BS ISO/IEC 18031:2011+A1:2017

Incorporating corrigendum October 2014

BSI Standards Publication

Information technology — Security techniques — Random bit generation

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 18031:2011+A1:2017

National foreword

This British Standard is the UK implementation of ISO/IEC 18031:2011+A1:2017, incorporating corrigendum October 2014. It supersedes BS ISO/IEC 18031:2011, which is withdrawn.

The start and finish of text introduced or altered by amendment is indicated in the text by tags. Tags indicating changes to ISO/IEC text carry the number of the IEC amendment. For example, text altered by ISO/IEC amendment 1 is indicated by [A1) (A1].

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

A list of organizations represented on this subcommittee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2017.
Published by BSI Standards Limited 2017

ISBN 978 0 580 82705 1

ICS 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 31 January 2017.

Amendments/corrigenda issued since publication

Date	Text affected
30 November 2014	Implementation of ISO/IEC corrigendum October 2014: Subclauses B.2, C.4.2, D.1 and last sentence in subclause C.4.1 deleted
30 April 2017	Implementation of ISO/IEC amendment 1:2017

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 18031:2011+A1:2017

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)

Documents
Document Sharing Hub

© ISO/IEC 2011

Full document details: BS ISO/IEC 18031:2011+A1:2017

COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2011

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

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

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 18031:2011+A1:2017

Contents

Page

Foreword vi
Introduction vii
1 Scope 1
2 Normative references 1
3 Terms and definitions 2
4 Symbols 5
5 Properties and requirements of an RBG 6
5.1 Properties of an RBG 6
5.2 Requirements of an RBG 7
5.3 Optional requirements for an RBG 8
6 RBG model 8
6.1 Conceptual functional model for random bit generation 8
6.2 RBG basic components 9
6.2.1 Introduction to the RBG basic components 9
6.2.2 Entropy source 10
6.2.3 Additional inputs 10
6.2.4 Internal state 11
6.2.5 Internal state transition functions 12
6.2.6 Output generation function 13
6.2.7 Support functions 13
7 Types of RBGs 14
7.1 Introduction to the types of RBGs 14
7.2 Non-deterministic random bit generators 14
7.3 Deterministic random bit generators 15
7.4 The RBG spectrum 15
8 Overview and requirements for an NRBG 16
8.1 NRBG overview 16
8.2 Functional model of an NRBG 16
8.3 NRBG entropy sources 18
8.3.1 Primary entropy source for an NRBG 18
8.3.2 Physical entropy sources for an NRBG 20
8.3.3 NRBG non-physical entropy sources 20
8.3.4 NRBG additional entropy sources 21
8.3.5 Hybrid NRBGs 22
8.4 NRBG additional inputs 22
8.4.1 NRBG additional inputs overview 22
8.4.2 Requirements for NRBG additional inputs 22
8.5 NRBG internal state 23
8.5.1 NRBG internal state overview 23
8.5.2 Requirements for the NRBG internal state 23
8.5.3 Optional requirements for the NRBG internal state 24
8.6 NRBG internal state transition functions 24
8.6.1 NRBG internal state transition functions overview 24
8.6.2 Requirements for the NRBG internal state transition functions 25
8.6.3 Optional requirements for the NRBG internal state transition functions 25
8.7 NRBG output generation function 26
8.7.1 NRBG output generation function overview 26
8.7.2 Requirements for the NRBG output generation function 26

Documents
Document Sharing Hub

© ISO/IEC 2011 – All rights reserved

Full document details: BS ISO/IEC 18031:2011+A1:2017

8.7.3 An optional requirement for the NRBG output generation function 26
8.8 NRBG health tests 26
8.8.1 NRBG health tests overview 26
8.8.2 General NRBG health test requirements 27
8.8.3 NRBG health test on deterministic components 27
8.8.4 NRBG health tests on entropy sources 28
8.8.5 NRBG health tests on random output 29
8.9 NRBG component interaction 31
8.9.1 NRBG component interaction overview 31
8.9.2 Requirements for NRBG component interaction 31
8.9.3 Optional requirements for NRBG component interaction 31

9 Overview and requirements for a DRBG 31
9.1 DRBG overview 31
9.2 Functional model of a DRBG 32
9.3 DRBG entropy source 34
9.3.1 Primary entropy source for a DRBG 34
9.3.2 Generating seed values for a DRBG 36
9.3.3 Additional entropy sources for a DRBG 36
9.3.4 Hybrid DRBG 37
9.4 Additional inputs for a DRBG 37
9.5 Internal state for a DRBG 37
9.6 Internal state transition function for a DRBG 38
9.7 Output generation function for a DRBG 39
9.8 Support functions for a DRBG 39
9.8.1 DRBG support functions overview 39
9.8.2 DRBG health test 39
9.8.3 DRBG deterministic algorithm test 40
9.8.4 DRBG software/firmware integrity test 40
9.8.5 DRBG critical functions test 40
9.8.6 DRBG software/firmware load test 40
9.8.7 DRBG manual key entry test 40
9.8.8 DRBG continuous random bit generator test 40
9.9 Additional requirements for DRBG keys 41

Annex A (normative) **Combining RBGs** 43

Annex B (normative) **Conversion methods** 44
B.1 Random number generation 44
B.1.1 Techniques for generating random numbers 44
B.1.2 The simple discard method 44
B.1.3 The complex discard method 44
B.1.4 The simple modular method 45
B.1.5 The complex modular method 45

Annex C (normative) **DRBGs** 46
C.1 DRBG mechanism examples 46
C.2 DRBGs based on hash-functions 46
C.2.1 Introduction to DRBGs based on hash-functions 46
C.2.2 Hash_DRBG 46
C.2.3 HMAC_DRBG 54
C.3 DRBGs based on block ciphers 60
C.3.1 Introduction to DRBGs based on block ciphers 60
C.3.2 CTR_DRBG 60
C.3.3 OFB_DRBG 69
C.4 DRBGs based on number theoretic problems 71
C.4.1 Introduction to DRBGs based on number theoretic problems 71
C.4.3 Micali Schnorr DRBG (MS_DRBG) 71
C.5 DRBG based on multivariate quadratic equations 82
C.5.1 Introduction to a DRBG based on multivariate quadratic equations 82
C.5.2 Multivariate Quadratic DRBG (MQ_DRBG) 82

Documents
Document Sharing Hub

© ISO/IEC 2011 – All rights reserved

Full document details: BS ISO/IEC 18031:2011+A1:2017

Annex D (normative) **Application specific constants** 94
D.2 Default moduli for the MS_DRBG (…) 94
D.2.1 Introduction to MS_DRBG default moduli 94
D.2.2 Default modulus *n* of size 1024 bits 94
D.2.3 Default modulus *n* of size 2048 bits 94
D.2.4 Default modulus *n* of size 3072 bits 94
D.2.5 Default modulus *n* of size 7680 bits 95
D.2.6 Default modulus *n* of size 15360 bits 95

Annex E (informative) **NRBG examples** 97
E.1 Canonical coin tossing example 97
E.1.1 Overview 97
E.1.2 Description of basic process 97
E.1.3 Relation to standard NRBG components 97
E.1.4 Optional variations 98
E.1.5 Peres unbiasing procedure 98
E.2 Hypothetical noisy diode example 99
E.2.1 Overview 99
E.2.2 General structure 99
E.2.3 Details of operation 100
E.2.4 Failsafe design consequences 104
E.2.5 Modified example 104
E.3 Mouse movement example 104

Annex F (informative) **Security considerations** 106
F.1 Attack model 106
F.2 The security of hash-functions 106
F.3 Algorithm and key size selection 106
F.3.1 Introduction 106
F.3.2 Equivalent algorithm strengths 107
F.3.3 Selection of appropriate DRBGs 108
F.4 The security of block cipher DRBGs 109
F.5 Conditioned entropy sources and the derivation function 109

Annex G (informative) **Discussion on the estimation of entropy** 110

Annex H (informative) **RBG assurance** 111

Annex I (informative) **RBG boundaries** 112

Annex J (informative) **Rationale for the design of statistical tests** 114
J.1 Introduction 114
J.2 Runs test 114
J.3 Long runs test 114

Annex K (informative) **Example cases for MQ_DRBG** 116
K.1 General 116
K.2 Example case for *requested_strength* = 80 and *block_length* = 112 117
K.3 Example case for *requested_strength* = 112 and *block_length* = 112 119
K.4 Example case for *requested_strength* = 80 and *block_length* = 128 121
K.5 Example case for *requested_strength* = 112 and *block_length* = 112 122
K.6 Example case for *requested_strength* = 128 and *block_length* = 128 124
K.7 Example case for *requested_strength* = 80 and *block_length* = 192 124
K.8 Example case for *requested_strength* = 112 and *block_length* = 192 126
K.9 Example case for *requested_strength* = 128 and *block_length* = 192 128
K.10 Example case for *requested_strength* = 192 and *block_length* = 192 129
K.11 Example case for *requested_strength* = 80 and *block_length* = 256 131
K.12 Example case for *requested_strength* = 112 and *block_length* = 256 133
K.13 Example case for *requested_strength* = 128 and *block_length* = 256 134
K.14 Example case for *requested_strength* = 192 and *block_length* = 256 134
K.15 Example case for *requested_strength* = 256 and *block_length* = 256 136

Bibliography 139

Documents
Document Sharing Hub

© ISO/IEC 2011 – All rights reserved

Full document details: BS ISO/IEC 18031:2011+A1:2017

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.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

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.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

This page deliberately left blank

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 18031:2011+A1:2017

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*

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub

Full document details: BS ISO/IEC 18031:2011+A1:2017

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

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

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.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

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

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

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.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

$\lceil X \rceil$	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.
$X \| Y$	Concatenation of two bit strings X and Y in that order.
$\| a \|$	The length in bits of string a.
x mod n	The unique remainder r, $0 \le r \le n$-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.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

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.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

8. 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 I).

9. The probability that the RBG can “misbehave” in some pathological way that violates the output requirements (e.g., constant output or small cycles, i.e., looping such that the same output is repeated) shall be sufficiently small. That means that the probability of error should be consistent with the overall confidence in correct operation that is required of the RBG, which need not be the same as the required strength of cryptographic security.

10. 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).

11. The implementation shall be designed to allow validation, including specific design assertions about what the RBG is intended not to do. The validation of an NRBG means that the NRBG behaves as expected, not just during normal operation, but also at the boundaries of the intended operational conditions. Security-relevant branches in the code that govern behaviour in exceptional conditions (e.g. initialisation, failed health tests, etc.) shall be validated by deliberately forcing all error conditions to occur during validation testing.

12. There shall be design evidence (theoretical, empirical, or both) to support all security requirements for the RBG, including protection from misbehaviour.

5.3 Optional requirements for an RBG

Optional requirements for an RBG are as follows.

1. If the RBG is capable of operating in more than one mode, the RBG should return information about the mode in which it is operating, upon request.

2. A consuming application may require that the RBG be run in a test mode, for example, when using a seed that is non-secret and therefore is capable of reconstructing the bits it generates. When an RBG is in the test mode, the RBG shall not be capable of being used to generate secret bits. When using a secret seed, the RBG shall not operate in the test mode.

3. Considered as a glass box, an RBG should have backward secrecy. If this attribute is supported it 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 predict any previous output bit.

4. Considered as a glass box, an RBG may have forward secrecy. If supported, it means that given all accessible information about the RBG (comprising some subset of inputs, algorithms, and outputs), it shall be infeasible (up to the specified security strength) to compute or predict any future output bit at the time that 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 should achieve without constraining or mandating how it is 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 in this International Standard.

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.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

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

NOTE There exist RBGs, which do not incorporate all functional components as depicted in Figure 1. Physical NRBGs, for instance, 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 may be the case where all functional requirements are met without incorporating all basic components. They are:

a) the entropy source;

b) additional inputs;

c) the internal state;

d) the internal state transition function;

e) the output generation function; and

f) support functions.

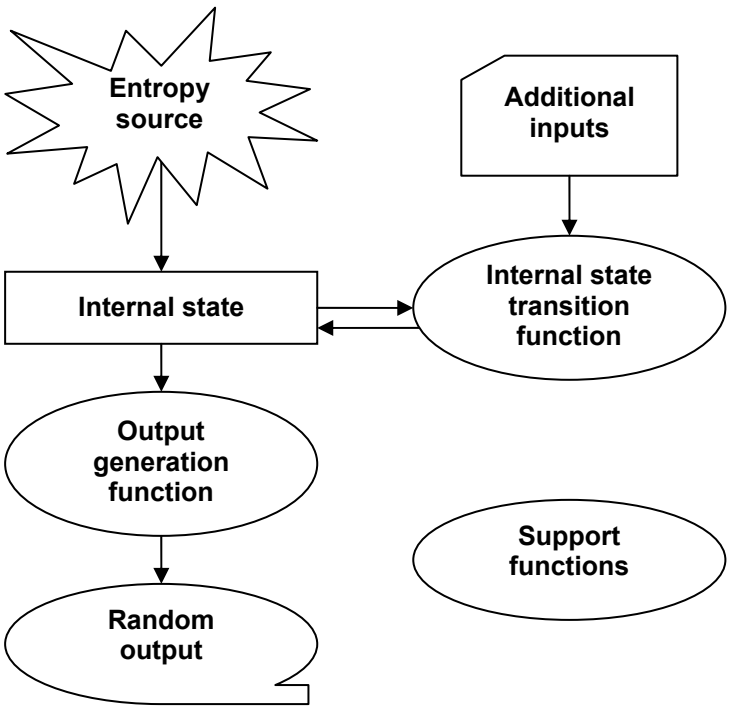

Figure 1 — RBG functional model

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

6.2.2 Entropy source

6.2.2.1 Entropy source overview

The entropy source is the source of unpredictable bits. These bits may be biased and may be dependent on one another to some extent; in fact this will often be the case. In the case of a DRBG, the entropy source may be a distinct, and possibly a remote NRBG. However, for an NRBG, the entropy source component is the combination of the existence of some entropy producing activity, the detection of this activity and the digitisation mechanism.

The entropy source shall produce bits with non-zero entropy. It is the only model component that produces entropy. 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 entropy 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, other than that the entropy source has non-zero entropy. However, any particular RBG design will necessarily establish specific conditions that shall hold for the design to work.

The model allows the entropy source to be decomposed in different ways. One way is recursive decomposition. In particular, it is possible that the entropy source for one RBG is another, distinct (i.e., external) RBG. Indeed it is possible that the external RBG uses a further RBG as an entropy source. There is no innate limit on such a recursive construction, however it should be clear that such a recursion shall ultimately end with a true (i.e., non-deterministic) entropy source without which the whole construction is invalidated. Furthermore, the construction should make sure that there is sufficient true entropy within the system to support the input criteria for each of the individual RBGs.

6.2.2.2 Requirements for an entropy source

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

1. The entropy source shall be based upon well-established principles, or extensively characterised 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 will reliably obtain or exceed 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 unauthorised external body shall be minimized according to the requirements of the application.

4. Loss or severe degradation of an entropy source shall be detectable.

6.2.3 Additional inputs

6.2.3.1 Additional inputs overview

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

6.2.3.2 A requirement for additional inputs

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

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

6.2.4 Internal state

6.2.4.1 Internal state overview

The internal state contains the memory of the RBG. It may include a seed, a counter, a clock value, user input, 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. Since the internal state transition functions and the output generating function are deterministic functions, and since deterministic functions have the property that they always give the same output for the same input, it may 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.

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 portion of the internal state that supports this functionality is known as the working state. Care should be taken to ensure that the value of the working state cannot be influenced by any outside event (for example, by restarting the RBG).

The second role of the internal state is to parameterise the deterministic internal state transition functions and/or output generation function in such a way that when this parameter is unknown, the deterministic function exhibits the properties required of it. For example, it may parameterise 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 (sometimes known as the random bit generator's key) 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 not required that all RBGs have a secret parameter.

NOTE 2 Key management concerns are discussed in ISO/IEC 11770-1[6].

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 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-initialised).

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 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 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 is available to an adversary, size $2x$ should be used (see e.g. [10]).

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

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. See ISO/IEC 11770-1 for further details on 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 An optional requirement for the internal state

The internal states that are used to produce public data, (e.g., nonces and initialisation 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 entropy source nor any other RBG data inputs are considered to act directly on the internal state. Instead, the entropy source and other RBG data inputs are arguments to functions that act on the internal state.

Since the internal state may 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 may 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 the entropy 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 (for example, 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 entropy source input.

In an NRBG, the internal state transition functions determine how the entropy collection affects the internal state of the RBG and may work in a variety of ways: these functions may be simple, stateless algorithms, or they may 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 function shall be verifiable via a known-answer test.

2. The internal state transition function shall, over time, depend on all the entropy carried by the internal state.

3. The internal state transition function 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 function to return to a prior state in normal operation (this excludes testing and authorised verification of the RBG output).

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

6.2.5.3 An optional requirement for the internal state transition functions

The internal state transition function may enable the RBG to recover from the compromise of the internal state (i.e., provide forward secrecy) through 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 may 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 may be as simple as the identity function or as complex as a cryptographic function. The OGF may format or block the output to conform to an external interface convention. Often, the OGF is a “one-way” function, so that it will be hard to invert the OGF in order to recover part of the internal state.

The OGF is deterministic, and will always produce 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 is 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 required 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.

6.2.6.3 An optional requirement for the output generation function

The output generation function should protect the internal state, so that analysis of RBG outputs does not reveal useful information about the internal state.

6.2.7 Support functions

6.2.7.1 Support function overview

The support functions are concerned with assessing the health of the RBG: they do not change the internal state. Support functions include functions that assess the entropy of the input, functions that assess the statistical quality of the output, and functions that 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 by known-answer tests. The “health” of the entropy source and the quality of the output needs to be checked using statistical techniques or tests.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

Detailed information on assessing the health of NRBGs and DRBGs can be found in clauses 8 and 9 respectively.

6.2.7.2 Requirements for the support function

The requirements for support functions of an RBG are as follows.

1. An RBG shall be designed to permit testing that will ensure that 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 operations while in the error state.

7 Types of RBGs

7.1 Introduction to the types of RBGs

Every RBG shall have a primary entropy source. RBGs are classified into two basic types depending on the nature of their primary entropy source. The primary entropy source will either be a non-deterministic entropy source or a deterministic entropy source.

If an entropy source is 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 a non-deterministic entropy source. In particular, no deterministic algorithm should be able to predict the output of a non-deterministic entropy source.

If an entropy source is a seed value, it is called a deterministic entropy source. This seed value (see 9.3) may be supplied to the RBG from an external system or internally from the RBG. If the seed value is supplied externally to the RBG, it is assumed by the RBG to have sufficient entropy and to have been drawn from a specified set. On the other hand, if the seed value is supplied internally from within the RBG, it is assessed to have sufficient entropy and to have been drawn from a specified set.

Care shall be taken to ensure that an adversary cannot gain sufficient control over the seed value to degrade the entropy of the output of the RBG.

An RBG is said to be an NRBG if its primary entropy source is non-deterministic. An RBG is said to be a DRBG if its primary entropy source is deterministic. An RBG may also utilise additional entropy sources that can be either deterministic or non-deterministic.

7.2 Non-deterministic random bit generators

NRBGs can be further specified into sub-classes depending on the nature of their non-deterministic entropy source. A non-deterministic entropy source will be either physical or non-physical.

A physical non-deterministic 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 power is applied to the measuring device. Examples of physical non-deterministic entropy sources 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, non-deterministic entropy source is any non-deterministic entropy source that is not a physical non-deterministic entropy source. Examples of non-physical, non-deterministic entropy sources include measuring the time between key presses or sampling of data in regularly used portions of 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 non-deterministic entropy source. A non-physical NRBG is an RBG with a non-physical non-deterministic entropy source.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

A (physical or non-physical) NRBG is said to be “pure” if all of its entropy sources are non-deterministic.

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) which are typical for DRBGs.

The advantage of using a seed value as an additional entropy source to the NRBG is that it allows the output of the NRBG to be parameterised by the user. Hence, the output of the NRBG can be individualised and may even allow different users to use the same non-deterministic entropy source without compromising the security of the system. Even if there are not several users who use the same entropy source, a secret seed value may be viewed as an additional security feature.

NOTE 1 Additional information on classes of RBGs is found at [5]

NOTE 2 A secret seed value, for instance, could be an additional DRBG-typical security feature. If the NRBG meets the requirements of the standard even with a simple state transition function and a simple output function (cf. 8.2 and E.1) a cryptographic state transition function or a cryptographic output function could be an additional DRBG-typical security feature.

7.3 Deterministic random bit generators

DRBGs can also be classified as either pure or hybrid. A DRBG is said to be “pure” if all of its entropy sources are seeds and “hybrid” if it uses a non-deterministic entropy source as an additional entropy source.

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).

NOTE 1 The security of a hybrid NRBG is based on its non-deterministic part while the security of hybrid DRBGs (see 9.3.4) is essentially based on their deterministic components. If an RBG may be viewed as both a hybrid NRBG or alternatively as a hybrid DRBG we denote it is as “hybrid RBG”. Loosely speaking, hybrid RBGs have two security anchors.

The advantage of using a non-deterministic entropy source as an additional entropy source is that it allows the output to be non-deterministic, and may help prevent cryptanalysis and/or add security features such as forward or backward secrecy.

NOTE 2 Additional information on classes of DRBGs is found at [4]

7.4 The RBG spectrum

The NRBG and DRBG classes are depicted in Figure 2.

The distinction between an NRBG and a DRBG can be seen as a classification 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 or the inclusion of non-deterministic entropy sources in 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 correct choice of RBG is a cost/benefit trade-off depending on the application requirements.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

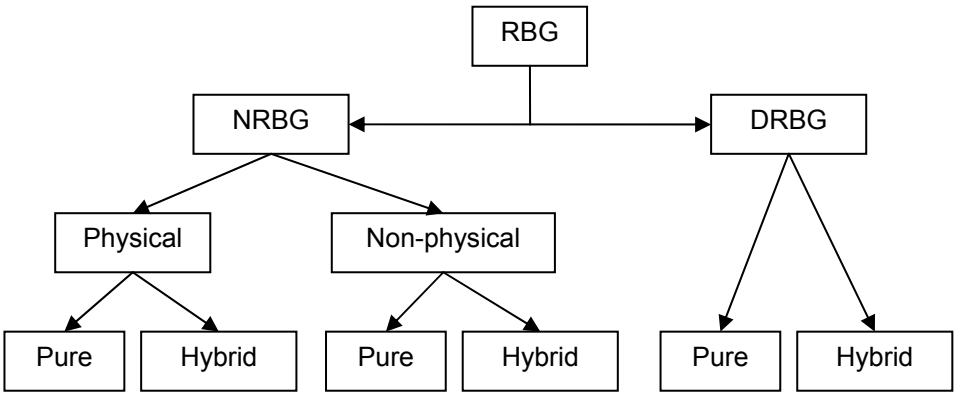

Figure 2 — RBG classes

8 Overview and requirements for an NRBG

8.1 NRBG overview

It is likely that in the majority of applications, the choice will be to use a DRBG for random bit generation. Thus, it is envisioned that the primary use of an NRBG satisfying this International Standard will be to generate random initial seeds for a DRBG. However, this International Standard does not preclude the use of an NRBG to generate all of the random bits needed by the application.

For the purposes of this International Standard, an NRBG shall meet the requirements specified in clauses 5 and 6. However, NRBG unique objectives and requirements, beyond those specified in clauses 5 and 6, which will be imposed on NRBG designs, are detailed in sections 8.3 through 8.9. NRBG examples can be found at Annex E.

8.2 Functional model of an NRBG

This clause will introduce a specialisation of the components and the general model, for the case of an NRBG. It will describe the general operation of an NRBG, including the objectives each NRBG functional component is intended to accomplish.

Figure 3 illustrates a functional block diagram depicting a conceptual NRBG that satisfies this International Standard. In this diagram, dashed lines indicate a component that is optional, depending on various factors. It is important to note that the components shown do not necessarily have to be implemented as actual physical components, but do need to be implemented functionally in some sense.

Each of the components, and the objectives and requirements on these components, 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 will be required in an NRBG. In some applications, there may be a legitimate argument that none of the requirements for a given component are applicable. If this can be adequately justified and documented, that component may 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 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 this entropy source data with the working state data to produce a new working state. The internal state transition function accomplishes this distillation by associating with each of the possible

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

combinations of input sequence and current working state a value of the next working state, as determined by the current value of the secret parameter. If the length of the entropy source input data used for each internal state transition is n bits 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, since 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. The distinguishing characteristic between a DRBG and an NRBG is that an NRBG shall add new entropy to its state at a rate greater than or equal to the rate that it outputs entropy. Therefore, whenever the consuming application requires random output, the NRBG ensures that the internal state contains sufficient entropy that has not yet been used to produce random output, and then uses the output generation function to process the current working state to produce random output.

The output generation function takes the internal state, which may need to be kept confidential from an adversary, and produces an output that cannot be distinguished from random. The output generation function will typically also be 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 will produce uniformly distributed binary output.

For many physical entropy sources the entropy per bit is sufficiently large so that these bits are either directly applicable or applicable after a simple post processing operation (e.g., XORing the entropy source output with the current internal state). Moreover, the behaviour of physical entropy sources can often be described using a stochastical model (see e.g. [7]). Such a model allows accurate statistical testing designed to detect the failure of the source (i.e., when the source produces output which is not sufficiently random for the application). It may therefore be recommendable to choose very simple state transition functions and output generation functions, which facilitates to determine the stochastical behaviour of the output bits. Secret parameters may not be necessary in such a case.

For non-physical entropy sources, the entropy per bit may be lower than for physical entropy sources and it may be difficult to accurately determine the moment when the entropy source fails. In such cases, more complex post processing operations may 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 will 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 health tests on the various components. Undetectable failures or compromises are addressed in two ways. First, the internal NRBG state includes a periodically changing secret parameter that parameterises the deterministic operation of the internal state transition function. Because of this component, knowledge of the remaining part of the internal state (known as the working state) of the NRBG and all the inputs to the NRBG is insufficient to determine the NRBG output. The second approach is the inclusion of a safety margin in the maintenance of entropy during NRBG operation, so that decreases in 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 meeting this International Standard 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.

The requirement that the NRBG should continue to operate in a manner no less secure than a DRBG in the event that the entropy source completely fails may be dropped if:

1. it can be verified that the health tests on the entropy source(s) will sufficiently soon afterwards detect possible weaknesses and failures of the entropy source(s) which would deteriorate the quality of the random bits in an unacceptable manner; and

2. appropriate measures are initiated in such a case (e.g. stopping the generation of random bits).

Documents
Document Sharing Hub

© ISO/IEC 2011 – All rights reserved

Full document details: BS ISO/IEC 18031:2011+A1:2017

NOTE It is usually feasible to meet these requirements for physical entropy sources as they use dedicated hardware. A generic proposal is discussed in [13].

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

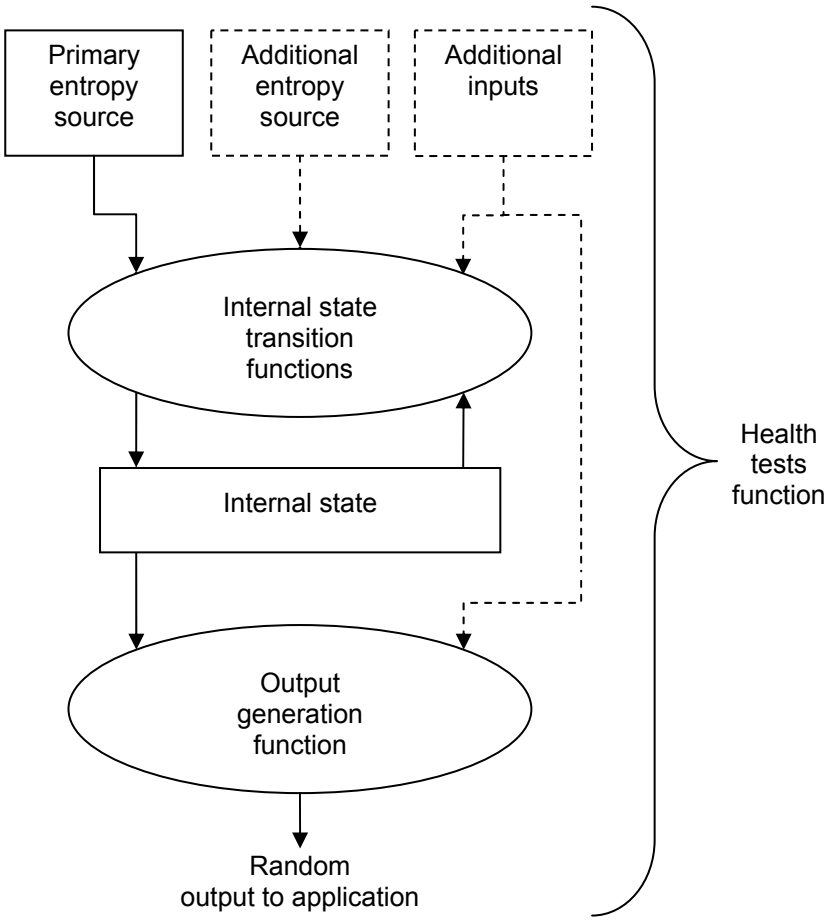

Figure 3 — Block diagram of an NRBG

8.3 NRBG entropy sources

8.3.1 Primary entropy source for an NRBG

8.3.1.1 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 solely on the unknown initial seed for unpredictability. In an NRBG, unpredictability is based on the use of one or more sources of entropy. These sources are known as non-deterministic entropy sources and can be further classified as either physical or non-physical.

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. Examples of physical entropy sources include measuring the time between radioactive emissions of an unstable substance, and the output of a noisy diode, which receives a constant input voltage level and outputs a continuous, normally distributed analogue voltage level.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

A non-physical entropy source is any non-deterministic entropy source that is not a physical entropy source. Examples of non-physical entropy sources are system data or RAM content 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 can be found in 8.3.2 and 8.3.3 respectively.

Depending on the entropy source, the output of the entropy source may not be adequate for direct use as random 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 NRBG by processing the entropy source output with an internal state transition function that produces binary output data.

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

8.3.1.2 Requirements for an NRBG primary entropy source

The primary entropy source is the foundation of the non-deterministic behaviour of the NRBG. By the very definition of an NRBG, the 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., it shall not be definable by any known algorithmic rule.

2. The NRBG shall contain a means by which the rate of entropy contribution from the entropy source may be assessed. This requires that the operation of the entropy source be based on well-established principles or extensively characterised behaviour so that an appropriate statistical model for the entropy source can be identified.

3. The entropy 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 on the entropy source.

4. Failure or severe degradation of the entropy source shall be detectable. Based on the frequency of health tests being performed, this detection may not be 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 produces falls below a threshold value.

8.3.1.3 Optional requirements for an NRBG primary entropy source

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

1. The entropy source shall be stationary in a mathematical sense. If it is not stationary, then there shall 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 will greatly complicate the process of estimating the rate of entropy contribution and increase the difficulty of validating the resulting NRBG design (unless the design includes additional entropy sources that do satisfy this requirement).

2. Appropriate health tests tailored to the known statistical model of the source should place special emphasis on detection of behaviour 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.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

8.3.2 Physical entropy sources for an NRBG

8.3.2.1 NRBG physical entropy source overview

A physical entropy source will typically provide a continuous stream of output whilst it has power applied. However, this output is not necessarily provided as a simple binary string. For example, if the physical entropy source is based on the time between radioactive emissions the entropy source may not provide a simple binary output string that contains entropy, but as a series of timings between emissions. Alternatively, if the physical entropy source used is based on voltage variations in a noisy diode then the output could be a series of voltage measurements.

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

Since a physical entropy source is likely to be contained within a cryptographic boundary of an NRBG, the emphasis of the security conditions are on the proper collection and interpretation of a true physical entropy source and not on the possibility that an adversary may be able to gain information and/or initiate undue influence of the source.

An advantage of a physical entropy source is that it is generally feasible to model as a stochastic process that can be handled. This in turn, leads to efficient online health checks that accurately determine the moment when a physical entropy source fails. Such tests can be used to prevent a physical NRBG from continuing to operate with a predictable entropy source, and may remove some of the need for complex state transition or output generation functions.

8.3.2.2 Requirements for an NRBG physical entropy source

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

1. The threshold values shall be chosen so that the output string contains a sufficient amount of entropy for the application. Note that there is a difference between the entropy displayed by a source (which may produce data at infinite precision) and the entropy provided by a binary interpretation of that source.

2. The total failure of the entropy source shall be immediately detectable. A degradation of the entropy source shall be detected sufficiently fast, where “sufficiently fast” depends on the degree of degradation.

8.3.2.3 An optional requirement for an NRBG physical entropy source

The entropy source should be formally analysed and the threshold values chosen in such a way that the output contains the greatest possible amount of entropy. This is possible because true physical entropy sources are comparatively simple compared to human entropy sources.

8.3.3 NRBG non-physical entropy sources

Non-physical entropy sources are usually provided by a system upon request of the NRBG. Hence, non-physical entropy sources are typically outside the defined protection boundary of the NRBG (see 5.4). The data is usually already binary in nature. For example, the source could 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 entropy source accurately as a stochastic process, it may be more difficult to determine whether such a source is operating correctly or not. Hence, non-physical NRBGs often use complex state transition and output

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

generation functions that guarantee that the RBG will be at least as secure as a DRBG for the period of time between the entropy source failing and the failure being detected.

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

8.3.4 NRBG additional entropy sources

8.3.4.1 NRBG additional entropy sources overview

The operation of an NRBG may also include one or more additional entropy sources. An additional entropy source may 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 characterised statistical model.

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

Finally, having multiple entropy sources may 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 have an additional entropy source that is not a non-deterministic entropy source but rather a deterministic entropy source, i.e., a seed value. An NRBG that also has a deterministic entropy source is known as a hybrid NRBG. Hybrid NRBGs are discussed in 8.3.5.

Despite providing additional unpredictability to the output of an RBG, the security of the RBG shall rest solely upon the primary entropy source. Hence, the output of an RBG shall remain secure even if the output of all of the additional entropy sources are known to an adversary and/or if an adversary has a certain measure of influence over the output of the additional entropy sources.

8.3.4.2 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:

a) primary entropy source is insufficiently reliable from a failure perspective. In this case, the additional entropy source shall satisfy the same requirements as the primary entropy source;

b) 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 satisfy the same requirements as the primary entropy source. Alternatively, instead of including an additional entropy source, it may be acceptable to solve this problem by using the NRBG only for initial seed generation for a DRBG; and

c) 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 as an entropy source. In this case, the additional entropy source shall satisfy the conditions given in 8.3.5.

8.3.4.3 Optional requirements for NRBG additional entropy sources

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

1. 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 entropy source), or a completely different component or process, this source should 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 facilitates the design and evaluation or independent verification processes by allowing the primary and secondary entropy

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

sources to be analysed separately, and also reduces the likelihood of a failure in the primary entropy source.

2. The RBG output should remain unpredictable even if the attacker can adaptively choose the values of the additional entropy sources; it should be unable to predict the next bit produced by the RBG with probability significantly greater than one half.

3. The secondary entropy 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 may not be free of adversary knowledge or influence. In this case, the additional entropy source should satisfy the same requirements as the primary source, although it may be acceptable for it to be somewhat more deterministic. That is, actions by the user or factors from the system environment could influence (although not completely determine) the output from this source in a perceptible way.

8.3.5 Hybrid NRBGs

An NRBG is considered a hybrid NRBG if it e.g. takes a deterministic source as an additional input. The main advantage of a hybrid NRBG is that its unpredictable output is now parameterised by a seed value. This may allow for extra security as knowledge of the output of the primary entropy source may now not be enough to allow an adversary to break the scheme, or it may allow a single non-deterministic entropy source to be used by several RBGs with different seed values without compromising security.

The extra functional requirements of a hybrid NRBG are as follows.

1. An adversary should not be able to predict the next bit with the probability significantly greater than one half, even if the attacker has complete control over the seed.

2. No adversary should be able to recover any information about the seed value by observing the output of the RBG.

3. No unauthorised person should be able to manipulate, influence or update the seed value.

8.4 NRBG additional inputs

8.4.1 NRBG additional inputs overview

The operation of an NRBG may require it to take certain public and/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 a correctly authenticated and authorised person or 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 requirement of any and 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 shall provide uninterruptible power which is a system issue and beyond the scope of this International Standard.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

8.5 NRBG internal state

8.5.1 NRBG internal state overview

This component consists of information that is carried over between calls to the NRBG, and all the information that is processed during a request. For this reason an internal state is a mandatory requirement; however it is not compulsory that any portion of the internal state depend upon previous states, i.e., there is no mandatory requirement for any portion of the internal state to be carried over to the next NRBG call (e.g., in the coin flip example, no internal state is carried over from one coin flip experiment to the next). In such cases, the internal state of an NRBG is totally dependent on the output of the entropy source at the time that the NRBG is used.

However, by retaining this state information the NRBG can produce a random output as a function of not only the current input from the entropy source but also several (or all) previous inputs. This provides a layer of protection against entropy source failure or degradation, as well as compromise of the random output by an adversary who has knowledge of or influence on the entropy source.

The internal state will consist of two parts. The working state is the portion of the internal state that is processed in combination with entropy source data by the internal state transition function to produce the new internal state. This portion may consist of a random “pool,” in addition to any optional counters or other values. The second part of the internal state is a value referred to as a secret parameter. The secret parameter is an additional input to the internal state transition function, which customizes the function for that particular instance of the NRBG. As such, it serves as an additional layer of protection against a degraded or compromised entropy source. Depending on the handling of the secret parameter, it could also protect against a compromised working state.

Since the entropy source output shall initially be placed into an internal state, the working state is mandatory. All, part or none of the internal state may be carried forward to the next NRBG invocation. The use of a secret parameter is optional, but where used, it is typically preserved between NRBG invocations and can only be updated by authenticated and authorised person or personnel within the boundary of the operational policy. More information on the issues associated with control of secret parameter is provided in ISO/IEC 11770-1[6].

8.5.2 Requirements for the NRBG internal state

The functional requirements of the internal state are as follows.

1. The design of the NRBG shall protect against knowledge of or influence over the internal state by an adversary.

NOTE 1 A possible means of accomplishing the above requirement include assigning the internal state to a memory region accessible only to the NRBG, hosting the NRBG on a standalone computer or device, or through security policies that physically protect the system and its environment.

2. The secret parameter, if it exists, shall be protected from adversarial knowledge by the NRBG protection boundary, which has been designed to detect unauthorised penetration attempts.

3. The initial value of the secret parameter, if it exists, shall contain sufficient entropy to meet the security requirement of the application. This initial secret parameter can either be generated by the NRBG or by another NRBG. If the secret parameter is generated by the NRBG itself, the NRBG shall operate in a special mode dedicated for this purpose, where the resulting random output becomes the secret parameter.

   However, if additional security is deemed to be a requirement to protect against possible scenarios of the adversary gaining knowledge of, or influencing the entropy source, then the initial secret parameter generation process shall obtain additional entropy data either from another system component or through interaction with the user to be combined in some way with the entropy source data (i.e., this additional entropy serves as a temporary secondary entropy source).

NOTE 2 Examples of user interaction might consist of, but are not limited to; key presses, timings between key presses or mouse movements.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

4. In the event that there are values that the secret parameter should not take (for example, “weak” cryptographic keys) then the secret parameter shall be tested to make sure that these secret parameter values are not used.

5. The secret parameter, if it exists, shall be replaced periodically. This supports the objective of forward and backward secrecy. The secret parameter should only be updated through commands issued by properly authenticated and authorised personnel within the boundary of the operational security policy.

6. If a secret parameter exists and unless it is obtained from another NRBG, the secret parameter replacement or updating process shall involve the entropy sources and internal state transition function.

NOTE 3 A reasonable replacement scheme would be simply to replace or to add the current secret parameter with ordinary random output from the NRBG that has not and will not be used for any other purpose.

7. If the entropy source(s) fail and the working state becomes compromised, the NRBG shall resist any attempts to force it to produce predictable output.

NOTE 4 This could be achieved by either ensuring that the NRBG ceases operation when the entropy source(s) fail, or, if the NRBG includes a secret parameter, by ensuring that the NRBG continues to operate in a manner no less secure than a DRBG. In particular, the secret parameter should be of sufficient length to resist any form of cryptanalytic attack against the NRBG, including exhaustive search.

8.5.3 Optional requirements for the NRBG internal state

The optional, but recommended features of the internal state are as follows.

1. The size of the internal state, in bits, should be sufficient to enable the NRBG to continue to act as a DRBG if the entropy sources fail or become completely known to, or controlled, by an adversary. If the entropy source data becomes constant, the maximum possible cycle length is bounded by the size of the internal state, and this places an upper bound on the work an adversary would need to perform to recover the internal state (through exhaustive search).

2. The secret parameter should be preserved between operational sessions in order to provide the NRBG with a unique state having sufficient entropy at each power-up initialisation without having to immediately create a new secret parameter value. If this is done, it shall be preserved in a way that protects it from adversarial access.

NOTE This protection could take the form of storage in a memory area accessible only to the NRBG process, storage in encrypted form, or storage in a removable token.

8.6 NRBG internal state transition functions

8.6.1 NRBG internal state transition functions overview

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

However, the main function of the internal state transition functions is to control the “carry over” parts of the internal state between invocations of the NRBG. This functionality is optional but recommended. Such a function will typically work in two parts.

1. It will carry the secret parameter over without change.

2. It will update the working state using a function that depends upon the current working state and (optionally) the secret parameter.

Figure 4 shows an example of an internal state transition function with a secret parameter.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

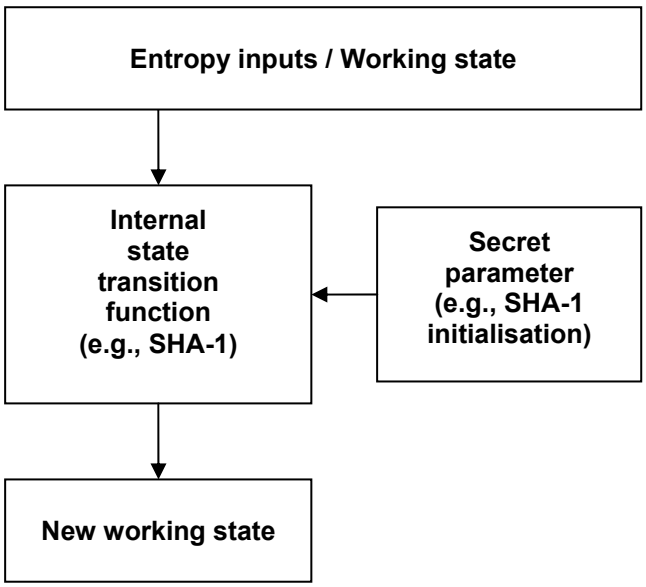

Figure 4 — Example of the internal state transition function

NOTE In Figure 4, the internal state transition function is SHA-1 and the secret parameter is the 160 bit application specific constant for SHA-1.

8.6.2 Requirements for the NRBG internal state transition functions

The use of an internal state transition function is mandatory only in that it needs to collect and store the output of the entropy sources. If this is the only function that the internal state transition functions provide then there are no additional requirements that the internal state transition functions shall satisfy.

If the internal state transition functions produce a new working state by combining the previous internal state with the output of the entropy source, then a functional requirement of the internal state transition function is as follows.

For each replacement of the entropy accumulating portion of the internal state, the entropy source data processed by the internal state transition function shall be of sufficient quantity to contain at least as many bits of entropy as the rated security strength (in bits) of the NRBG.

8.6.3 Optional requirements for the NRBG internal state transition functions

The optional, but recommended features of the internal state transition functions are as follows.

1. The internal state transition functions may achieve backward secrecy through appropriate use of a one-way function, such as a cryptographic hash-function.

2. The internal state transition functions should have the property that all of the bits in the working state and the entropy source input shall influence the internal state transition functions output bits.

3. The operation of the internal state transition function should be protected against observation and analysis via power consumption, timing, radiation emission, or other side channel attacks. The values that the internal state transition function operates on (internal state, secret parameter and entropy source input) are the critical values upon which confidentiality of random output is based. Side channel analysis could potentially defeat this confidentiality.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

8.7 NRBG output generation function

8.7.1 NRBG output generation function overview

This component provides random output to the requesting application by processing all or a subset of the bits in the current internal state, (both working state and, possibly, secret parameter), and any optional additional inputs. The output generation function serves as an important component in obtaining backward and forward secrecy. The component provides backward secrecy by preventing the random output from revealing information about the previous or current values of the internal state, entropy source inputs, or random outputs.

8.7.2 Requirements for the NRBG output generation function

The functional requirements for the output generation function are as follows.

1. The output generation function shall not introduce biases into the random output.

2. The random output from the output generation function shall pass the statistical health tests when they are required (see 8.8.5).

3. The output generation function shall process at least as many bits from the internal state as the number of bits in each random output block produced by the output generation function. Depending on the type of output generation function used, it may be necessary to process significantly more than this number of bits from the internal state.

4. The output generation function shall not reuse data from the working state portion of the internal state when providing random data to the requesting application. That is, either (a) the internal state transition function shall replace sufficient portions of the working state between calls to the output generation function to ensure no reuse or (b) the output generation function shall use a previously unused portion of the working state to ensure that data is not reused.

5. The output generation function shall not leak any information about the internal state that may potentially enable future output to be compromised.

8.7.3 An optional requirement for the NRBG output generation function

The following requirement is optional. It may be the case that in certain designs the inclusion of this requirement is highly recommended.

The output generation function should not be able to be inverted to reveal information about the internal state. That is, knowledge of the random output produced by the output generation function should not reveal any information about the current state.

8.8 NRBG health tests

8.8.1 NRBG health tests overview

This component ensures that the overall NRBG process continues to operate correctly and that the NRBG output continues to be random. These tests should detect failures in the NRBG and prevent the NRBG from being used until the health tests can be successfully passed. These tests are an integral part of the NRBG design; they are performed automatically at power-up or initialisation, without intervention by other applications, processes, or users, and may also be requested by the user at any time.

The health tests presented in the following text include three sets of tests; tests on the deterministic components of the NRBG, tests on the entropy sources, and tests on the random output produced by the NRBG.

The tests on the deterministic components shall apply to all NRBG designs. There may be cases where the rate of input entropy or random output is too low to feasibly implement all of the statistical health tests on

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

either the entropy sources or the NRBG output. In such cases the designer may attempt to modify the tests or test thresholds, to permit smaller sample sizes, while keeping the Type 1 error probability approximately the same.[1)]

General requirements and requirements that apply to each of these sets of tests are presented in the following four sub-clauses. In some cases, the NRBG may have additional features or functionality not addressed by this International Standard, and may include specialised tests. In these cases, the designer shall thoroughly document the objectives of these additional features and the basis for the additional tests.

The frequency at which health tests need to be performed depend on the overall design of the RBG. For example, if it can be ensured that any failure of the entropy source can be detected quickly, as guaranteed by frequent testing of the entropy source, the deterministic component may be simplified.

8.8.2 General NRBG health test requirements

The functional requirements for all three categories of health tests introduced above are as follows.

1. The NRBG shall automatically perform thorough health tests at each power-up or initialisation.

2. The NRBG shall allow for health tests (on the entropy sources, deterministic components and random output) to be performed “on demand.”

3. All outputs from the NRBG shall be inhibited while health tests are being performed, in order to conceal information about the operation of the NRBG and to prevent the release of any information about possible failures (this includes the health tests on deterministic components in 8.8.3, health tests on entropy sources in 8.8.4 and health tests on random output in 8.8.5). Data that has successfully passed all tests on random output may be used as random output following the completion of all health tests.

NOTE An exception to this requirement may exist if health tests are continually applied to the NRBG (e.g., health tests on the entropy source or statistical tests on the output). In this scenario, it would be impractical if not impossible to meet this requirement.

4. If the NRBG is implemented as software or firmware, the health tests performed at initialisation shall include an integrity check on the implementation code (RAM, ROM, or programmable logic device). Examples of ways to do this include a digital signature or message authentication code applied to the software or firmware.

If increased assurance is desired, the implementation of the NRBG should require the performing of the health tests with increased frequency during an operational session without serious degradation in performance, in addition to during power-up initialisation (a reasonable interval is one based on the frequency of requests for random bits).

8.8.3 NRBG health test on deterministic components

The objective of these tests is to ensure that the deterministic components of the NRBG continue to correctly process any possible set of inputs. Since, by definition, there is no unpredictability in these components, the accepted method of testing is to use known-answer tests. Such tests initialise the component or function to a fixed initial state, input a fixed input to the function, then compare the resulting output with the correct output computed previously by an independent implementation of the function (e.g., a verified computer simulation used during NRBG development) and stored with the NRBG implementation.

The functional requirements for the health tests on the deterministic components are as follows.

---

1) For the purposes of this International Standard, a Type 1 error is defined in terms of a test of a hypothesis in which an assessment of a random sequence results in a decision that the sequence is not random when in fact it is random. Additional information about Type 1 errors may be found at [8].

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

1. The known-answer tests shall be included in the overall health tests performed at each power-up and/or re-initialisation, and on demand. The known-answer tests should be included in the overall health tests performed at periodic intervals.

2. The comparison sequence (the result to be compared with the known answer) produced for any known-answer test shall be sufficiently long so that the probability of passing the test with failed or degraded components is acceptably low. Since 32-bit checksums are used in many information assurance applications, 32 bits is a recommended length for known-answer test values.

3. The known-answer tests shall include the entire NRBG, including not only each component of the NRBG, but also the overall NRBG control components. This may be done by setting the internal state to a fixed pattern, blocking the entropy source data and replacing the entropy source with a fixed bit sequence, and running the NRBG process in its operational mode. This action will produce an output sequence of at least the length determined according to point #2 above. The output is then compared to the predetermined value developed through independent implementation or simulation.

4. The known-answer tests shall exercise each aspect of the function being tested. This generally requires that the fixed input pattern be long enough to provide a representative sample of possible inputs to each major functional component of the function being tested.

5. All output from the NRBG shall be inhibited while known-answer health tests are being performed, in order to conceal information about the operation of the NRBG and to prevent the release of any information about possible failures. Data that has successfully passed the known-answer health tests shall not be used as random output following the completion of all health tests. The current value of the NRBG internal state following successful completion of the known-answer test shall not be used.

The known-answer tests on the deterministic components of the NRBG may be eliminated in favour of implementing the NRBG as two redundant and independent processes (other than the entropy sources) whose outputs are continuously compared. In this case, a mismatch shall be handled as a health test failure, with entry to the error state.

8.8.4 NRBG health tests on entropy sources

The objective of these tests is to detect variation in the behaviour of the entropy source from the intended behaviour. Since the entropy source is assumed to not produce unbiased, independent binary data in the vast majority of cases, traditional randomiser tests (e.g., monobit, chi square, and runs tests that test the hypothesis of unbiased, independent bits) will virtually always fail and thus not be useful. In general, tests on the entropy sources will have to be tailored carefully to the entropy source, taking into account the non-uniform statistical behaviour of the correctly operating entropy source.

For non-deterministic entropy sources obeying very complicated statistical models, and for non-physical entropy sources in particular, it may not be feasible to develop statistical tests that correspond precisely to a statistical model of the entropy source. In these cases it may be more appropriate to identify simpler tests that, instead of determining whether a statistic calculated from a data sample falls within an acceptable range, determine instead whether the data sample contains any occurrences of values known to be associated with failures. Selection of the patterns used for such tests should take into account the entropy source's likely failure behaviour.

The functional requirements for the health tests on the entropy sources are as follows.

1. The tests on the entropy sources shall be included in the overall health tests performed at each power-up and/or re-initialisation, at periodic intervals during use, and on demand.

NOTE 1 For example, the periodic interval for tests on the entropy source could be determined by the module's implementation policy.

2. As a minimum, each entropy source shall be tested for activity. That is, the test shall collect a quantity of data from the source and confirm that it does not consist solely of a constant output. (Constant outputs are those consisting only of a single value of the digitised entropy source output. For example, if the noisy

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

diode in the example produced the value 0110 at each sampling, it would fail an activity test.) The size of the data sample collected will depend on the characteristics of the entropy source, and shall be chosen such that when the entropy source is operating correctly, the probability of no activity within a sample of that size is acceptably low (10^{-4} is a recommended value for this Type 1 error rate).

3. The tests applied to each of the entropy sources, and rationale for their appropriateness, shall be thoroughly documented. The rationale shall indicate why the tests are believed to be very likely to detect failures in the entropy sources.

4. If any of the health tests on the deterministic components return a failure result, the NRBG shall enter an error state and indicate a failure condition. The NRBG shall not perform any random output generation while in the error state. In order to exit the error state, the NRBG shall require some form of intervention, preferably user intervention (e.g., power cycling or re-initialisation), followed by successful passing of the health tests.

NOTE 2 To recover or exit from an error state, an NRBG will be required to follow its maintenance procedures.

Although optional, it is recommended that the tests on each entropy source should include tests that take into account the known characteristics of the entropy source.

8.8.5 NRBG health tests on random output

The objective of these tests is to provide a final check on the randomness of the output from the NRBG. In general, the inclusion of the internal state transition function and output generation function results in the health tests on the random output from an NRBG playing a smaller role than they would if the tests were applied directly to output from a non-deterministic entropy source. These functions will typically do such thorough mixing that even a complete failure of the entropy sources would not cause detectable statistical irregularities in the random NRBG output. This is a consequence of the requirement that the NRBG continue to operate as a DRBG if the entropy sources fail or come under the influence of an adversary. However, statistical tests on random output are still useful, and are addressed in the requirements below.

The functional requirements for the health tests on the random output are as follows.

NOTE 1 The functional requirements address mainly statistical tests. As indicated above, usually, statistical tests are ineffective to validate the quality of the random output when strong cryptographic post-processing is applied.

1. The tests on the random output shall be included in the overall health tests performed at each power-up and/or re-initialisation, at periodic intervals during use, and when requested by the user.

2. The NRBG shall be tested for failure to a constant value by performing a test on each block of random output produced by the out generation function. See 9.8.8 for continuous RBG test specifics.

3. If any of the health tests on the random output return a failure result, the NRBG shall enter an error state and indicate a failure condition to the application or user. The NRBG shall not perform any random output generation while in the error state. The NRBG shall require user intervention (e.g., power cycling or re-initialisation), followed by successful passing of the health tests, in order to exit the error state.

NOTE 2 For example, to recover or exit from an error state, an NRBG will be required to follow its maintenance procedures.

Optionally is the following health test on the random output. When this recommendation is chosen, the health test shall meet functional requirement #1 above. As a minimum, the NRBG health test should include the following set of tests on a sequence of 20,000 bits of random output from the NRBG.

The overall set of tests on the NRBG is considered to have passed if all four individual tests are passed. The indicated thresholds below correspond to a Type 1 error probability of 10^{-4}.

NOTE 3 These four tests are rooted in FIPS 140-2 [9] and also adapted in AIS-31[5].

Documents
Document Sharing Hub

© ISO/IEC 2011 – All rights reserved

Full document details: BS ISO/IEC 18031:2011+A1:2017

a. Monobit test: Let X be the number of ones in the sample. The test is passed if $9725 < X < 10275$.

b. Poker test: Divide the sequence into 5,000 consecutive four-bit segments. Count the number of occurrences of each of the sixteen possible four-bit values. Let $f(i)$ be the number of occurrences of the four-bit value i (where $0 \le i \le 15$). Evaluate the following: $X = \frac{16}{5000}\sum_{i=0}^{15} f(i)^2 - 5000$. The test is passed if $2.16 \le X \le 46.17$.

c. Runs test: A run is defined as a maximal sequence of consecutive bits of either all ones or all zeroes. The occurrences of runs for both consecutive zeroes and consecutive ones of all lengths from one to six should be counted and stored. The test is passed if these counts are each within the corresponding interval specified in Table 1 below. This shall hold for both the zeroes and ones. For the purposes of this test, runs of length greater than six are considered to be of length six.

d. Long runs test: A long run is defined to be a run of length 27 or more of either zeroes or ones. The test is passed if there are no long runs.

NOTE 4 The rationale for the runs and long runs tests is discussed in Annex J.

Table 1 — Runs test intervals

Run Length	Required Interval
1	2,315 – 2,685
2	1,114 – 1,386
3	527 – 723
4	240 – 384
5	103 – 209
6+	103 – 209

Additional tests may be performed on the random output if:

1. it is deemed that greater assurance is required. If such is the case then the tests in the optional requirement above should be augmented or replaced by more comprehensive tests (sample sizes or additional statistical tests having a Type 1 error probability of 10^{-4} or less); and/or

2. a health test on random output returns a failure result; the NRBG should repeat the test a moderate number of times, but shall not exceed three. If the random output passes the test during this set of attempts, the NRBG can resume normal operation.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

8.9 NRBG component interaction

8.9.1 NRBG component interaction overview

The components of an NRBG described in this International Standard are designed to accomplish certain security objectives by satisfying the specific objectives and requirements. This section presents some additional requirements involving the interrelationships among the components.

8.9.2 Requirements for NRBG component interaction

1. The interaction and timing of the internal state transition function and the output generation function shall ensure that the rate at which the internal state transition function processes additional entropy source input is sufficient to provide usable data for use by the output generation function, with the additional constraint that the output generation function shall not reuse data from the working state. Note that this does not necessarily require the entropy source and internal state transition function to be operating when the output generation function produces output. For example, the working state could be much larger than its minimum acceptable length (i.e., the random output block length), providing a sufficient number of bits for the production of many random output blocks.

2. The processing of entropy source input and internal state (both working state and secret parameter) by the internal state transition function shall be such that the entropy source input, working state, and secret parameter each independently contribute entropy into the working state after each application of the internal state transition function, regardless of the entropy contribution from the others. This allows independent safety margins on the maintenance of entropy within the NRBG.

8.9.3 Optional requirements for NRBG component interaction

The optional, but recommended features applying to the interaction of the components of an NRBG are as follows.

1. The functional components of the NRBG and the interaction among these components should be designed such that, if the entropy sources become completely degraded in a manner undetectable by the health tests, the remaining NRBG components should continue to operate and interact as a DRBG. A natural way to accomplish this is by designing the NRBG as a DRBG modified to operate on input from one or more entropy sources.

2. The NRBG should be designed so that the working state continues to accumulate influence from the entropy sources even when the output generation function does not require new working state data (this could be done as a background process when processor or system resources are available). This is especially recommended if there are long periods of time between application requests for random output, during which the internal state might be more vulnerable to observation due to the increased length of time that it would otherwise remain unchanged.

9 Overview and requirements for a DRBG

9.1 DRBG overview

A DRBG uses an approved deterministic algorithm to produce a pseudorandom sequence of bits from an initial value called a seed, along with possible other inputs. Because of the deterministic nature of the process, a DRBG is said to produce “pseudorandom” rather than random bits, i.e., the string of bits produced by a DRBG is predictable and can be reconstructed, given knowledge of the algorithm, the seed and any other input information. However, if the input is kept secret, and the algorithm is well designed, the bit strings will appear to be random.

A DRBG is classified into two types depending on its inputs. Although a DRBG shall take a seed value as its primary entropy source, it may also take additional entropy sources as inputs. If these additional entropy sources are all deterministic (i.e. seed values) then the DRBG is called a pure DRBG. If at least one of these

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

entropy sources is non-deterministic, then the DRBG is called a hybrid. A hybrid DRBG shall satisfy all the requirements of a pure DRBG plus it shall also satisfy some extra security requirements discussed in 9.3.4.

At any given time after a seed has initialised a DRBG, a DRBG exists in a state that is defined by all prior input information. The primary seed can be thought of as defining different “instances” of a DRBG, each of which outputs a sequence of output bits (possibly depending on a series of inputs, or the values of the additional entropy sources).

The security of the DRBG should depend only on the primary entropy source, although other entropy sources can be used to provide additional entropy to maintain the unpredictability of the output even if the primary seed is compromised. Hence, the primary seed shall provide sufficient entropy so that the desired level of security is achieved by the DRBG.

The seeds for a DRBG require that the entropy be provided by an entropy input source (e.g., an NRBG). The security of an implementation that uses a DRBG is a system implementation issue; both the DRBG and its entropy input source shall be considered.

For the purposes of this International Standard, a DRBG shall meet the requirements specified in Clause 5 and 6. DRBG unique objectives and requirements, beyond those specified in Clause 5 and 6, which will be imposed on DRBG designs are detailed in 9.3 – 9.9. DRBG examples can be found in Annex C.

9.2 Functional model of a DRBG

This section will introduce a specialisation of the components and general model, for the case of a DRBG. It will describe the general operation of a DRBG, including the objectives each DRBG functional component is intended to accomplish.

Figure 5 provides a functional block diagram for a DRBG that satisfies this International Standard. It is important to note that the components shown do not necessarily have to be implemented as actual separate program routines, but do need to be implemented functionally in some sense. Each of the components, and the objectives and requirements on these components, 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 will be required in a DRBG. In some applications, there may be a legitimate argument that none of the requirements for a given component are applicable. If this can be adequately justified and documented, that component may be omitted from the DRBG, 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. For example, there are two common scenarios in which a DRBG may wish to update a seed – if a seed has been used for “too long” or it is suspected that an attacker has exerted some malicious control over the internal state. A re-seeding capacity is, therefore, not required if the device ceases operation after some limit (time, number of calls, etc.) and the internal state is suitably protected (say, by protective hardware).

The following is an overview of the way in which these components interact to produce pseudorandom output. The seed is either directly or indirectly supplied by a non-deterministic entropy source.

The seed can be directly supplied in one of two ways. The first way to directly supply a seed would be if the seed were produced by an NRBG that meets the requirements of this International Standard (i.e., an approved NRBG). The second way would be if the seed were supplied by an entropy source internal to the DRBG. Such an entropy source would have to be tested to make sure that it produces output with a sufficient entropy rate in a manner similar to an NRBG.

The seed is indirectly supplied if the seed is produced by another DRBG meeting the requirements stated herein (i.e., an approved DRBG), which in turn shall be correctly seeded.

After the seed is initially supplied it is loaded into the internal state by a specialised internal state transition function referred to as seeding. Access to this specialised internal state transition function should only be available to authorised parties, and at no point shall it be possible for this seed value to be observed or altered.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

After the DRBG is supplied with an initial seed and made ready for use, it will, on demand, do two things: it will update its internal state and output a block of random appearing data. The internal state will be updated by a deterministic internal state transition function, referred to as *updating internal state* in Figure 5, that takes as input the current internal state, any additional inputs supplied by the user and the outputs of any non-deterministic entropy source. In particular, it will not take the original seed as input unless the seed is held in the internal state specifically for this purpose. The output of the DRBG will be computed by a deterministic output generating function, which takes the current (just updated) internal state as input.

It is often useful to think of the internal state as being the combination of a working state that is continually updated, and a series of secret parameters. These secret parameters often control the operation of the internal state transition functions and output generating function in a manner similar to cryptographic keys (and will indeed often be cryptographic keys). The control of these secret parameters is outside the scope of this International Standard.

NOTE 1 Key management guidance may be found at ISO/IEC 11770. [6]

Typically, a DRBG will also contain a mechanism to update the internal state in the event that a new seed value is supplied. Such an update will be performed by a specialised internal state transition function and is known as reseeding. The internal state after reseeding may or may not depend upon the previous internal state depending on the reseeding method but may merely involve clearing the DRBG's working state and loading the new seed in a similar manner to the initial seeding operation.

A secure DRBG will also include mechanisms designed to increase the likelihood of continued secure operation in the event of failures or compromises. The potential for failure to perform as intended is addressed through the inclusion of health tests on the various components, such as known-answer tests and continuous output tests.

Each of the DRBGs specified has been designed to provide forward and backward secrecy when observed from outside the DRBG boundary, given that the observer does not know the seed or any state values.

When observed from within the DRBG boundary, each of the specified DRBGs provides backward secrecy.

In the case of a DRBG, forward secrecy depends on the use of a secret seed and the insertion of additional input during each generation process with sufficient entropy to meet the security requirements. Forward secrecy may be provided during one instance of a DRBG by the addition of user input. However, the degree of forward secrecy depends on the amount of entropy introduced by the user input. If the user input introduces entropy that is at least equal to twice the strength of the DRBG for every request for pseudorandom bits, then "full" forward secrecy is usually provided.

NOTE 2 For example, if a DRBG is to provide 128 bits of security, then to have the full forward secrecy, we would need both the seed and each additional input to have at least 128 bits of entropy.

This may not be practical for many applications. However, user input with even a small amount of entropy provides some degree of forward secrecy. It may be appropriate for an application to require user input with high entropy for critical applications (e.g., the generation of digital signature keys).

With respect to backward secrecy, a properly implemented DRBG is instantiated using a secret seed. The seed is used to determine the initial state of the DRBG. The output is some function of the internal state, and the internal state is updated during each request for bits. If the seed or any state becomes known, prior outputs could be determined unless a cryptographically strong one-way function is used in the DRBG design to transition from one internal state to the next internal state.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

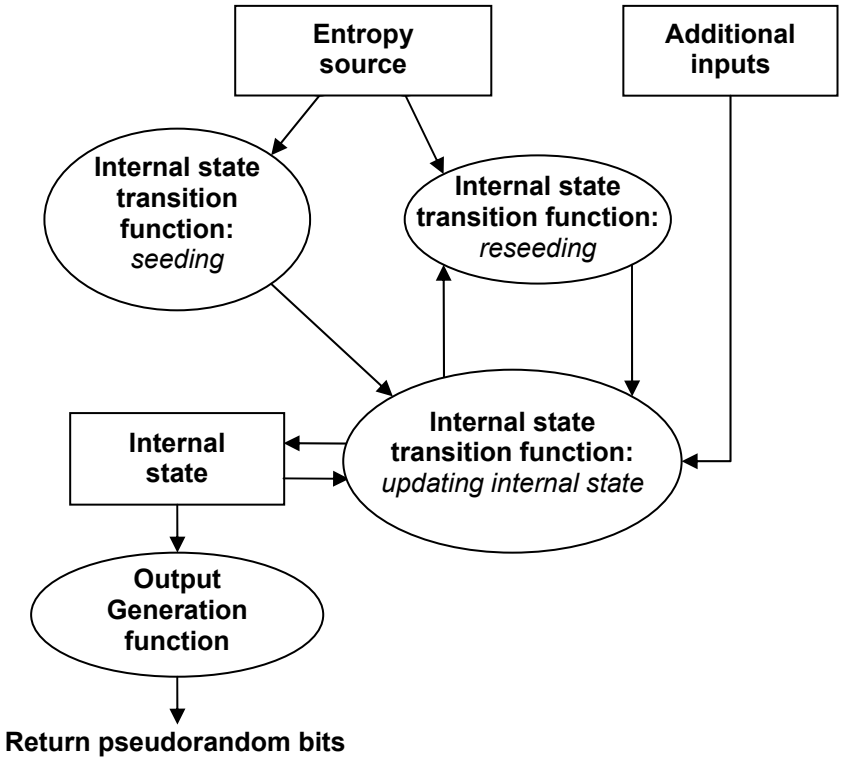

Figure 5 — DRBG model

9.3 DRBG entropy source

9.3.1 Primary entropy source for a DRBG

The primary entropy source for a DRBG is a seed value. This seed value shall be drawn from an entropy source with a given entropy output rate such as an NRBG, and shall be input to the DRBG prior to requesting pseudorandom bits from the DRBG. For more details about the requirements of the source of the seed, see 9.3.2.

The seed, seed size and the entropy (i.e., randomness) of the seed shall be selected to minimize the probability that the sequence produced by one seed significantly matches the sequence produced by another seed, and to reduce the probability that the seed can be guessed or exhaustively tested. Since this International Standard does not require full entropy for a seed but does require sufficient entropy, the length of the seed may be greater than the security strength specified, in order to accommodate the needed entropy. That is, the seed length shall, at minimum, be the number of bits in the specified security strength, however, the seed length should be more than the minimum in order to increase assurance of sufficient entropy and address any concerns of possible reuse of a seed. Note that using the minimum seed length is acceptable only if the seed is generated by a full entropy source, such as an NRBG or DRBG seeded by an NRBG. As the general case is that the seed may be of variable size, the specification of the DRBG algorithms make this assumption, although a specific implementation may be optimised to support a specific length seed.

Entropy analysis for a seed is a critical component to the security assurance associated with the DRBG. The importance of entropy analysis is most readily demonstrated when something goes wrong. For example, if the security strength of 80 bits is desired and believed to be achieved, but only 40 bits is actually achieved, this amount of entropy is exhausted quite easily. Even if the actual entropy is 60 bits, a determined adversary could still exhaust the entropy at a reasonable cost.

The generation or entry of entropy into a DRBG using an insecure method may result in voiding the intended security assurances. To ensure unpredictability, care should be exercised in obtaining and handling seeds. The seed and its use by a DRBG shall be generated and handled as follows.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

1. Seed construction: A seed shall include entropy input and should include a personalisation string (see 9.4 for further information on personalisation string) when the source of entropy is not considered strong enough. The combination of the entropy input and the optional personalisation string is called the seed material. A derivation function shall be used to distribute the entropy input across the entire seed (e.g., so that the seed is not constructed with all the entropy on one end of the seed) whenever a personalisation string is used, or a personalisation string is not used and the entropy input is not independent and uniformly distributed throughout the entropy input string. Whether or not the personalisation string is present, the resulting seed shall be unique.

2. Seed use: DRBGs may be used to generate both secret and public information. In either case, the seed shall be kept secret. A single instantiation of a DRBG should not be used to generate both secret and public values. Cost and risk factors shall be taken into account when determining whether different instantiations for secret and public values can be accommodated.

NOTE 1 If the DRBG has backward and forward secrecy and if the implementation is resistant against side-channel attacks one may neglect this security feature to not use a single instantiation of a DRBG to generate both secret and public values without loss of security.

A seed that is used to initialise one instantiation of a DRBG shall not be intentionally used to reseed the same instantiation or used as a seed for another DRBG.

3. Seed entropy: The entropy input for the seed shall contain sufficient entropy for the desired level of security, and the entropy shall be distributed across the seed. A consuming application may or may not be concerned about collision resistance. In order to accommodate possible collision concerns, a seed shall have entropy that is equal to or greater than 120 bits or the required security strength for the consuming application; whichever is greater (i.e., entropy ≥ max (120, *security_strength*)). If a selected DRBG and the seed are not able to provide the strength required by the consuming application, then a different DRBG shall be used.

4. Seed size: The minimum size of the seed depends on the selected DRBG, the security strength required by the consuming application and the entropy source. The seed size shall be at least equal to the required security strength in bits and may be larger, depending on the entropy rate and assurances thereof of the entropy source (see above discussion). For example, if 160 bits of entropy are required, the quality of the entropy source may necessitate a seed size of 240 bits or more to achieve the 160 bits of entropy.

5. Seed privacy: Seeds shall be handled in a manner consistent with the security required for the target data. For example, if the only secrets in a cryptographic system are the keys; then the seeds used to generate keys shall be treated as if they are keys.

6. Seed Usage period: A DRBG seed shall have a specified usage period, after which it shall no longer be used. Seeds shall have a specified finite seedlife. The seed shall be updated periodically, or the seed shall be rendered inoperable by replacement after its seedlife. If seeds become known (i.e., the seeds are compromised), unauthorised entities may be able to determine the DRBG output. The usage period may be given by a time span or a maximum number of outputs that may be generated with that same seed.

In some implementations (e.g., smartcards), an adequate reseeding process may not be possible, and reseeding may actually reduce security. In these cases, the best policy might be to replace the DRBG, thus obtaining a new seed in the process (e.g., obtain a new smart card). In other applications, reseeding is an appropriate design choice. Reseeding (i.e., replacement of one seed with a new seed) is a means of recovering the secrecy of the output of the DRBG if a seed should happen to become known to an adversary. Periodic reseeding is a good countermeasure to the potential threat that the seeds and DRBG output become compromised. However, the result from reseeding is only as good as the NRBG (or chain of DRBGs that is initiated by an NRBG) used to provide the new seed. Generating too many outputs from a seed (and other input information) may provide sufficient information for successfully predicting future outputs. Periodic reseeding will reduce security risks, reducing the likelihood of a compromise of the target data that is protected by cryptographic mechanisms that use the DRBG.

© ISO/IEC 2011 – All rights reserved

Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017

Reseeding of the DRBG shall be performed in accordance with the specification for the given DRBG. When a new seed is obtained during reseeding, the new seed shall be checked to assure that two consecutive seeds are not identical. More than one seed shall not be saved by the DRBG. A seed shall not be saved in its original form, but shall be transformed by a one-way process. When a new seed is generated and compared to the "old" seed (i.e., the transformed old seed), the transformed new seed shall replace the old transformed seed in memory. The old transformed seed shall be destroyed. If the new seed is determined to be identical to the old seed, another new seed shall be generated.

7. Seed separation: It is recommended that when resources permit (e.g., storage capacity), different (i.e., non-repeating) seeds should be used for the generation of different types of random data (i.e., the "instances" of the DRBGs should be different). For example, the seed used to generate public values should be different than the seed used to generate secret values. The seed used by a DRBG technique to generate asymmetric key pairs should be different than a seed used by the same (or a different) DRBG technique to seed other DRBGs, which should, in turn, be different than a seed used by the same (or a different) DRBG technique to generate symmetric keys. The seed used by a DRBG technique to generate random challenges should be different than the seed used by the same (or a different) DRBG technique to generate PINS or passwords. However, the amount of seed separation is a cost/benefit decision.

NOTE 2 If the DRBG has backward and forward secrecy and if the implementation is resistant against side-channel attacks one may use the random number generated by one instance of the DRBG for different types of random data without loss of security.

9.3.2 Generating seed values for a DRBG

A seed value for an approved DRBG shall be produced in one of three ways.

1. The seed shall be produced by an approved NRBG that produces output at a sufficient entropy rate.

2. The seed shall be produced by an approved DRBG that produces output at a sufficient entropy rate. This DRBG shall also have a seed, which shall be produced in accordance with these seed generation requirements. Hence, a chain of DRBGs can be envisaged that produce seeds for the next DRBG. However, this chain shall always begin with either an approved NRBG or an approved DRBG that produces its own seed values. In other words, if a seed is produced by an approved DRBG or chain of approved DRBGs, the highest level DRBG has to get its seed from an NRBG or a DRBG with an internal NRBG.

3. The seed shall be generated by an appropriate entropy source. This source may be biased and/or produce dependent bits. If a seed value is produced in this way then the developer shall assess the rate of entropy production of the source and ensure that the source meets all the requirements of entropy sources found in 6.2.2, 8.3 and 8.8.4.

9.3.3 Additional entropy sources for a DRBG

The operation of a DRBG may also include one or more additional entropy sources. An additional entropy source may be useful for a variety of reasons.

A DRBG may have an additional entropy source that is not a deterministic entropy source but rather a non-deterministic entropy source. A DRBG that also has a non-deterministic entropy source is known as a hybrid DRBG. Hybrid DRBGs are discussed in 9.3.4.

Despite providing additional unpredictability to the output of an RBG, the security of the RBG shall rest solely upon the primary entropy source. Hence, the output of an RBG shall remain secure even if the output of all of the additional entropy sources are known to an adversary and/or if an adversary has a certain measure of influence over the output of the additional entropy sources.

The advantage of using a non-deterministic entropy source as an additional entropy source is that it allows the output of the DRBG to be non-deterministic, and may help prevent cryptanalysis and/or add security features such as forward and backward secrecy.

© ISO/IEC 2011 – All rights reserved
Documents
Document Sharing Hub
Full document details: BS ISO/IEC 18031:2011+A1:2017