INTERNATIONAL STANDARD

ISO/IEC 18031

Second edition
2011-11-15

Information technology — Security techniques — Random bit generation

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

Reference number
ISO/IEC 18031:2011(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web 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. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 18031 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT Security techniques*.

This second edition cancels and replaces the first edition (ISO/IEC 18031:2005), which has been technically revised. It also incorporates the Technical Corrigendum ISO/IEC 18031:2005/Cor.1:2009.

Introduction

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

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

- random keys and initialisation 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 PIN and password generation;
- nonces.

The purpose of this International Standard 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. This International Standard identifies the two types as non-deterministic and deterministic random bit generators.

A non-deterministic random bit generator can be defined as a random bit generating mechanism that 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, may or may not be publicly available. The seed is processed by a function which provides an output.

NOTE This International Standard also recognizes and discusses the existence of Hybrid Random Bit Generators.

Information technology — Security techniques — Random bit generation

1 Scope

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

This International Standard

- specifies the characteristics of the main elements required for a non-deterministic random bit generator,
- specifies the characteristics of the main elements required for a deterministic random bit generator,
- establishes the security requirements for both the non-deterministic and the deterministic random bit generator.

Where there is a requirement to produce sequences of random numbers from random bit strings, Annex B gives guidelines on how this can be performed.

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 International Standard.

2 Normative references

The following referenced documents are indispensable for the application 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 technology — Security techniques — Message Authentication Codes (MACs) — Part 2: Mechanisms using a dedicated hash-function*

ISO/IEC 10116, *Information technology — Security techniques — Modes of operation for an n-bit block cipher*

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

ISO/IEC 18032, *Information technology — Security techniques — Prime number generation*

ISO/IEC 18033-3, *Information technology — Security techniques — Encryption algorithms — Part 3: Block ciphers*

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

3 Terms and definitions

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

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 values cannot be determined from knowledge of the current value or subsequent values

3.3
biased source
source of bit strings (or numbers) from a sample space where some bit strings (or numbers) are more likely to be chosen than some other bit strings (or numbers)

NOTE 1 Equivalently, if the sample space consists of r elements, some elements will occur with probability different from $1/r$.

NOTE 2 This term can be contrasted with unbiased source (3.35).

3.4
bit stream
continuous output of bits from a device or mechanism

3.5
bit string
finite sequence of ones and zeroes

3.6
black box
idealized mechanism that accepts inputs and produces outputs, but is designed such that an observer cannot see inside the box or determine exactly what is happening inside that box

NOTE This term can be contrasted with glass box (3.14).

3.7
block cipher
symmetric encipherment system with the property that the encryption operates on a block of plaintext, i.e. a string of bits of a defined length, to yield a block of ciphertext

[ISO/IEC 18033-1]

3.8
cryptographic boundary
explicitly defined continuous perimeter that establishes the physical bounds of a cryptographic module and contains all the hardware, software and/or firmware components of a cryptographic module

[ISO/IEC 19790]

3.9
deterministic algorithm
characteristic of an algorithm that states that given the same input, the same output is always produced

3.10
deterministic random bit generator
DRBG
random bit generator that produces a random-appearing sequence of bits by applying a deterministic algorithm to a suitably random initial value called a seed and, possibly, some secondary inputs upon which the security of the random bit generator does not depend

NOTE In particular, non-deterministic sources may also form part of these secondary inputs.

3.11
entropy
measure of the disorder, randomness or variability in a closed system

NOTE The entropy of a random variable X is a mathematical measure of the amount of information provided by an observation of X.

3.12
entropy source
component, device or event which produces outputs which, when captured and processed in some way, produce a bit string containing entropy

3.13
forward secrecy
assurance that the knowledge of subsequent (future) values cannot be determined from current or previous values

3.14
glass box
idealized mechanism that accepts inputs and produces outputs and is designed such that an observer can see inside and determine exactly what is going on

NOTE This term can be contrasted with black box (3.6).

3.15
hash-function
function which maps strings of bits to fixed-length strings of bits, satisfying the following two properties.

- It is computationally infeasible to find for a given output, an input that maps to this output.
- It is computationally infeasible to find for a given input, a second input, which maps to the same output

NOTE Computational feasibility depends on the specific security requirements and environment.

[ISO/IEC 10118-1]

3.16
human entropy source
entropy source that has some kind of random human component

3.17
hybrid DRBG
DRBG that uses a non-deterministic entropy source as an additional entropy source

3.18
hybrid NRBG
NRBG that takes a seed value as an additional entropy source

NOTE Hybrid NRBG may be physical or non-physical.

3.19
initialisation value
value used in defining the starting point of a cryptographic algorithm

NOTE An example of where an initialisation value is used is in a hash-function or an encryption algorithm.

3.20
Kerckhoffs 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 is not known to an adversary

NOTE A Kerckhoffs box lies between a black box and a glass box in terms of the knowledge of an adversary.

3.21
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 Known-answer testing of a deterministic mechanism may also include testing the integrity of the software which 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.

3.22
min-entropy
lower bound of entropy that is useful in determining a worst-case estimate of sampled entropy

NOTE 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.

3.23
non-deterministic random bit generator
NRBG
RBG whose security depends upon sampling an entropy source

NOTE The entropy source will be sampled whenever the RBG produces output, and possibly more often.

3.24
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 for a given output an input which maps to this output

[ISO/IEC 11770-3]

3.25
output generation function
function in an RBG that computes the output of the RBG from the internal state of the RBG

3.26
pseudorandom sequence of bits
sequence of bits or a number that appears to be selected at random even though the selection process is done by a deterministic algorithm

3.27
pure DRBG
DRBG whose entropy sources are seeds

3.28
pure NRBG
NRBG whose entropy sources are non-deterministic

NOTE The pure NRBG may be physical or non-physical.

3.29
random bit generator
RBG
device or algorithm that outputs a sequence of bits that appears to be statistically independent and unbiased

3.30
reseeding
specialised internal state transition function which updates the internal state in the event that a new seed value is supplied

NOTE The usage of the term 'reseeding' is not unique in the literature. Some authors denote as 'reseeding' each mechanism that replaces the current value of the internal state by a fresh value. This International Standard follows this terminology. However, often one distinguishes between 'reseeding' and 'seed update'. The term 'reseeding' then only comprises mechanisms that replace the internal state by a new value, which does not depend on the current value (essentially a new seeding process). In contrast 'seed update' denotes a mechanism that computes the new internal state from its current value and other (usually non-deterministic) data (cf. 9.6, item 3).

3.31
secret parameter
input to the RBG during initialisation, which provides additional entropy in the case of an entropy source failure or compromise

3.32
seed
string of bits that is used as input to a DRBG

NOTE The seed will determine a portion of the state of the DRBG.

3.33
seedlife
period of time between initialising the DRBG with one seed and reseeding (fully initialising) that DRBG with another seed

3.34
state
condition of a random bit generator or any part thereof with respect to time and circumstance

3.35
unbiased source
source of bit strings (or numbers) from a sample space where all potential bit strings (or numbers) have the same possibility of being chosen

NOTE 1 Equivalently, if the sample space consists of r elements, all elements will occur with probability $1/r$.

NOTE 2 This term can be contrasted with biased source (3.3).

4 Symbols

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

Symbol	Meaning
Pr[x]	Probability of occurrence of x.
IV	Initialisation 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 bit wise addition mod 2) of two bit strings X and Y of the same length.
$X \parallel Y$	Concatenation of two 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, $0 \le r \le n\text{-}1$, when integer x is divided by n. For example, 23 mod 7 = 2.
	Used in a figure to illustrate a “switch” between sources of input.

5 Properties and requirements of an RBG

5.1 Properties of an RBG

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

1. 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 International Standard;

2. Unbiased: That is, each potential outcome has the same chance of occurring; and

3. 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 an RBG. The RBGs specified in this International Standard will try to simulate a series of idealized coin flips.

As indicated above, unpredictability is a required property of an RBG. It should not be possible to predict the output of a properly implemented and working RBG. Forward secrecy refers to the inability to predict future output of the RBG based on the knowledge of previous output values and/or internal states. The inability to determine prior output of an RBG, given knowledge of the current or any future output of the RBG, is known as backward secrecy.

The decision whether to incorporate backward and/or forward secrecy is determined by the requirements of the consuming application.

The following factors should be considered when deciding to incorporate backward and/or forward secrecy.

1. In some instances, achieving backward secrecy is more important than achieving forward secrecy. For example, if a cryptosystem is stolen, an adversary may attempt to read the old messages processed by that system. Forward secrecy is not really a concern, since the system is no longer in use by the original owner. Achieving backward secrecy is straightforward (for example by the appropriate use of a one-way function in the design), although there may be a performance cost associated with providing this property, depending on the design.

2. Trying to achieve forward secrecy may not be appropriate for some cryptosystems. For example, a smart card may be initialised at the point of manufacture with sufficient entropy in the seed and is set to expire after a limited time (e.g., two or three years). In this case, it may be much easier to replace the card with a new smart card that is seeded with a different seed than it is to build forward secrecy into the RBG design.

3. In some instances, achieving forward secrecy may be more important than achieving backward secrecy. Consider, for example, the secure generation of nonces. It is not necessary for a random bit generation algorithm to have backward secrecy as all of the previous outputs will be known. However, forward secrecy may be useful to prevent an adversary with knowledge of the generator from being able to predict later outputs.

5.2 Requirements of an RBG

The following requirements apply to all RBGs, both deterministic and non-deterministic.

The requirements provided below are fundamental to the security of cryptographic mechanisms that require random input.

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

1. Under reasonable assumptions, it shall not be feasible to distinguish the output of the RBG from true random bits that are uniformly distributed. Informally, all possible outputs occur with equal probability and a series of outputs appears to conform to the uniform distribution.

2. Given a sequence of output bits, it shall not be known to be feasible to compute or predict any other output bit, either past or future.

3. Throughout the lifetime of the RBG, it shall not be possible to predict output stream sequence repeats.

4. The RBG shall not leak relevant secret information (e.g., internal state of a DRBG) through the output of the RBG.

5. The RBG shall not leak secret information from the perspective of an adversary.

NOTE 1 An example of when this could happen would be a timing attack.

6. 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 International Standard and the requirements of the consuming application.

7. On detection of an error, the RBG shall either (a) enter a permanent error state, or (b) 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 innately in the design.

NOTE 2 See 8.8 and 9.8 for NRBG and DRBG information on RBG errors and health tests.