source of entropy should be used.

— The design of the state transition and the output generation function shall be evaluated. The security claims made for the DRBG in the submitted security target shall be reducible to hard cryptographic problems: for instance, to well-studied properties of the employed cryptographic primitives, such as the one-wayness of a hash function or of a block cipher when viewed as a function of its key. The argument supporting the claimed security properties shall consider the assumed distribution of seed values as deduced from the properties of the seeding process. All claimed security properties shall be shown to hold under standard cryptographic assumptions

against attackers able to execute the underlying cryptographic primitives (such as hash functions, block ciphers) up to 2^n times or to execute other calculations with a similar cost factor, where n is the claimed security strength of the RBG.

— The implementation of the state transition and output functions shall be checked for conformance with their specifications and against known-answer tests.

The key property of a deterministic RBG to be evaluated is that it shall, under a wide range of attack scenarios, provide output that cannot by a computationally bounded adversary be distinguished from ideally distributed random data. A computationally bounded adversary in this sense is an adversary who cannot execute more instances of some simple cryptographic operation (e.g. calculation of a hash on a single message block) than is indicated by the claimed security strength of the RBG.

The following properties shall generally be checked:

— Forward and backward secrecy can be shown under standard cryptographic assumptions. Enhanced backward secrecy and prediction resistance should be checked in generators that are to be evaluated with the aim of using them in arbitrary cryptographic applications or if the consuming application needs these properties. Of these two properties, enhanced backward secrecy is mandatory for evaluation in this document if the RBG is to be used in arbitrary applications; prediction resistance is only mandatory between reseeds, but is in general a highly desirable property to achieve between different invocations of the RBG.

— Initial seeding and (optionally) reseeding provide enough entropy to support the cryptographic security guarantees claimed under the previous point.

— Appropriate health tests are performed on all components of the RBG. This is most important for the seeding and reseeding mechanisms; health testing of the deterministic component can be done once at start-up by processing some test vectors.

Protection from side channel and fault attacks is of equal importance for the deterministic and the non-deterministic component.

If the security target for the RBG claims a security level for the RBG according to ISO/IEC 19790, then evaluation of side channel resistance shall at a minimum include the applicable test and evaluation methods given in ISO/IEC 17825. If no security level according to ISO/IEC 19790 is claimed in the security target, then the evaluator can determine an applicable security level from the stated operating conditions and the intended uses of the RBG and use that security level as the basis for evaluating side channel mitigation.

8.2 Structural overview of a deterministic random bit generator

Structurally, there is no fundamental difference between DRBGs and NRBGs. All remarks and requirements of 6.3.2 and 6.3.3 therefore apply unchanged. In terms of evaluation requirements, the difference between DRBGs and NRBGs is one of emphasis: DRBGs are random bit generators without continual reseeding; therefore, their security is entirely based on computational complexity assumptions. This has the following consequences:

— Unlike for non-deterministic random bit generators, where certain applications which can tolerate small statistical defects within the output random numbers no such measures are absolutely needed, the use of strong cryptographic mechanisms within the random bit generator is mandatory for deterministic random bit generators. Evaluators shall assure that forward secrecy, backward secrecy and enhanced backward secrecy of the DRBG can be shown to hold under standard cryptographic assumptions if the seed of the DRBG provides sufficient entropy. Even understood weaknesses of the DRBG construction shall be treated with the utmost caution. No empirically demonstrable statistical bias in the output random numbers is to be deemed acceptable for DRBGs.

— The lack of continual reseeding also means that special attention shall be paid on the need to ensure that the initial seeding was of high quality. Therefore, independently of the security claims made by the vendor in the security target, a DRBG should be seeded with at least 120 bits of Shannon

entropy and should have at least 200 bits internal state. If either of these conditions is not met, the evaluation report shall note this, but can still accept all security claims made by the vendor.

— It is mandatory that neither the seed value nor subsequent iterations of the internal state value become known to an adversary. While a non-deterministic random bit generator will generally have some ability to recover from state compromise, this is not true for deterministic random bit generators.

9 Conformance testing of DRBG

9.1 Overview

The goals of conformance testing for DRBGs are as given in Clause 7. In general, all remarks given in Clause 7 apply. For this reason, this clause will only point out some issues within the documentation of an RBG design that require special attention in the evaluation of a DRBG.

9.2 Testing

9.2.1 Design documentation

In the case of a DRBG, the design documentation submitted according to the requirements laid out in Clause 7 shall provide evidence of achieving backward secrecy, forward secrecy and enhanced backward secrecy based on standard cryptographic hardness assumptions. Additionally, the design shall consider side channel and fault injection attacks against the state transition and output generation functions and reliably protect the confidentiality of the internal state. The requirements on the documentation and design goals of the random source to be used for seeding are as in Clause 7.

NOTE 1 The fact that this document mentions the above points specifically with regards to deterministic random bit generators does not in any way imply that they are not to be considered in the evaluation of hybrid random bit generators under Clause 7. In principle, the same points apply to the evaluation of the deterministic parts of all hybrid random bit generators.

NOTE 2 Enhanced backward secrecy can from a purely cryptanalytic point of view be obtained by constructions that have a strong state transition function, but which leak part of the internal state in the output; imagine for instance a sponge construction based on an arbitrary random-looking mapping instead of a permutation as the cryptographic core component. However, it is worth noting that constructions that partially leak the internal state without post-processing it can yield bad randomness immediately if the attacker can manipulate the internal state e.g. by applying a fault attack. Also, side channel attacks can become easier if there is state leakage, as the adversary can build e.g. power templates with partially known data for the exact device under attack. Adding cryptographic security guarantees protecting against partial state leakage is therefore recommended in DRBG designs.

9.2.2 Analysis of seed entropy

The vendor shall provide an analysis of seed entropy. The requirements for this analysis are defined in Clause 7.

The vendor shall supply a stochastic model of the entropy source if using a dedicated physical entropy source and specify statistical tests that are suitable to derive a reliable lower bound on the amount of entropy collected. If seeding is done from a non-physical entropy source, heuristic reasoning in accordance to 6.2.5 in support of the entropy claim for the seeding process shall be submitted. Furthermore, a claim as to the entropy provided by the initial seeding shall be specified and shall be justified using the stochastic model or the heuristic lower-bounding of collected entropy that was performed by the vendor. Entropy claims for subsequent reseed processes are required only if they are of relevance to security claims for the overall construction. In addition, a rationale shall be provided showing that the result of the initial seeding cannot be observed or otherwise inferred at potentially practical cost by an adversary and that any subsequent reseedings meet the same requirements at least insofar as they are critical to the security claims made by the vendor.

In addition to any security claims made for the seeding process, the vendor shall submit a security claim also for the DRBG output. A rationale shall be provided showing that under cryptographic standard assumptions the security claims for the DRBG output will be met if the same condition is fulfilled at the seeding process. The seeding process is in principle allowed to deviate significantly from an ideal random source provided that the aforementioned conditions are met.

10 Testing methodology

10.1 General

To perform consistent evaluation both for NRBG and DRBG, it is recommended to use the following numbering scheme. This will also guarantee full traceability of the requirements and testing requirements.

10.2 Vendor requirements

The vendor requirements are numbered under the following frame:

VE<requirement_number>.<assertion_sequence_number>.<sequence_number>

— requirement number refers to conformance testing clause and sub clause for NRBG and DRBG,

— assertion sequence means the order in the sub clause,

— sequence number means the detailed justification to be provided.

10.3 Tests requirements

The test requirements are numbered and provide the list of the tests performed by the evaluator or tester.

TE<requirement_number>.<assertion_sequence_number>.<sequence_number>

where <requirement_number>, <assertion_sequence_number> and <sequence_number> are as defined in 10.2

Annex A
(normative)

General statistical methodology

A.1 General

This annex describes examples of general statistical procedures and models and examples of RNG designs and stochastic models which are common for different parts of this document (e.g. chi-square test, etc.). [25][26][27][28]

A.2 Statistical tests

A.2.1 General remarks

Statistical tests to be used in the evaluation of an RBG shall be tailored to the stochastic model of the source and executed using the stage of random bit generation targeted by the stochastic model.

The vendor shall further identify statistical tests which will detect failures or conditions of insufficient entropy. The evaluator will examine the rationale provided by the vendor for correctness and completeness, check that the statistical tests suggested will reliably detect the plausible failure conditions identified, and carry out testing to determine if the random source does deliver the amount of entropy required. In the case of non-deterministic RBGs without a cryptographically secure conditioning component, they will in addition check that the distribution of the output random numbers is close to ideal. The requirements for achieving the PTG.2 property of Reference [7] can serve as informative guidance.

A.2.2 Baseline testing

In addition, baseline statistical testing shall be performed on both the raw random numbers (or more generally the stage of random number generation targeted by the stochastic model) as well as the output random numbers. Any findings that contradict security claims or that are incompatible with the stochastic model shall lead to the security claims in question being rejected.

The recommended set of baseline tests is the test procedure A and B of Reference [7].

A.2.3 Health and total failure testing

Health tests shall be performed online and shall be designed to detect entropy defects significant enough to threaten any security claims made for the RBG. The nature of the health tests shall be determined by examining all technically plausible failure modes of the RBG.

Total failure tests shall reliably detect a total failure of the noise source. After a total noise source failure, the RBG shall not output random bits if any claimed security properties for these random bits have been affected by the failure of the random source. It shall be shown that in the event of failure, detection of failure is expected to happen with high likelihood at any subsequent invocation of the RBG.

Again, the type of statistical tests to be employed here depends on an examination of all technically plausible cases which would lead the RBG to cease emitting entropy.

A.2.4 Other considerations

Statistical tests performed during the evaluation of an RBG should be carried out on several independent instances of the device and under varying operating conditions to test the assumption that the entropy claims made by the vendor are valid for the entire operating envelope of the device under test.

A.3 Examples of stochastic models

A.3.1 Overview

In the sequel, some simple examples of stochastic models are introduced. As per 3.26, a stochastic model of a random source is a partial mathematical description of the expected statistical properties of the source which allows the derivation of entropy claims if the stochastic model is combined with appropriate test data. The vendor therefore has the following tasks:

— finding a family of distributions which contains the true distribution of the random source,

— showing using arguments based on the engineering details of the entropy source that the proposed family of distributions is reasonably expected to contain the true distribution produced by the random source,

— showing that the family of distributions is restricted enough to allow for a determination of the produced entropy to be made based on statistical methods, e.g. parameter estimation, and

— selecting appropriate statistical tests as well as stating an entropy claim for the source.

A.3.2 Remarks

— Usually, the family of distributions can be characterized by one or several parameters.

— It seems reasonable to require the stochastic model to assume stationarity, since otherwise for real-world RBGs the verification of the stochastic model can become too difficult. However, whenever a claim of stationarity is made, it shall of course be supported by technical arguments based on the design of the RBG under study.

The evaluator shall verify that the stochastic model is appropriate given the design of the source. They further shall verify that the statistical methods suggested by the vendor to derive an entropy estimate are suitable for that purpose and give reliable results. In addition, it shall be checked that the entropy claim made by the vendor is met by the design under all operating conditions. Finally, the entropy output expected from the entire device, given any subsequent post-processing that can be performed before outputting data to consuming applications, shall be found to be suitable given the attack profile the RBG is to be evaluated against.

It is obvious that some steps of this process can depend on details of implementation. The following examples are meant to show how some of these steps can influence the process of evaluation of a random source. They are not meant to address all possible concerns or to propose designs ready for deployment as is in security-critical applications.

In Reference [14], an estimation method is proposed which accumulates intermediate value of the number of frequencies.

A.3.3 Notations and conventions

In A.2.4. random variables are denoted by uppercase letters and the values of a realization of the corresponding experiment is denoted by lowercase letters. The same convention applies to sequences of random variables respectively variates: here the position in the sequence will be denoted by a subscripted index, i.e. A_i is the *i-th* element in the sequence of random variables A_1, A_2, ..., A_i,..., and a_i will be the corresponding realization. Random variables in this annex are real-valued unless indicated otherwise. Functions of a single real variable t is denoted in lower-case letters, e.g. $f(t)$. In a slight abuse