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)

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

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 $S(n+1)$:= 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
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
$X \mid\mid Y$	concatenation of two separate bit-strings X and Y in that order
$\mid a \mid$	the length in bits of string a
x mod n	The unique remainder r, $0 \leq r \leq n$-1, when integer x is divided by n. For example, 23 mod 7 = 2.

5 Properties and requirements of a random bit generator

5.1 Properties of arandom 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