INTERNATIONAL STANDARD

ISO/IEC 18367

First edition
2016-12-15

Information technology — Security techniques — Cryptographic algorithms and security mechanisms conformance testing

Technologie de l'information — Techniques de sécurité — Essais de conformité des algorithmes cryptographiques et des mécanismes de sécurité

Reference number
ISO/IEC 18367:2016(E)

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

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

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

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

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

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword - Supplementary information

The committee responsible for this document is ISO/IEC JTC 1, *Information technology*, SC 27, *IT Security techniques.*

Introduction

This document describes cryptographic algorithms and security mechanisms conformance testing methods.

The purpose of this document is to address conformance testing methods of cryptographic algorithms and security mechanisms implemented in a cryptographic module. This will allow a complete security evaluation of both the cryptographic module and the implemented cryptographic algorithms and security mechanisms.

This document is related to ISO/IEC 19790 and ISO/IEC 24759. ISO/IEC 19790 specifies the security requirements for cryptographic modules. At a minimum, a cryptographic module implements at least one approved security function (i.e., cryptographic algorithm or security mechanism). ISO/IEC 24759 addresses the test requirements for each of the security requirements in ISO/IEC 19790. However, ISO/IEC 24759 does not address test methods for cryptographic algorithms and security mechanisms conformance testing.

Information technology — Security techniques — Cryptographic algorithms and security mechanisms conformance testing

1 Scope

This document gives guidelines for cryptographic algorithms and security mechanisms conformance testing methods.

Conformance testing assures that an implementation of a cryptographic algorithm or security mechanism is correct whether implemented in hardware, software or firmware. It also confirms that it runs correctly in a specific operating environment. Testing can consist of known-answer or Monte Carlo testing, or a combination of test methods. Testing can be performed on the actual implementation or modelled in a simulation environment.

This document does not include the efficiency of the algorithms or security mechanisms nor the intrinsic performance. This document focuses on the correctness of the implementation.

2 Normative references

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

ISO/IEC 14888-3:2016, *Information technology — Security techniques — Digital signatures with appendix*

ISO/IEC 19790:2012, *Information technology — Security techniques — Security requirements for cryptographic modules*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19790 and the following apply.

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

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

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

3.1
approval authority
any national or international organisation/authority mandated to approve and/or evaluate security functions

Note 1 to entry: An approval authority in the context of this definition evaluates and approves security functions based on their cryptographic or mathematical merits but is not the testing entity which would test for conformance to this document.

[SOURCE: ISO/IEC 19790:2012, 3.4]

3.2
approved mode of operation
set of services which includes non-security relevant services and at least one service that utilizes an approved security function or process

Note 1 to entry: Not to be confused with a specific mode of an approved security function, e.g. Cipher Block Chaining (CBC) mode.

Note 2 to entry: Non-approved security functions or processes are excluded.

[SOURCE: ISO/IEC 19790:2012, 3.7]

3.3
approved security function
security function (e.g. cryptographic algorithm) approved by an approval authority

3.4
black box
idealized mechanism that accepts inputs and produces outputs, but is designed such that an observer cannot see inside the box or determine exactly what is happening inside that box

Note 1 to entry: This term can be contrasted with *glass box* (3.12).

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

3.5
critical security parameter
CSP
security-related information whose disclosure or modification can compromise the security of a cryptographic module

EXAMPLE Secret and private cryptographic keys, authentication data such as passwords, PINs, certificates or other trust anchors.

[SOURCE: ISO/IEC 19790:2012, 3.18, modified]

3.6
cryptographic algorithm
well-defined computational procedure that takes variable inputs, which may include cryptographic keys, and produces an output

[SOURCE: ISO/IEC 19790:2012, 3.20]

3.7
cryptographic algorithm boundary
boundary encompassing the complete cryptographic algorithm implementation

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

[SOURCE: ISO/IEC 19790:2012, 3.21, modified]

3.9
firmware
executable code of a cryptographic module that is stored in hardware within the cryptographic boundary and cannot be dynamically written or modified during execution while operating in a non-modifiable or limited operational environment

EXAMPLE Storage hardware can include but is not limited to PROM, EEPROM, FLASH, solid state memory, hard drives, etc.

[SOURCE: ISO/IEC 19790:2012, 3.45]

3.10
functional specification
high-level description of the ports and interfaces visible to the operator and high-level description of the behaviour of the IUT

Note 1 to entry: Here, the IUT means a cryptographic algorithm implementation (see 3.13).

[SOURCE: ISO/IEC 19790:2012, 3.47, modified]

3.11
functional testing
testing of the IUT functionality as defined by the *functional specification* (3.10)

[SOURCE: ISO/IEC 19790:2012, 3.48, modified]

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

Note 1 to entry: This term can be contrasted with black box (3.4).

[SOURCE: ISO/IEC 18031:2011, 3.14]

3.13
implementation under test
IUT
implementation which is tested for conformance to the selected cryptographic algorithm or security mechanism standard

3.14
independent verification test
test verifying the conformance for algorithms where their outputs are non-deterministic (or randomized) for defined input vectors in an independent way, instead of literally following the algorithm steps

Note 1 to entry: The signature generation function of DSA involves per-message secret number internally, and therefore, the resultant signature cannot be derived from the input vectors in a deterministic way without the knowledge of per-message secret number. Here, the signature verification function can be used to verify the relation between the public key, message and resultant signature, without knowing the per-message secret number.

Note 2 to entry: Independent verification tests can involve the reference implementation of the inverse function of the selected cryptographic algorithm, e.g. using the digital signature verification function to verify the corresponding signature generation function.

Note 3 to entry: In contrast to the other KATs, the independent verification test does not prepare expected values in advance.

EXAMPLE Applying Miller-Rabin primality test to verifying prime number generation functions, independent of the implementation details of the IUT.

3.15
key agreement
process of establishing a shared secret key between entities in such a way that neither of them can predetermine the value of that key

[SOURCE: ISO/IEC 11770-3:2015, 3.18, modified]

3.16
key derivation function
KDF
function that outputs one or more shared secrets, for use as keys, given shared secrets and other mutually known parameters as input

[SOURCE: ISO/IEC 11770-3:2015, 3.22]

3.17
key derivation key
key that is used as input to the key expansion function to derive other keys

3.18
key establishment
process of making available a shared secret key to one or more entities, where the process includes key agreement and key transport

[SOURCE: ISO/IEC 11770-3:2015, 3.23]

3.19
key expansion function
function which takes as input a number of parameters, at least one of which is a MAC algorithm key, and which gives as output keys appropriate for the intended algorithm and application, and which has the property that it is computationally infeasible to deduce either the output without prior knowledge of the secret input or the secret input from the output

3.20
key extraction function
function which takes as input a number of parameters, at least one of which is secret, which gives as output a MAC algorithm key for use as input to a key expansion function, and which has the property that it is computationally infeasible to deduce either the output without prior knowledge of the secret input or the secret input from the output

3.21
keying material
data necessary to establish and maintain cryptographic keying relationships

EXAMPLE Keys, initialization values.

[SOURCE: ISO/IEC 11770-1:2010, 2.27]

3.22
key management
administration and use of the generation, registration, certification, deregistration, distribution, installation, storage, archiving, revocation, derivation and destruction of keying material in accordance with a security policy

[SOURCE: ISO/IEC 19790:2012, 3.64]

3.23
key transport
process of transferring a key from one entity to another entity, suitably protected

[SOURCE: ISO/IEC 11770-3:2015, 3.25]

3.24
known-answer test
KAT
method of testing a deterministic mechanism where a given input is processed by the mechanism and the resulting output is then compared to a corresponding known value

Note 1 to entry: The known-answer tests are designed to test the conformance of the implementation under test (IUT) to the various specifications of the referenced cryptographic algorithm.

Note 2 to entry: A known-answer test is considered as a kind of black box testing.

[SOURCE: ISO/IEC 18031:2011, 3.21, modified]

3.25
Monte Carlo test
MCT
subset of known-answer test utilising randomly generated input vectors and the corresponding known output result, designed to pseudo exhaust the presence of flaws by exercising the entire IUT in a manner that cannot be detected with the controlled input vectors

Note 1 to entry: The types of implementation flaws which can be detected by Monte Carlo tests include pointer problems, insufficient allocation of space, improper error handling and incorrect behaviour of the IUT.

3.26
multi-block message test
MMT
set of tests designed to test the ability of the implementation to process multi-block messages which will require the chaining of information from one block to the next

3.27
public security parameter
PSP
security-related public information whose modification can compromise the security of a cryptographic module

EXAMPLE Public cryptographic keys, public key certificates, self-signed certificates, trust anchors, one-time passwords associated with a counter and internally held date and time.

[SOURCE: ISO/IEC 19790:2012, 3.99, modified]

3.28
random bit generator
RBG
device or algorithm that outputs a sequence of bits that appears to be statistically independent and unbiased

[SOURCE: ISO/IEC 18031:2011, 3.29]

3.29
salt
random data item produced by the signing entity during the generation of message representative

Note 1 to entry: Also known as the randomizer in ISO/IEC 14888-3.

Note 2 to entry: Also known as the per-message secret number in Reference [16].

3.30
security function
cryptographic algorithms together with modes of operation, such as block ciphers, stream ciphers, symmetric or asymmetric key algorithms, message authentication codes, hash functions or other security functions, random bit generators, entity authentication and SSP generation and establishment all approved either by ISO/IEC or an approval authority

[SOURCE: ISO/IEC 19790:2012, 3.106]

3.31
sensitive security parameter
SSP
critical security parameter (CSP) or public security parameter (PSP)

3.32
shared secret key
key which is shared with all the active entities via a key establishment mechanism for multiple entities

Note 1 to entry: Also known as "shared secret."

[SOURCE: ISO/IEC 11770-5:2011, 3.28, modified]

3.33
simulation
exercise of source code (e.g. VHDL code) prior to physical entry into the module (e.g. an FPGA or custom ASIC)

Note 1 to entry: The behaviour of the source code within the simulator can be logically identical when placed into the module or instantiated into logic gates. However, many other variables exist that can alter the actual behaviour (e.g., path delays, transformation errors, noise, environmental, etc.). It is not guaranteed that the actual behaviour of the IUT is identical, as many other variables cannot be identified with certainty.

3.34
zeroisation
method of destruction of stored data and unprotected SSPs to prevent retrieval and reuse

[SOURCE: ISO/IEC 19790:2012, 3.134]

4 Symbols and abbreviated terms

AES	advanced encryption standard
ASN.1	abstract syntax notation one
CBC	cipher block chaining mode of operation
CCM	counter mode with cipher block chaining-message authentication code
CFB	cipher feedback mode of operation
CMAC	cipher-based message authentication code
CSP	critical security parameter
DER	distinguished encoding rules
DRBG	deterministic random bit generator
DSA	digital signature algorithm

EAL	evaluation assurance level
ECB	electronic codebook mode of operation
ECDSA	elliptic-curve digital signature algorithm
FFC	finite field cryptography
GCM	galois/counter mode
GMAC	GCM message authentication code
HMAC	keyed-hash message authentication code
IUT	implementation under test
KAS	key agreement scheme
KAT	known-answer test
KC	key confirmation
KDF	key derivation function
MAC	message authentication code
MCT	Monte Carlo test
MMT	multi-block message test
OFB	output feedback mode of operation
PKCS	public key cryptography standards
PSS	probabilistic signature scheme
RBG	random bit generator
RSA	algorithm developed by Rivest, Shamir and Adleman
SHA	secure hash algorithm
TDEA	triple data encryption algorithm

$\lceil X \rceil$	ceiling: the smallest integer greater than or equal to X. For example, $\lceil 5 \rceil = 5$ and $\lceil 5.3 \rceil = 6$
$X \oplus Y$	bitwise exclusive-or (also bitwise addition mod 2) of two bit strings X and Y of the same length
X || Y	concatenation of two bit strings X and Y in that order
x mod n	unique remainder r, $0 \leq r \leq n-1$, when integer x is divided by n. For example, 23 mod 7 = 2

5 Objectives

The requirements specified in this document are derived from the following objectives for cryptographic algorithm implementations to

— provide assurance that the cryptographic algorithm implementation adheres to the specifications detailed in the associated cryptographic standard, and

— detect implementation non-conformities made by implementers by testing the algorithm's specifications, components, features and/or functionality for correctness and completeness.

6 Types of cryptographic algorithms and security mechanisms from a conformance testing perspective

6.1 General

This document will address approved security functions from a conformance testing point perspective. In particular, this document will address those defined in Clause 2. It will include within its scope the associated security mechanisms of the cryptographic algorithms or the security mechanisms. The considered implementations can be software, firmware, hardware or a combination thereof.

6.2 Asymmetric key algorithms

This subclause describes the different types of asymmetric key algorithms.

Asymmetric key algorithms consist of asymmetric key cryptographic primitive(s) and supporting functions. Some of asymmetric key algorithms are non-deterministic, due to salt or internally generated random numbers. The implementation of asymmetric key algorithms would contain more conditional branches than symmetric key algorithms. In considering these aspects of asymmetric key algorithms, the known-answer test (see 7.2.2) and/or independent verification test (see 3.14) are applicable.

In addition to these testing methodologies, other conformance testing methodologies (e.g. source code inspection) are also applicable.

6.3 Digital signature

This subclause describes the different types of digital signature algorithms.

The same perspective as asymmetric key algorithms is still applicable to digital signature algorithms.

6.4 Digital signature with message recovery

This subclause describes the different types of digital signature with message recovery algorithms.

The same perspective as asymmetric key algorithms is still applicable to digital signature with message recovery.

6.5 Hashing algorithms

This subclause describes the different types of hashing algorithms.

Hashing algorithms will be dedicated hash functions, functions based on block cipher algorithms or functions based on modular arithmetic. In considering the nature of hashing algorithms, the known-answer test (see 7.2.2) and Monte Carlo test (see 7.2.4) are applicable. Other conformance testing methodologies (e.g. source code inspection) are also applicable.

6.6 Key establishment mechanisms

This subclause describes the different types of key establishment mechanisms.

Key establishment mechanisms consist of asymmetric/symmetric key cryptographic primitive(s) and supporting functions. Supporting functions can be hashing algorithms, random bit generation, asymmetric key pair generation function, and public key validation function.

In considering the complex nature of key establishment mechanisms, the known-answer test (see 7.2.2) and independent verification test (see 3.14) are applicable. As a prerequisite for this conformance

testing, underlying algorithm implementations have passed the conformance testing elsewhere in this document.

Note that some input parameters are transmitted through a communication channel. It should be tested through the conformance testing that an IUT has an ability to distinguish valid parameters with invalid parameters.

In addition to these testing methodologies, other conformance testing methodologies (e.g. source code inspection) are also applicable.

6.7 Lightweight cryptography

This subclause describes the different types of lightweight cryptography algorithms.

Lightweight cryptography includes asymmetric key algorithms, block ciphers and stream ciphers. So, the applicable perspective is the same as one for asymmetric key algorithms, block ciphers and stream ciphers.

6.8 Message authentication algorithms

This subclause describes the different types of message authentication algorithms.

Message authentication algorithms can consist of underlying algorithms (e.g. block cipher algorithms, hash algorithms).

In considering this aspect of message authentication algorithms, the known-answer test (see 7.2.2) is applicable. As a prerequisite for this conformance testing, underlying algorithm implementations have passed the conformance testing elsewhere in this document.

In addition to this testing methodology, other conformance testing methodologies (e.g. source code inspection) are applicable.

6.9 Random bit generator algorithms

6.9.1 Deterministic random bit generator algorithms

This subclause describes the different types of deterministic random bit generator algorithms.

Deterministic random bit generators can consist of underlying algorithms (e.g. block cipher algorithms, hash algorithms).

In considering this aspect of deterministic random bit generator, the known-answer test (see 7.2.2) is applicable. As a prerequisite for this conformance testing, underlying algorithm implementations have passed the conformance testing elsewhere in this document.

In addition to this testing methodology, other conformance testing methodologies (e.g. source code inspection) are also applicable.

6.9.2 Non-deterministic random bit generator algorithms

This subclause describes the different types of non-deterministic random bit generator algorithms.

Currently, there is no standard specification of NRBG in ISO/IEC 18031, so the associated conformance testing methodologies are not specified in this document.