



BSI Standards Publication

Information technology — Security techniques — Random bit generation

National foreword

This British Standard is the UK implementation of ISO/IEC 18031:2025. It supersedes BS ISO/IEC 18031:2011+A1:2017, which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee IST/33/2, Cryptography and Security Mechanisms.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2025
Published by BSI Standards Limited 2025

ISBN 978 0 539 15236 4
ICS 35.030; 35.040

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 28 February 2025.

Amendments/corrigenda issued since publication

Date	Text affected



**International
Standard**

ISO/IEC 18031

**Information technology —
Security techniques — Random bit
generation**

*Technologies de l'information — Techniques de sécurité —
Génération de bits aléatoires*

**Third edition
2025-02**

Reference number
ISO/IEC 18031:2025(en)

Full document details: BS ISO/IEC 18031:2025

© ISO/IEC 2025



Documents

Document Sharing Hub



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2025

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

Published in Switzerland

Contents

Page

Foreword	v
Introduction	vi
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Symbols	7
5 Properties and requirements of a random bit generator	8
5.1 Properties of a random bit generator	8
5.2 Requirements of an RBG	9
5.3 Additional information for an RBG	10
6 RBG model	10
6.1 Conceptual functional model for random bit generation	10
6.2 RBG basic components	11
6.2.1 Introduction to the RBG basic components	11
6.2.2 Randomness source	11
6.2.3 Additional inputs	12
6.2.4 Internal state	12
6.2.5 Internal state transition functions	13
6.2.6 Output generation function	14
6.2.7 Health test	15
7 Types of RBGs	15
7.1 Introduction to the types of RBGs	15
7.2 Non-deterministic random bit generators	16
7.3 Deterministic random bit generators	17
7.4 The RBG spectrum	17
8 Overview and requirements for an NRBG	17
8.1 NRBG overview	17
8.2 Functional model of an NRBG	18
8.3 NRBG entropy sources	20
8.3.1 General	20
8.3.2 Primary entropy source for an NRBG	20
8.3.3 Physical entropy sources for an NRBG	22
8.3.4 NRBG non-physical entropy sources	22
8.3.5 NRBG additional entropy sources	23
8.3.6 Hybrid NRBGs	24
8.4 NRBG additional inputs	24
8.4.1 NRBG additional inputs overview	24
8.4.2 Requirements for NRBG additional inputs	24
8.5 NRBG internal state	25
8.5.1 NRBG internal state overview	25
8.5.2 Requirements for the NRBG internal state	25
8.5.3 Additional information for the NRBG internal state	26
8.6 NRBG internal state transition functions	26
8.6.1 NRBG internal state transition functions overview	26
8.6.2 Requirements for the NRBG internal state transition functions	27
8.6.3 Recommendations for the NRBG internal state transition functions	27
8.7 NRBG output generation function	27
8.7.1 NRBG output generation function overview	27
8.7.2 Requirements for the NRBG output generation function	28
8.8 NRBG health tests	28
8.8.1 NRBG health tests overview	28
8.8.2 General NRBG health test requirements	29

8.8.3	NRBG health test on deterministic components.....	29
8.8.4	NRBG health tests within entropy sources.....	30
8.8.5	NRBG health tests on random output.....	31
8.9	NRBG component interaction.....	32
8.9.1	NRBG component interaction overview.....	32
8.9.2	Requirements for NRBG component interaction.....	32
8.9.3	Recommendations for NRBG component interaction.....	33
9	Overview and requirements for a DRBG.....	33
9.1	DRBG overview.....	33
9.2	Functional model of a DRBG.....	33
9.3	DRBG randomness source.....	36
9.3.1	Primary randomness source for a DRBG.....	36
9.3.2	Generating seed values for a DRBG.....	37
9.3.3	Additional randomness sources for a DRBG.....	38
9.3.4	Hybrid DRBGs.....	38
9.4	Additional inputs for a DRBG.....	38
9.5	Internal state for a DRBG.....	39
9.6	Internal state transition function for a DRBG.....	39
9.7	Output generation function for a DRBG.....	40
9.8	Health tests for a DRBG.....	40
9.8.1	DRBG health tests overview.....	40
9.8.2	DRBG health test.....	41
9.8.3	DRBG deterministic algorithm test.....	41
9.8.4	DRBG software/firmware integrity test.....	41
9.8.5	DRBG critical functions test.....	41
9.8.6	DRBG software/firmware load test.....	41
9.8.7	DRBG manual key entry test.....	42
9.8.8	Continuous tests on noise sources in entropy sources.....	42
9.9	Additional requirements for DRBG keys.....	42
	Annex A (normative) Combining RBGs.....	44
	Annex B (normative) Conversion methods for random number generation.....	45
	Annex C (informative) Deterministic random bit generators.....	48
	Annex D (informative) NRBG examples.....	75
	Annex E (informative) Security considerations.....	84
	Annex F (informative) Discussion on the estimation of entropy.....	88
	Annex G (informative) RBG assurance.....	89
	Annex H (normative) RBG boundaries.....	90
	Annex I (informative) Rationale for the design of statistical tests.....	92
	Bibliography.....	93

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 or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and <https://patents.iec.ch>. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

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. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC/JTC 1 *Information Technology*, Subcommittee SC 27, *Information security, cybersecurity and privacy protection*.

This third edition cancels and replaces the second edition (ISO/IEC 18031:2011), which has been technically revised. It also incorporates the Amendment ISO/IEC 18031:2011/Amd 1:2017 and the Technical Corrigendum ISO/IEC 18031:2011/Cor 1:2014.

The main changes are as follows:

- removal of the MQ_DRBG, Micali-Schnorr DRBG, Dual_EC_DRBG and SHA-1;
- addition and harmonization of the terms and definitions in [Clause 3](#);
- addition of conversion methods for random number generation;
- update of the requirements for DRBGs and NRBGs.

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 and www.iec.ch/national-committees.

Introduction

This document sets out specific requirements that, when met, will result in the development of a random bit generator that can be applicable to cryptographic applications.

Numerous cryptographic applications involve the use of random bits. These cryptographic applications include the following:

- random keys and initialization values (*IVs*) for encryption,
- random keys for keyed MAC algorithms,
- random private keys for digital signature algorithms,
- random values to be used in entity authentication mechanisms,
- random values to be used in key-establishment protocols,
- random PINs and passwords,
- nonces.

The purpose of this document is to establish a conceptual model, terminology and requirements related to the building blocks and properties of systems used for random bit generation in or for cryptographic applications.

It is possible to categorize random bit generators into two types, namely, non-deterministic and deterministic random bit generators.

A non-deterministic random bit generator can be defined as a random bit generating mechanism that continuously uses a source of entropy to generate a random bit stream.

A deterministic random bit generator can be defined as a bit generating mechanism that uses deterministic mechanisms such as cryptographic algorithms to generate a random bit stream. In this type of bit stream generation, there is a specific input (normally called a seed) and perhaps some optional input, which, depending on its application, can either be publicly available or not. The seed is processed by a function which provides an output.

NOTE This document also discusses hybrid random bit generators, which incorporate elements of both non-deterministic and deterministic generators.

In this document, variable symbols and variable descriptive terms are given in italic font.

Information technology — Security techniques — Random bit generation

1 Scope

This document specifies a conceptual model for a random bit generator for cryptographic purposes, together with the elements of this model.

This document specifies the characteristics of the main elements required for both non-deterministic and deterministic random bit generators. It also establishes the security requirements for both non-deterministic and deterministic random bit generators.

Techniques for statistical testing of random bit generators for the purposes of independent verification or validation and detailed designs for such generators are outside the scope of 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 9797-2, *Information security — Message authentication codes (MACs) — Part 2: Mechanisms using a dedicated hash-function*

ISO/IEC 10118-3, *IT Security techniques — Hash-functions — Part 3: Dedicated hash-functions*

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

ISO/IEC 20543, *Information technology — Security techniques — Test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408*

ISO/IEC 29192-5, *Information technology — Security techniques — Lightweight cryptography — Part 5: Hash-functions*

3 Terms and definitions

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

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

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

3.1 algorithm

clearly specified mathematical process for the computation of a set of rules that, if followed, will give a prescribed result

3.2 backward secrecy

assurance that previous *random bit generator (RBG)* (3.40) output values cannot be determined with practical computational effort from knowledge of current or subsequent (future) output values

3.3

bit stream

continuous output of bits from a device or mechanism

3.4

bit-string

finite sequence of ones and zeroes

3.5

block cipher

symmetric encryption system with the property that the encryption *algorithm* (3.1) operates on a block of plaintext to yield a block of ciphertext

Note 1 to entry: The block ciphers standardized in ISO/IEC 18033-3 have the property that plaintext and ciphertext blocks are of the same length.

[SOURCE: ISO/IEC 18033-1:2021, 3.6]

3.6

conditioning

method of processing the data to reduce bias and/or ensure that the entropy rate of the output is no less than some specified amount

3.7

cryptographic boundary

explicitly defined perimeter that establishes the boundary of all components (i.e. a set of hardware, software, or firmware) of the cryptographic module

[SOURCE: ISO/IEC TS 30104:2015, 3.4]

3.8

deterministic algorithm

algorithm (3.1) that, when given a particular input, always produces the same output

3.9

deterministic random bit generator

DRBG

random bit generator (3.40) that produces a random-appearing sequence of bits by applying a *deterministic algorithm* (3.8) to a suitably random initial value called a seed and, possibly, some secondary inputs upon which the security of the random bit generator does not depend

Note 1 to entry: In particular, non-deterministic sources may also form part of these secondary inputs.

3.10

deterministic random bit generator boundary

DRBG boundary

conceptual boundary that is used to explain the operations of a *deterministic random bit generator* (DRBG) (3.9) and its interaction with and relation to other processes

3.11

enhanced backward secrecy

assurance that the knowledge of the current internal state of a *random bit generator* (3.40) does not allow an adversary to derive, with practical computational effort, knowledge about previous output values

Note 1 to entry: Another term often found in the literature that is similar to enhanced backward secrecy is backtracking resistance.

[SOURCE: ISO/IEC 20543:2019, 3.6 modified — Note 1 to entry replaced and commas added after “derive” and “effort”]

3.12**enhanced forward secrecy**

assurance that it is not feasible to determine (future) output values after sufficient *entropy* (3.13) has been mixed into the internal state, given knowledge of the current and previous internal state

Note 1 to entry: *Deterministic random bit generators* (3.9) are unable to achieve enhanced forward secrecy without the insertion of sufficient fresh entropy at the end of the sentence. Unlike forward and backward secrecy as well as *enhanced backward secrecy* (3.11), enhanced forward secrecy rests entirely on the ability of a continuous reseeding process to supply as much entropy as 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, it is possible to consider a random bit generator design with a random source that produces (at each invocation) a 128-bit random string R with an estimated 128 bits of min entropy, with a 512-bit internal state $S(n)$, an internal state transition function giving $S(n+1) := \text{SHA3-512}(S(n)||R)$, and an output generation function applying SHAKE-256 on $S(n)||R$ with up to 1 024 bits of output per invocation.

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

Note 4 to entry: If insufficient entropy is mixed into the internal state, enhanced forward secrecy is not achieved, and it is possible that a compromise of the internal state is not cured by the additional entropy due to “iterative guessing attacks.”

3.13**entropy**

measure of the disorder, randomness or variability in a closed system

Note 1 to entry: The entropy of a random variable X is a mathematical measure of the amount of information provided by an observation of X .

3.14**entropy rate**

assessed amount of *entropy* (3.13) in a *bit-string* (3.4) divided by the number of bits in the *bit-string* (3.4)

3.15**entropy source**

combination of a noise source, health tests, and optional *conditioning* (3.6) that produce random *bit-strings* (3.4) for use by a *random bit generator* (3.40)

Note 1 to entry: Entropy sources can be physical or non-physical, depending on the noise source.

3.16**forward secrecy**

assurance that the knowledge of subsequent (future) output values cannot be determined with practical computational effort from current or previous output values

Note 1 to entry: This definition is specific to the context of *random bit generators* (3.40). It should not be confused with “forward secrecy” defined in the ISO/IEC 11770 series for key management.

3.17**full entropy**

entropy rate (3.14) that is practically close to 1

3.18**full entropy source**

source that produces output with *full entropy* (3.17)

3.19**full forward secrecy**

property of a *deterministic random bit generator (DRBG)* (3.9) in which sufficient *entropy* (3.13) is provided during every generation process to meet the security requirements for the security strength to be supported by the DRBG

3.20

glass box

idealized mechanism that accepts inputs and produces outputs and is designed such that an observer can determine exactly how the outputs are computed from the inputs

3.21

hash-function

function that maps strings of bits of variable (but usually upper-bounded) length to fixed-length strings of bits, satisfying the following three properties:

- for a given output, it is computationally infeasible to find an input that maps to this output;
- for a given input, it is computationally infeasible to find a second input that maps to the same output;
- it is computationally infeasible to find any two distinct inputs which map to the same output

Note 1 to entry: Computational feasibility depends on the specific security requirements and environment. Refer to ISO/IEC 10118-1:2016, Annex C.

[SOURCE: ISO/IEC 10118-1:2016, 3.4 modified – third bullet point added to the definition; "two properties" changed to "three properties".]

3.22

hybrid DRBG

hybrid deterministic random bit generator

deterministic random bit generator (DRBG) (3.9) that is capable of accepting external input values during its operation

3.23

hybrid NRBG

hybrid non-deterministic random bit generator

random bit generator (3.40) with non-deterministic input from a noise source and that uses complex, stateful post-processing (e.g. cryptographic processing)

Note 1 to entry: A hybrid *non-deterministic random bit generator (NRBG)* (3.30) can be physical or non-physical depending on its *entropy* (3.13) source.

3.24

independent and identically distributed

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

[SOURCE: ISO/IEC 20543:2019, 3.14]

3.25

initialization value

value used in defining the starting point of a cryptographic *algorithm* (3.1)

Note 1 to entry: Examples of cryptographic algorithms that use an initialization value include hash-functions and encryption algorithms.

3.26

Kerckhoffs's box

idealized cryptosystem where the design and public keys are known to an adversary, but in which there are secret keys and/or other private information that are not known to an adversary

3.27**known-answer test**

method of testing a deterministic mechanism where a given input is processed by the mechanism, and the resulting output is then compared to a corresponding known value

Note 1 to entry: Known-answer testing of a deterministic mechanism may also include testing the integrity of the software that implements the deterministic mechanism. For example, if the software implementing the deterministic mechanism is digitally signed, then the signature can be recalculated and compared to the known signature value.

3.28**min-entropy**

lower bound of *entropy* (3.13) that is useful in determining a worst-case estimate of sampled entropy

Note 1 to entry: The bit-string X (or more precisely, the corresponding random variable that models random bit-strings of this type) has min-entropy k , if k is the largest value such that $\Pr[X = x] \leq 2^{-k}$. That is, X contains k bits of min-entropy or randomness.

3.29**noise source**

component of an entropy source that contains the non-deterministic, entropy-producing activity (e.g. thermal noise or hard-drive seek times)

Note 1 to entry: A noise source does not output digital data, unlike a physical or non-physical noise source.

3.30**non-deterministic random bit generator****NRBG**

random bit generator (3.40) that samples one or multiple entropy sources and, if operating correctly, has an output that is expected to be unpredictable for attackers with unbounded computational capabilities

[SOURCE: ISO/IEC 20543:2019, 3.19, modified — “continuously” and “over short timescales” have been deleted; “one or” has been added.]

3.31**non-physical entropy source**

entropy source in which *entropy* (3.13) is credited from one or more *non-physical noise sources* (3.33) e.g. random-access memory (RAM) content or thread number

3.32**non-physical noise source**

noise source that exploits system data, peripheral data, or user interaction and outputs digital data

3.33**one-way function**

function with the property that it is easy to compute the output for a given input but it is computationally infeasible to find an input that maps to a given output

[SOURCE: ISO/IEC 11770-3:2021, 3.30]

3.34**output generation function**

function in a *random bit generator* (RBG) (3.40) that computes the output of the RBG from the internal state of the RBG

3.35**physical entropy source**

entropy source in which *entropy* (3.13) is counted only from a *physical noise source* (3.36)

3.36**physical noise source**

noise source (3.29) that exploits physical phenomena from dedicated hardware or physical experiments and produces digitized random data

3.37**protection boundary**

physical or conceptual perimeter that defines the secure domain into which an attacker cannot observe or influence the process in a malicious way (according to a chosen threat model)

3.38**pure DRBG****pure deterministic random bit generator**

deterministic random bit generator (3.9) whose only external input is an initial seed

3.39**pure NRBG****pure non-deterministic random bit generator**

random bit generator (3.40) that takes its non-deterministic input from a noise source, and for which any post-processing is non-cryptographic or stateless cryptographic

3.40**random bit generator****RBG**

device or *algorithm* (3.1) that outputs a sequence of bits that appears to be statistically independent and unbiased

3.41**randomness source**

source of randomness for a *random bit generator* (3.40) that can be an entropy source, a *non-deterministic random bit generator* (3.30), or a *deterministic random bit generator* (3.9)

3.42**reseeding**

specialized internal state transition function that updates the internal state in the event that a new seed value is supplied by either computing a new internal state from the current internal state and the new seed value, or by replacing the internal state based only on the new seed value

Note 1 to entry: The term reseeding is used in a variety of ways in the literature. In this document, reseeding refers to a mechanism that replaces the current value of the internal state by a fresh value, which can either (partially) depend on the current value, or not. Elsewhere, a distinction is sometimes made between reseeding and seed update. In such cases, the term reseeding is only used for mechanisms that replace the internal state by a new value that does not depend on the current value (essentially a new seeding process), and the term seed update is used for a mechanism that computes the new internal state as a function of its current value and other (usually non-deterministic) data (see 9.6, item 3).

3.43**secret parameter**

input to the *random bit generator* (3.40) that provides additional randomness in the event of a failure or compromise of the randomness source

Note 1 to entry: In practice, the secret parameter is often a key.

Note 2 to entry: The secret parameter is only useful if it has sufficient randomness.

Note 3 to entry: The secret parameter is not the same as a seed.

3.44**security strength**

number associated with the amount of work (i.e. the number of operations of some sort) that is required to break a cryptographic *algorithm* (3.1) or system

Note 1 to entry: If the security strength associated with an algorithm or system is n bits, then it is expected that (roughly) 2^n basic operations are required to break it.

3.45

seed

bit-string (3.4) that is used as input to initialize the internal state of a *deterministic random bit generator (DRBG)* (3.9)

Note 1 to entry: The seed will determine a portion of the state of the DRBG.

3.46

seedlife

period of time between initializing or reseeding the *deterministic random bit generator (DRBG)* (3.9) with one *seed* (3.45) and reseeding that DRBG with a different seed

3.47

seed material

data used to form a seed for input to a *deterministic random bit generator (DRBG)* (3.9)

Note 1 to entry: The term is often used to refer to the bit stream provided by a randomness source.

3.48

seed value

input *bit-string* (3.4) from a randomness source that provides *entropy* (3.13) for a *deterministic random bit generator* (3.9)

3.49

state

condition of a *random bit generator* (3.41) or any part thereof at a particular instant

3.50

stochastic model

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

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. This is because cryptographic post-processing can render even very low entropy data which is 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 bits.

[SOURCE: ISO/IEC 20543:2019, 3.30 modified — "from the noise source" added to the definition; in Note 1 to entry, the last word "numbers" has been replaced by "bits"; the last sentence has been split into two.]

3.51

working state

subset of the internal state that is used by a *deterministic random bit generator* (3.9) mechanism to produce pseudorandom bits at a given point in time

4 Symbols

For the purposes of this document, the following symbols apply.

0^l	all-zero bit-string of length l
$\text{Pr}[x]$	probability of occurrence of x
IV	initialization value
$\lceil X \rceil$	Ceiling: the smallest integer greater than or equal to X . For example, $\lceil 5 \rceil = 5$, and $\lceil 5,3 \rceil = 6$.
$X \oplus Y$	bitwise exclusive-or (also bitwise addition mod 2) of bit-strings X and Y of the same length
$X \parallel Y$	concatenation of two separate bit-strings X and Y in that order
$ a $	the length in bits of string a
$x \bmod n$	The unique remainder r , $0 \leq r \leq n-1$, when integer x is divided by n . For example, $23 \bmod 7 = 2$.

5 Properties and requirements of a random bit generator

5.1 Properties of a random bit generator

The properties of randomness can be demonstrated by tossing a coin in the air and observing which side is uppermost when it lands, where one side is called “heads” (H) and the other is called “tails” (T). A coin also has a rim, but the probability that a coin can land on its rim is so unlikely an occurrence that, for the purpose of this demonstration, it can be ignored.

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 idealized coin toss described above. The result of each coin flip is:

- a) Unpredictable: Before the flip, it is unknown whether the coin will land showing a head or a tail. Also, if that flip is kept secret, it is not possible to determine what the flip was if any subsequent flip outcome is known. The unpredictability after the flip depends on whether the observer can observe the coin flip or not. The notion of entropy quantifies the amount of unpredictability or uncertainty relative to an observer and will be discussed more thoroughly later in this document;
- b) Unbiased: that is, each potential outcome has the same chance of occurring; and
- c) Independent: the coin flip is said to be memoryless; whatever happened before the current flip does not influence it.

Such a series of idealized coin flips is directly applicable to a random bit generator (RBG). The RBGs specified in this document try to simulate a series of idealized coin flips.

As indicated above, unpredictability is a required property of an RBG. Predicting the output of a properly implemented and working RBG is not expected to be possible.

The decision whether to incorporate enhanced forward secrecy (which is an optional feature of an RBG) is determined by the needs of the consuming application. The following factors should be considered when deciding to incorporate enhanced forward secrecy:

- 1) An RBG without enhanced forward secrecy can be secure for a consuming application if exposure of the internal state of the RBG is unlikely or if any exposure is mitigated by replacement of the RBG. For example, a smart card may be initialized at the point of manufacture with sufficient entropy in the seed, and the smart card is set to expire after a limited time (e.g. two or three years). Compromise of a smart

card should be difficult and evident to the user. Therefore, a compromised smart card can be replaced with a new one.

- 2) An RBG should be capable of providing enhanced forward secrecy if the consuming application has identified a risk that the internal state can be exposed without detection. An example is a system which is compromised by a cyber attack that accessed the internal state of an RBG. If the compromise is not discovered or properly remediated, the system can continue performing cryptographic operations relying upon the RBG. Enhanced forward secrecy can enable the RBG to return to a secure internal state.

5.2 Requirements of an RBG

All RBGs specified in this document, both deterministic and non-deterministic, shall satisfy the requirements provided below in 1) to 12). These requirements are fundamental to the security of many cryptographic mechanisms that require random input and help prevent misuse in consuming applications.

The threshold between feasible and infeasible shall be determined by the overall requirement for the minimum acceptable strength of cryptographic security that is required by the application.

- 1) RBGs for general cryptographic applications shall output bit sequences that are indistinguishable from uniformly distributed and independently generated bit sequences under standard cryptographic assumptions. Suitable standard assumptions can be the hardness of breaking a block cipher or the difficulty of inverting a particular one-way function. The seeding process (and, optionally, any re-seeding that can happen during their operation) shall provide entropy that is commensurate with the intended security strength of the RBG, which shall be assessed using the evaluation methodology laid out in ISO/IEC 20543. See [Annex G](#) for RBG assurance.
- 2) Both the deterministic components and the entropy sources of RBGs shall be designed to resist side channel and fault attacks that are potentially feasible in the application environment.
- 3) From the perspective of an adversary, the internal state of the RBG shall always contain at least 128 bits of entropy and at least as much as the security strength of the consuming application requires. The perspective of an adversary includes all information that the adversary can effectively exploit, i.e. by means of classical computational attacks, quantum attacks, or relevant side channel and fault attacks.
- 4) The RBG shall not generate bits unless the generator has been assessed to possess sufficient entropy. The criteria for sufficiency shall be the greater of the requirements of this document and the requirements of the consuming application.
- 5) On detection of an error, the RBG shall either a) enter a permanent error state, or b) not output any random bits and be able to recover from a loss or compromise of entropy if the permanent error state is deemed unacceptable for the application requirements. These requirements may be satisfied procedurally or inherently in the design.

NOTE 1 See [8.8](#) and [9.8](#) for NRBG and DRBG information on RBG errors and health tests.

- 6) The design and implementation of an RBG shall have a defined protection boundary. The protection boundary shall be as specified in ISO/IEC 19790 (see [Annex H](#)).
- 7) The probability that the RBG can “misbehave” in some pathological way that violates the output requirements (e.g. give an output that is constant or has a short cycle, i.e. looping such that the same output is repeated) shall be sufficiently small. This implies that the probability of error should be consistent with the overall confidence in correct operation that is required of the RBG, which is not necessarily the same as the required strength of cryptographic security.
- 8) The RBG design shall include methods to prohibit predictable influence, manipulation, or predicting the output of the RBG by observing the generator's physical characteristics (e.g. power consumption, timing, or emissions).
- 9) The implementation shall be designed to allow validation, including specific design assertions about what the RBG is not intended to do. The validation of an RBG means that the RBG behaves as expected, not just during normal operation but also at the boundaries of the intended operational conditions.

With respect to code-dependent RBGs, security-relevant branches in the code governing behaviour in exceptional conditions (e.g. initialization, failed health tests) shall be validated by deliberately forcing all error conditions to occur during validation testing.

- 10) There shall be design evidence (theoretical, empirical, or both) to support all security requirements for the RBG, including protection from misbehaviour.
- 11) An RBG shall have enhanced backward secrecy. This means that given all accessible information about the RBG (comprising some subset of inputs, algorithms and outputs), it shall be computationally infeasible (up to the specified security strength) to compute or otherwise determine any previous output bit.
- 12) An RBG shall have forward secrecy.

NOTE 2 Requirement 11 (enhanced backward secrecy) implies backward secrecy.

In some circumstances it is desirable to combine RBGs; if RBGs are combined then the method used shall be in accordance with one of the approaches specified in [Annex A](#).

5.3 Additional information for an RBG

Additional information for an RBG is as follows.

- 1) If the RBG is capable of operating in more than one mode (e.g. operational mode or test mode), the RBG should, upon request, return information about the mode in which it is operating.
- 2) A consuming application may require that the RBG be run in a test mode, for example, when using a non-secret seed for generating bits for testing purposes only. When an RBG is in the test mode, the RBG shall not be capable of being used to generate (secret) operational output. The RBG shall not operate in the test mode using a seed to be used in the operational mode.
- 3) Considered as a glass box, an RBG may provide enhanced forward secrecy. If supported, it means that given all accessible information about the RBG (comprising some subset of inputs, algorithms, internal state, and outputs), it shall be infeasible (up to the specified security strength) to compute or predict any future output bit at the time that enhanced forward secrecy was requested.

6 RBG model

6.1 Conceptual functional model for random bit generation

[Figure 1](#) depicts a conceptual functional model for random bit generation. This model and the associated requirements and objectives specify what RBGs are expected to achieve without constraining or mandating how the implementation shall be done, regardless of whether the RBG is non-deterministic or deterministic. Since not all of the important aspects of random bit generation can be algorithmically specified, this functional view of random bit generation is central to the definition of RBGs.

The functional model encompasses everything required to produce random bits. This holistic approach is necessary to ensure that RBG output will be as random as desired.

When an RBG product or system component does not directly incorporate all of the functional components or address all of the functional requirements specified in this document, that RBG can still conform to this document if the RBG component is used within a system that supplies the missing elements and satisfies the remaining requirements. Such an RBG is considered to possess residual functional requirements. Those residual requirements become a system prerequisite for use of the RBG.

NOTE [Figure 1](#) shows that there are RBGs that do not incorporate all functional components. Physical NRBGs, for example, do not necessarily need additional inputs.

6.2 RBG basic components

6.2.1 Introduction to the RBG basic components

The model as shown in [Figure 1](#) has six basic components; however, it can be possible to meet all the functional requirements without incorporating all the basic components. They are:

- a) the randomness source;
- b) additional inputs;
- c) the internal state;
- d) the internal state transition function;
- e) the output generation function; and
- f) health tests.

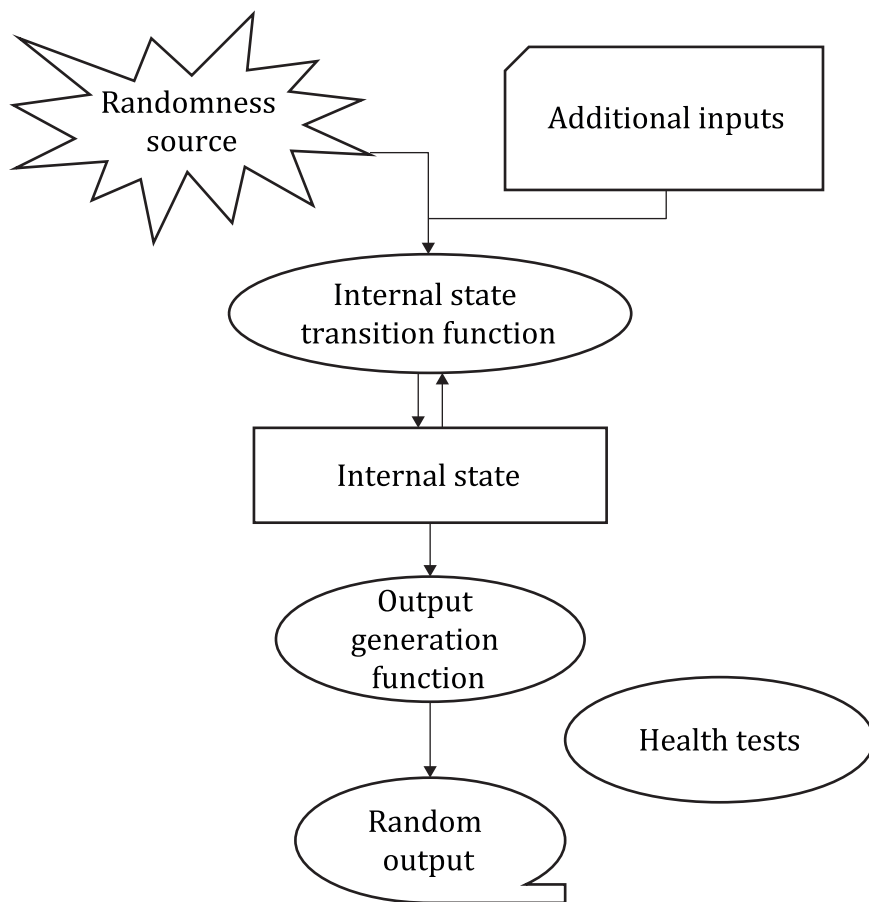


Figure 1 — RBG functional model

6.2.2 Randomness source

6.2.2.1 Randomness source overview

The randomness source is the source of unpredictable bits. These bits can be biased and in many cases, can be dependent on one another to some extent. In the case of a DRBG, the randomness source can be a distinct and possibly remote NRBG. However, for an NRBG, the randomness source component is an entropy source that has a noise source with some entropy-producing activity, a method for detecting this activity, and a digitization mechanism.

The randomness source shall produce bits with non-zero entropy. It is the only model component that produces entropy. For an NRBG, the entropy source encompasses everything that is not deterministic in the RBG model, along with whatever is required for the source to manifest itself in bits (as opposed to analogue signals or other non-digital activity). In a DRBG, the randomness source has traditionally been left unspecified, even though the unpredictability of the generator ultimately depends on the entropy of a special input called a seed. The model does not assume anything about the predictability, bias, or independence of the bits produced by the source. The only assumption is that the randomness source has non-zero entropy. However, all RBG designs are required to establish specific conditions that shall hold for the design to work.

6.2.2.2 Requirements for an entropy source

The requirements for the entropy source used by an RBG are as follows.

- 1) The entropy source shall be based upon well-established principles, or extensively characterized behaviour.
- 2) The entropy rate shall be assessable, or the collection shall be self-regulating, so that the amount of entropy per collection unit or event reliably achieves or exceeds a designed lower bound.
- 3) The entropy source shall be designed so that any manipulation (such as the ability to control the entropy source), influence (such as the ability to bias the entropy source), or its observation by some unauthorized external body shall be minimized according to the requirements of the RBG design.
- 4) Loss or severe degradation of an entropy source shall be detectable.

Methods for evaluating the process for generating random bits from the entropy source are specified in ISO/IEC 20543.

6.2.3 Additional inputs

6.2.3.1 Additional inputs overview

Additional inputs can be used to personalize the output of an RBG based on parameters such as the time/date, the intended use of the data, and other user-supplied information or data, or both, that is necessary to control the internal functionality of the RBG. These inputs typically include either commands or time-variant parameters, or both, such as a clock or internal counter. The additional input may be used to insert new entropy into the RBG. The entropy of the output shall not depend solely upon any properties of the additional inputs.

Additional input can be inserted during instantiation, generation, and reseeding.

6.2.3.2 A requirement for additional inputs

The form and use of additional inputs shall not degrade the entropy of the RBG.

6.2.4 Internal state

6.2.4.1 Internal state overview

The internal state contains the memory of the RBG and includes both secret and non-secret information. It can include a seed, a counter, a clock value, user input, the security strength of the generation process, the security strength of output, etc. The internal state consists of all the parameters, variables, and other stored values that the deterministic parts of the RBG use or act upon. The internal state is usually implicit in the specification of the deterministic functions for which it exists; the model makes the state explicit in order to better address the security issues that it can affect.

Functionally, the internal state plays two different roles. The first is to ensure that the output appears random even when the entropy of the input is insufficient to generate a truly random sequence of that length. This randomness property is ensured by the output generation function. Since the internal state transition

functions and the output generation function are deterministic functions, and since deterministic functions have the property that they always give the same output for the same input, it can be necessary to have a state variable that is constructed to change with each block of RBG output. The purpose of this variable is to ensure that the output appears random even though no, or little, true entropy has been added to the system (e.g. since it was reseeded with fresh entropy).

This is particularly relevant for a DRBG. For a DRBG, unless the seed or secret parameter is updated by an external device, it is incumbent upon the design that only in the remotest possibility can a repeat sequence of the state occur. Therefore, the seed (or the secret parameter) shall be updated periodically or the device shall become inoperable after a predetermined period of time.

The second role of the internal state is to parameterize either the deterministic internal state transition functions or the output generation function, or both, in such a way that when this parameter is unknown, the deterministic function exhibits the properties required of it. For example, it can parameterize the output generation function in such a way that the internal state cannot be deduced from the output when the secret parameter is unknown. The secret parameter is typically one or more cryptographic keys, and these shall be generated in a suitably random fashion. When such a secret parameter exists, it shall have a well-defined security life cycle, which includes provision for its generation, distribution, update and destruction.

NOTE 1 It is optional for RBGs to have a secret parameter.

NOTE 2 Key management concerns are discussed in ISO/IEC 11770-1.

6.2.4.2 Requirements for the internal state

The requirements for the internal state of an RBG are as follows.

- 1) The internal state shall be protected in a manner that is consistent with the use and sensitivity of the output.
- 2) The internal state shall be functionally maintained properly across power failures, reboots, etc. or shall regain a secure condition before any output is generated (i.e. either the integrity of the internal state shall be assured, or the internal state shall be re-initialized).
- 3) The state elements that accumulate or carry entropy for the RBG shall have at least x bits of entropy, where x is the desired security strength expressed in bits of security. The state elements should accumulate more than the minimum number of bits of entropy for reasons of assurance and to reduce the risk of using identical values in two different cryptosystems. Generally, it is recommended to keep the entropy in the internal state somewhat larger than x , e.g. $x+64$ typically provides a sufficient margin, while if very large amounts of pre-computation and memory are available to an adversary, a size of $2x$ should be used (e.g. Reference [19]).
- 4) The secret portion of the internal state shall have a specified finite period after which the RBG shall either cease operation or be reseeded with sufficient additional entropy. Operations using an old seed shall cease after the specified period elapses unless a new seed is provided. See ISO/IEC 11770-1 for further details on the key life cycle.
- 5) A specific internal state shall not be reused, except strictly by chance. This implies that the same seed shall not be deliberately input to a different instance of a DRBG.

6.2.4.3 Recommendation for the internal state

The internal states that are used to produce public data, (e.g. nonces and initialization values), should be fully independent from the states used to produce secret data such as cryptographic keys.

6.2.5 Internal state transition functions

6.2.5.1 Internal state transition functions overview

Internal state transition functions alter the internal state of the RBG. In a DRBG, the internal state transition functions are composed of one or more cryptographic algorithms that are part of the generator

specification. The internal state transition functions encompass everything in the RBG that can set or alter the internal state. As a matter of convention, neither the randomness source nor any other RBG data inputs are considered to act directly on the internal state. Instead, the randomness source and other RBG data inputs are arguments to functions that act on the internal state.

Since the internal state can comprise several distinct components, the internal state transition functions are likely to be composed of several distinct collections of functions, distinguished by the state component upon which they act. The functions in such a collection can be as trivial as the identity function or as complex as a cryptographic function. For a clock or counter register, the function that sequences the register as a clock or counter is a member of the internal state transition functions. While the output of these functions is always directed to a component of the internal state, the input may be from the randomness source, another external RBG input, various components of the internal state, or any combination thereof.

The internal state transition functions shall also allow for the secure manipulation (e.g. updating) of the secret parameter. Access to these functions shall be strictly controlled.

In a DRBG, the internal state transition functions are responsible for much of the security of the generator, including how much output the RBG may produce for a given randomness source input.

In an NRBG, the internal state transition functions determine how the entropy collection affects the internal state of the RBG. These functions can work in a variety of ways: they can be simple, stateless algorithms, or can combine previous states with newly collected entropy in order to accumulate entropy.

6.2.5.2 Requirements for the internal state transition functions

The requirements for the internal state transition functions of an RBG are as follows.

- 1) The internal state transition functions shall be verifiable via a known-answer test.
- 2) The internal state transition functions shall, over time, depend on all the entropy carried by the internal state.
- 3) The internal state transition functions shall resist observation and analysis via power consumption, timing, radiation emissions, or other side channels, as appropriate.
- 4) It shall not be feasible (either intentionally or unintentionally) to cause the internal state transition functions to return to a prior state in normal operation (this excludes testing and authorized verification of the RBG output).

6.2.5.3 Recommendation for the internal state transition functions

The internal state transition functions should enable the RBG to recover from the compromise of the internal state (i.e. provide enhanced forward secrecy) through a periodic incorporation of entropy.

6.2.6 Output generation function

6.2.6.1 Output generation function overview

The output generation function (OGF) acts on the internal state to produce output bits, as requested by the consuming application. A request to the OGF contains (at least implicitly) the number of output bits required and the required minimum security strength for those bits. Such a request can cause the internal state to be updated by the internal state transition functions. The OGF accepts the working state component of the internal state as input and produces the RBG output. It can be as complex as a cryptographic function. The OGF can format or block the output to conform to an external interface convention. The OGF should be a “one-way” function, so that it is hard to invert the OGF in order to recover part of the internal state.

The OGF is deterministic and always produces the same output for any particular value of the internal state. Therefore, there is an important relationship between the OGF and the internal state transition function. The internal state transition function shall always be invoked at least once to update at least the working state between successive actions of the output generation function.

6.2.6.2 Requirements for the output generation function

The requirements for the output generation function of an RBG are as follows.

- 1) The output generation function shall be deterministic (given all inputs) and shall be testable by a known-answer test. The known-answer test output shall be separated from operational output.
- 2) The output generation function shall use information from the internal state that contains sufficient entropy to support the requested security strength.
- 3) The output shall be inhibited until the internal state exhibits/obtains sufficient assessed entropy.
- 4) Once a particular internal state has been used for output, the internal state shall be changed in order to produce more output.
- 5) The output generation function shall resist observation and analysis via power consumption, timing, radiation emissions, or other side channels, as appropriate.
- 6) The output generation function shall resist attempts to interfere with its normal operation (i.e. fault attacks) as required by the operational environment of the RBG.
- 7) The output generation function shall protect the internal state, so that the analysis of RBG outputs does not reveal useful information about the internal state.

6.2.7 Health test

6.2.7.1 Health test overview

Health tests are concerned with assessing the health of the RBG. Health tests include functions that:

- assess the entropy of the input;
- assess the statistical quality of the output; and
- check whether the internal functions have been compromised.

The internal functionality (i.e. the "health" of the deterministic portions of the RBG) can most effectively be checked using statistical tests on the noise source and known-answer tests on deterministic components. The health of the entropy source and the quality of the output shall be checked using statistical techniques or tests.

Detailed information on assessing the health of NRBGs and DRBGs can be found in [8.8](#) and [9.8](#), respectively.

6.2.7.2 Requirements for the health test

The requirements for health tests of an RBG are as follows.

- 1) An RBG shall be designed to permit testing that ensures the generator operates correctly.
- 2) When an RBG fails a test, the RBG shall enter an error state and output an error indicator. The RBG shall not perform any random output generation while in the error state (see [8.8.4](#)). The RBG shall be able to recover and return to the operational state (e.g. repair and test) if the permanent error state is deemed unacceptable for the application requirements.

7 Types of RBGs

7.1 Introduction to the types of RBGs

Every RBG shall have a primary randomness source. RBGs are classified into two basic types, depending on the nature of their primary randomness source. The primary randomness source is either an entropy source or a seed value.

If randomness is obtained from a system from which any amount of entropy can be extracted by sampling (provided that this system runs for a sufficiently long period of time), then it is deemed to be non-deterministic. In particular, no deterministic algorithm should be able to predict the output of a non-deterministic randomness source. The primary randomness source for an NRBG is an entropy source, which provides non-deterministic input.

If randomness is obtained from a deterministic process, then the output of that process is deemed to be deterministic. A DRBG is instantiated with a seed value obtained as specified in [9.3.2](#). The DRBG produces deterministic output if no non-deterministic input is provided during the generation process. An RBG may also utilize additional randomness sources that can be either deterministic or non-deterministic. The amount of entropy required by the consuming application shall be provided by an appropriate combination of these randomness sources.

Care shall be taken to ensure that an adversary cannot gain sufficient control over the randomness source in order to degrade the entropy in the RBG output.

This document is concerned with the generation of sequences of random bits. Some applications can require the generation of sequences of random numbers. [Annex B](#) specifies the conversion methods for generating sequences of random numbers from sequences of random bits which shall be used to generate sequences of random numbers.

7.2 Non-deterministic random bit generators

Non-deterministic random bit generators (NRBGs) can be further specified into sub-classes depending on the nature of their entropy source.

A physical entropy source is one in which dedicated hardware is used to measure the physical characteristics of a sequence of events in the real world. Such devices typically continue to provide an output, provided that power is applied to the measuring device. Examples of physical entropy sources and their measurement include measuring the time between radioactive emissions of an unstable atom and measuring the noise characteristics of an unstable or “noisy” diode. For more details about the requirements of physical entropy sources, see [8.3.2](#).

A non-physical entropy source is any entropy source that is not a physical entropy source. Examples of measuring the output of non-physical entropy sources include measuring the time between key presses or sampling data in regularly used portions of random-access memory (RAM) memory. For more details about the requirements of non-physical entropy sources, see [8.3.3](#).

A physical NRBG is an RBG with a physical entropy source. A non-physical NRBG is an RBG with a non-physical entropy source.

A pure NRBG obtains input only from a physical or non-physical entropy source.

Hybrid NRBGs obtain entropy from entropy sources and seed values from deterministic sources. A hybrid NRBG shall satisfy all the security criteria imposed on pure NRBGs and shall also satisfy certain extra security requirements (see [8.3.5](#)) that are typical for DRBGs. The advantage of using a seed value from another randomness source as an additional input to the NRBG is that it allows the output of the NRBG to be parameterized by the user.

Multiple users can use the same NRBG and its associated entropy source as long as the entropy is not reused (e.g. the same entropy bit-string is not provided to more than one user or to the same user repeatedly).

NOTE 1 Additional information on classes of RBGs can be found in References [\[14\]](#) and [\[15\]](#).

NOTE 2 The use of a secret seed value, for example, can be an additional DRBG-type security feature. If the NRBG meets the requirements of this document even with a simple state transition function and a simple output function (see [8.2](#) and [E.1](#)), a cryptographic state transition function or a cryptographic output function can be an additional DRBG-type security feature.

7.3 Deterministic random bit generators

Deterministic random bit generators (DRBGs) can also be classified as either pure or hybrid. A DRBG is considered “pure” if the only external input is the initial seed and is considered “hybrid” if it is capable of accepting external input values during operation.

A hybrid DRBG shall satisfy all the security conditions imposed on pure DRBGs and shall also satisfy certain extra security requirements (see 9.3.4).

The security of a pure DRBG depends only on its initial seed and its cryptographic algorithm(s) for producing output. The security of a hybrid DRBG also depends on the randomness and frequency of its external input.

NOTE 1 If an RBG can be viewed as both a hybrid NRBG or alternatively as a hybrid DRBG, it is denoted as hybrid RBG. Loosely speaking, hybrid RBGs have two security anchors.

The advantage of using a hybrid DRBG is that it can help prevent either cryptanalysis or add security features such as enhanced forward secrecy, or both.

NOTE 2 Additional information on classes of DRBGs can be found in References [13] and [15].

7.4 The RBG spectrum

The classes of RBGs are depicted in Figure 2.

The distinction between an NRBG and a DRBG can be seen as a sub-division of a spectrum of design choices. At one end of the spectrum is a DRBG that is designed to use a single seed for its entire lifetime. At the other end of the spectrum is a simple NRBG that is designed as the output of a strong, highly random, non-deterministic entropy source. Intermediate RBG designs allow for periodic reseeding of DRBGs. Intermediate RBG designs allow for NRBGs whose output also depends upon a seed value or NRBGs that process the output of the entropy source in a complex manner similar to the way in which a DRBG processes a seed value. The choice of type of RBG is a cost/benefit trade-off depending on the application requirements.

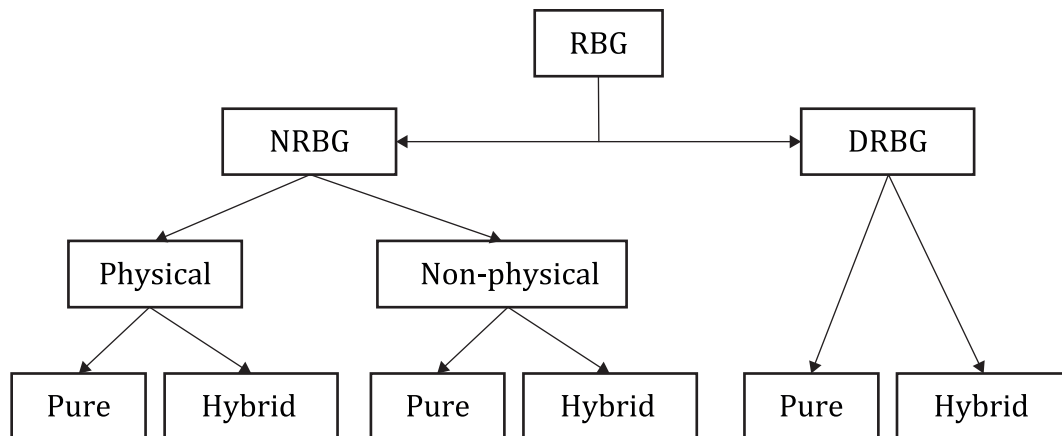


Figure 2 — RBG classes

8 Overview and requirements for an NRBG

8.1 NRBG overview

A consuming application may choose to use a DRBG for random bit generation. In such a case, the primary role of an NRBG is to generate random initial seeds for a DRBG. However, this document does not preclude using an NRBG, potentially with additional strong cryptographic post-processing, to generate all of the randomness required by an application.

Objectives and requirements that are unique to NRBGs, beyond the requirements in [Clauses 5](#) and [6](#), are specified in [8.3](#) to [8.9](#). Examples of NRBGs can be found in [Annex D](#).

8.2 Functional model of an NRBG

This subclause introduces a specialization of the components and the general model for the case of an NRBG. It describes the general operation of an NRBG, including the objectives that each NRBG functional component is intended to accomplish.

[Figure 3](#) provides a functional block diagram depicting a conceptual NRBG. In this diagram, dashed lines indicate a component that is optional. It is important to note that the mandatory components shown are not required to be implemented as actual physical components but shall be implemented functionally.

Each of the components and their objectives and requirements are intended to prevent various security weaknesses associated with random bit generation that have been known to occur in cryptographic applications and environments. In general, each of the following components are required in an NRBG. In some applications, it can be the case that none of the requirements for a given component are applicable. If this can be adequately justified and documented, that component can be omitted from the NRBG, either because the threat being countered by the component is not present in the intended application, or because the objective served by the component is being met implicitly or addressed by other means.

The following paragraph is an overview of the way in which these components interact to produce random output. The non-deterministic entropy source, while in general not producing acceptable random output by itself, behaves probabilistically.

An internal state transition function based on one or more deterministic cryptographic functions combines a specified quantity of the entropy source data with the working state data to produce a new working state. If the bit-length of the entropy source input data used for each internal state transition is n , and the size of the working state is m bits, then the internal state transition function is a function from the space of $(n+m)$ -bit sequences to the space of m -bit sequences. The parameter sizes typically used in an NRBG will make the number of possible inputs to this function vastly larger than the number of possible outputs from the function. This results in a very large number of combinations of input sequence and current working state being assigned to any particular output for the new working state. Furthermore, if the cryptographic functions upon which typical internal state transition functions are based have been thoroughly analysed cryptographically, it is reasonable to assume that such an internal state transition function will map the input space to the output space in a nearly uniform way (i.e. each output has approximately the same number of preimages). These assumptions lead toward satisfaction of the objective of creating uniformly generated binary sequences in the working state. DRBGs and NRBGs shall add new entropy to their state at a rate greater than or equal to the rate that they output entropy. Therefore, whenever the consuming application requires random output, both DRBGs and NRBGs ensure that the internal state contains sufficient entropy that has not yet been used to produce random output, and then use the output generation function to process the current working state to produce random output.

The output generation function takes the internal state, which can be required to be kept confidential from an adversary, and produces an output that cannot be distinguished from random. The output generation function is typically also a cryptographic function or other function having similar uniform distribution characteristics. An argument similar to that for the internal state transition function leads to the conclusion that, with appropriate parameter size choices, the output generation function produces a uniformly distributed binary output.

The behaviour of physical noise sources in physical entropy sources should be described using stochastic models (see Reference [\[16\]](#)). Such a model allows accurate statistical testing designed to detect defects of the noise source (i.e. when the noise source produces output that is not sufficiently random). A detailed understanding of the underlying physics of the system can be used to inform the stochastic model and can enable targeted statistical testing. It is possible to choose very simple internal state transition functions and output generation functions to facilitate the determination of the stochastic behaviour of the output bits. If the NRBG is intended to be used directly for the generation of sensitive key material (in contrast to seeding a DRBG), it is recommended to use cryptographic post-processing of the noise source output.

For non-physical entropy sources, the entropy per output bit can be lower than for physical entropy sources and it can be difficult to accurately determine the moment when the entropy source fails. In such cases, more complex post-processing operations can be required to ensure that the output of the RBG is suitably random and to cope with an undetected failure of the entropy source.

A secure NRBG shall also include mechanisms designed to increase the likelihood of continued secure operation in the event of failures or compromises. Detectable failures are addressed through the inclusion of periodic and continuous health tests on the various components. Undetectable failures or compromises are addressed in two ways: the internal state shall include either a periodically changing secret parameter (e.g. a key) or a safety margin in the maintenance of entropy during NRBG operation. In the first case, the secret parameter shall parameterize the deterministic operation of the internal state transition function so that knowledge of the working state and all the inputs to the NRBG are insufficient to determine the NRBG output. If a safety margin is used, decreases in the available input entropy (due to unexpected events or statistical model inaccuracies) are less likely to result in biased random output. An objective for an NRBG will be for the NRBG to continue to operate in a manner no less secure than a DRBG in the event that the entropy source completely fails.

This objective can be met by the inclusion of a DRBG in an NRBG design, or by:

- 1) verifying that the health tests on the entropy source(s) can quickly detect possible weaknesses or failures in the entropy source that can unacceptably damage the quality of the random bits; and
- 2) initiating appropriate measures when such errors are detected.

NOTE It is usually feasible to meet these conditions for physical entropy sources since they use dedicated hardware. A generic proposal is discussed in Reference [23].

Enhanced forward and backward secrecy are properties of a properly implemented and working NRBG, given that the inputs are unknown. In fact, it is an inherent feature of an NRBG, since the NRBG always draws upon fresh entropy for each call. Thus, enhanced forward and backward secrecy are automatically provided if the entropy source(s) deliver sufficient entropy.

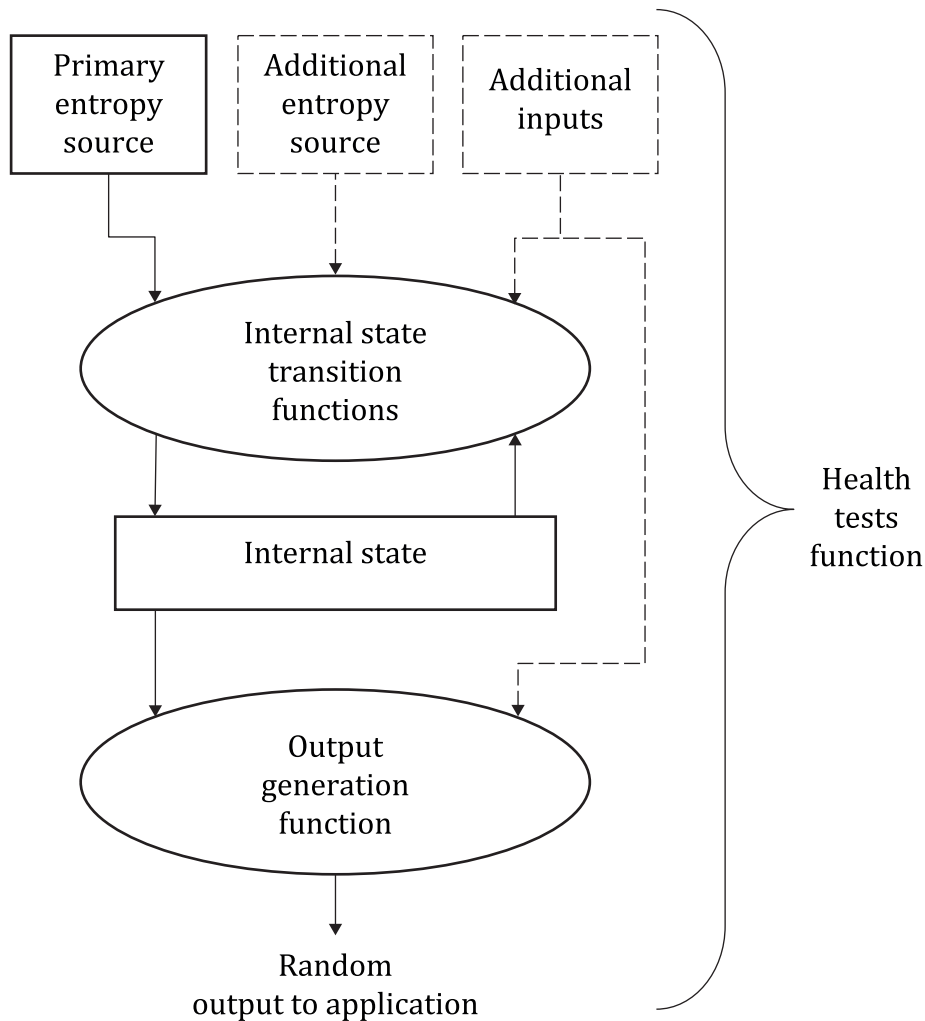


Figure 3 — Block diagram of an NRBG

8.3 NRBG entropy sources

8.3.1 General

The estimation of the quantity of entropy produced by an entropy source is discussed in [Annex F](#).

8.3.2 Primary entropy source for an NRBG

8.3.2.1 NRBG primary entropy source overview

This component serves as the source of unpredictability in the NRBG by providing data to be processed by the internal state transition function. This source of unpredictability differs from that in a DRBG, which relies on the unknown initial seed for unpredictability. In an NRBG, unpredictability is based on the use of one or more noise sources.

A physical noise source is accessed by dedicated hardware that measures the physical characteristics of a sequence of events produced by that noise source. Examples of physical noise source measurements include measuring the time between radioactive emissions of an unstable substance and measuring the output of a noisy diode that receives a constant input voltage level and outputs a continuous, normally distributed analogue voltage level.

A non-physical noise source is any non-deterministic noise source that is not a physical noise source. Examples of non-physical noise sources are system data or the RAM contents of a PC, or an aperiodic signal

based on an irregularly occurring event or process timing interactions, such as the sampling of a high-speed counter whenever a human operator presses a key on a keyboard.

Specific details and requirements for physical entropy sources and non-physical entropy sources (which contain physical and non-physical noise sources) can be found in [8.3.3](#) and [8.3.4](#), respectively.

Depending on the noise source, it is possible that the output of the entropy source is not adequate for direct use as an RBG output, either because it is not in the form of a binary digital sequence or because it exhibits statistical biases. These shortcomings are remedied in an entropy source by digitization and/or conditioning the noise source output to reduce bias or ensure that the entropy rate of the entropy source output is no less than some specified amount.

The output entropy from the NRBG cannot ever be greater than the input entropy provided by the entropy source.

8.3.2.2 Requirements for an NRBG primary entropy source

The primary entropy source is the foundation of the non-deterministic behaviour of the NRBG. By definition, an NRBG design shall include this component.

The functional requirements for the primary entropy source are as follows.

- 1) Although the entropy source is not required to produce unbiased and independent outputs, it shall have the property that it exhibits probabilistic behaviour; i.e. the output shall not be predictable by any known algorithmic rule.
- 2) The entropy source shall contain a means by which the rate of entropy contribution from its constituent noise source can be assessed. This requires that the operation of the noise source be based on well-established principles or extensively characterized behaviour so that an appropriate statistical model for the noise source can be identified.
- 3) The noise source shall be amenable to bench testing by a validation laboratory or some other independent verification process to ensure proper operation. In particular, it shall be possible to collect a data sample from the entropy source during the validation or independent verification process to allow for the evaluation of the claimed statistical model, the entropy rate, and the appropriateness of the health tests in the entropy source.
- 4) Failure or severe degradation of the entropy source shall be detectable. Based on the frequency of health tests being performed, it is possible that this detection is not immediate.
- 5) The entropy source shall be protected from adversarial knowledge or influence. In particular, it shall be infeasible for the adversary to influence the entropy source in such a way that the entropy that the source produced falls below a threshold value without being detected.

8.3.2.3 Recommendations for an NRBG primary entropy source

The recommended features of the primary entropy sources are as follows.

- 1) The entropy source should be stationary in a mathematical sense. If it is not stationary, then there should be bounds on the probability of fluctuation. The probabilistic behaviour of such a source should not change significantly over time. For example, if the entropy source produces outputs from a certain alphabet with a statistical distribution, it should be consistent in this bias over time. An entropy source that is not stationary would greatly complicate the process of estimating the rate of entropy contribution and increase the difficulty of validating the resulting NRBG design.
- 2) Appropriate health tests tailored to the known statistical model of the source should place special emphasis on the detection of misbehavior having increased likelihood near the boundary between the nominal operating environment and abnormal conditions. This requires a thorough understanding of the operation of the entropy source.

8.3.3 Physical entropy sources for an NRBG

8.3.3.1 NRBG physical entropy source overview

The physical noise source within a physical entropy source typically provides a continuous stream of output while it has power applied. However, this output is not necessarily provided as a simple binary string. For example, if the physical noise source output is based on the time between radioactive emissions, it is possible that the noise source does not provide a simple binary output string that contains entropy but a series of timings between emissions. Alternatively, if the physical noise source used is based on voltage variations in a noisy diode, then the output can be a series of voltage measurements.

NOTE In these examples, the entropy is based on the variability of the timings (in the first example) and on assignments made for the threshold values (in the second example).

Typically, this noise source output shall be interpreted before it can be of any use. This can be done by comparing the physical data provided by the noise source to a series of threshold values. The data provided by the noise source can either be compared to a single threshold value, so that a single reading of the source produces a single bit of output, or a series of threshold values, so that a single reading of the source produces several output bits.

Since a physical entropy source is likely to be contained within a cryptographic boundary of an NRBG, the focus of the security requirements in this document is on the proper collection and interpretation of a true physical noise source. Other measures can be considered to reduce the possibility that an external adversary obtains information from, or exerts undue influence over, the noise source.

An advantage of a physical noise source is that it can generally be modeled as a stochastic process. This in turn, leads to efficient online health checks that accurately determine the moment when a physical noise source fails. Such tests can be used to prevent an NRBG based on the use of a physical entropy source from continuing to operate with a failed noise source (e.g. the noise source becomes predictable).

An evaluation methodology for physical noise sources is specified in ISO/IEC 20543.

8.3.3.2 Requirements for the physical noise source of an NRBG

The functional requirements of a physical noise source are as follows.

- 1) The threshold values for the noise source shall be chosen so that the output string contains a sufficient amount of entropy.

NOTE There is a difference between the entropy displayed by a noise source (which can produce data at an arbitrary degree of precision) and the entropy provided by a binary interpretation of that source.

- 2) The total failure of the noise source shall be immediately detectable. A degradation of the noise source shall be detected quickly.

8.3.3.3 Recommendation for the physical entropy source of an NRBG

The entropy source should be formally analysed and the threshold values for its noise source chosen in such a way that the output contains the greatest possible amount of entropy. Making such an analysis is feasible because true physical noise sources are comparatively simple compared to entropy derived from non-physical sources.

8.3.4 NRBG non-physical entropy sources

The entropy from non-physical noise sources within non-physical entropy sources is usually provided by a system upon request from the NRBG. Hence, non-physical noise sources are typically outside the defined protection boundary of the NRBG. The data is usually already binary in nature. For example, the noise source output can be a digital representation of the time between key presses (as measured by a system), certain unpredictable network statistics, or the contents of unpredictable portions of the RAM.

Since a non-physical entropy source is, at least partly, outside the control of the NRBG, sufficient precautions shall be taken to minimize the possibility of an adversary gaining any knowledge of the data and/or the likelihood of influencing the entropy source. Furthermore, as it is a lot harder to model a non-physical noise source accurately as a stochastic process, it can be more difficult to determine whether such a source is operating correctly or not. Hence, NRBGs based on non-physical entropy sources often use complex state transition and output generation functions that guarantee that the RBG is at least as secure as a DRBG for the period of time between the noise source failing and the failure being detected.

There are no extra security requirements for a non-physical entropy source.

An evaluation methodology for non-physical noise sources is specified in ISO/IEC 20543.

8.3.5 NRBG additional entropy sources

8.3.5.1 NRBG additional entropy sources overview

The operation of an NRBG can also include one or more additional entropy sources. An additional entropy source can be useful for a variety of reasons. It can provide a layer of protection against degradation of the NRBG output due to the primary entropy source failing or straying from the characterized statistical model of its noise source.

In situations where the primary entropy source has some degree of external visibility, an additional entropy source that is less externally accessible lessens the usefulness of knowledge of the primary entropy source to the adversary.

Finally, having multiple entropy sources can provide the capability for split control, enabling applications where multiple users require access to the same NRBG output but distrust each other's potential influence over the individual entropy sources. For such applications, it is possible to design the NRBG so that a user's trust in a single entropy source is sufficient for trust in the final NRBG output.

An NRBG may use an additional source of randomness e.g. a seed value produced by a pure DRBG (see [9.3.2](#)). NRBGs that use a seed value from a deterministic process are discussed in [8.3.5.2](#) and [8.3.5.3](#).

Despite the additional sources of randomness providing additional unpredictability to the output of an RBG, the security of the RBG shall rest solely upon the primary entropy source.

8.3.5.2 Requirements for NRBG additional entropy sources

The functional requirements for additional entropy sources are as follows.

An additional entropy source shall be included if the:

- 1) primary entropy source is insufficiently reliable from a failure perspective. In this case, the additional entropy source shall meet or exceed the same requirements as the primary entropy source;
- 2) primary entropy source produces entropy at a rate that is insufficient for the desired rate of random bit generation. In this case, the additional entropy source shall meet or exceed the same requirements as the primary entropy source. Alternatively, instead of including an additional entropy source, it can be acceptable to solve this problem by using the NRBG only for the initial seed generation for a DRBG; and
- 3) application or environment require that the RBG exhibit features that are best provided by a hybrid NRBG, i.e. that the RBG takes a seed value from another randomness source as input. In this case, the additional entropy source shall satisfy the conditions given in [8.3.6](#).

Whether the additional entropy source(s) is/are of the same type as the primary entropy source (i.e. a second version of the same type of entropy source), or a completely different component or process, this source shall operate independently of the primary source to ensure that the combined entropy sources will not lose entropy due to statistical dependence. Independence of entropy sources also facilitate the design and evaluation or independent verification processes by allowing the primary and secondary entropy sources to be analysed separately. It also reduces the likelihood of a failure in the primary entropy source.

8.3.5.3 Recommendations for NRBG additional entropy sources

The optional but recommended features of the additional entropy sources are as follows.

- 1) The RBG output should remain unpredictable even if the attacker can adaptively choose the values of the additional entropy sources; the attacker should be unable to predict the next bit produced by the RBG with probability significantly greater than one half.
- 2) Additional entropy source(s) should be included in the NRBG design if either of the following is true:
 - a) the primary entropy source is somewhat non-stationary (i.e. inconsistent) in its statistical behaviour, making the estimation of input entropy more difficult; or
 - b) there is a concern that the primary entropy source is possibly not free of adversary knowledge or influence. In this case, the additional entropy source should satisfy the same requirements as the primary source, although it can be acceptable for it to be somewhat more deterministic. That is, actions by the user or factors from the system environment can influence (although not completely determine) the output from this source in a perceptible way.

8.3.6 Hybrid NRBGs

An NRBG is a hybrid NRBG if it uses complex, stateful post-processing (e.g. cryptographic processing). The main advantage of a hybrid NRBG is that the post-processing adds computational security to the already unpredictable entropy source output as a second security anchor.

The additional functional requirements of a hybrid NRBG are as follows:

- 1) The post-processing shall provide backward secrecy, forward secrecy and enhanced backward secrecy.
- 2) The hybrid NRBG shall provide enhanced forward secrecy.
- 3) No unauthorized person shall be able to manipulate, influence, or update any seed value used for post-processing.

8.4 NRBG additional inputs

8.4.1 NRBG additional inputs overview

The operation of an NRBG can require taking certain public or user-generated inputs such as commands, power variations and time-variant data such as counters, clocks or user-supplied data. It may be assumed that these additional inputs are either directly observable or under the direct control of an adversary. Therefore, it is vital that the manipulation of these inputs does not reduce the effectiveness of the RBG or that only correctly authenticated and authorized personnel have the ability to manipulate these inputs, and then only within a defined operational policy.

8.4.2 Requirements for NRBG additional inputs

The functional requirements of all additional inputs shall include protection against their manipulation (commands, clock, timers, power etc.) that comprises the security of the RBG.

NOTE This can be accomplished by limiting the influence that these inputs have over the overall control of the NRBG. Power input is, of course, a special case. Disruption of power will obviously result in a complete denial of service. If this is a concern, then the operating environment of the NRBG is expected to provide uninterruptible power, which is a system issue and beyond the scope of this document.