INTERNATIONAL STANDARD

ISO/IEC 19896-2

First edition
2018-08

IT security techniques — Competence requirements for information security testers and evaluators —

Part 2: Knowledge, skills and effectiveness requirements for ISO/IEC 19790 testers

Techniques de sécurité IT — Exigences de compétence pour l'information testeurs d'assurance et les évaluateurs —

Partie 2: Exigences en matière de connaissances, de compétences et d'efficacité pour ISO / IEC 19790 testeurs

Reference number
ISO/IEC 19896-2:2018(E)

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

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. 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 of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 27, *IT security techniques*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

A list of all parts in the ISO/IEC 19896 series can be found on the ISO website.

Introduction

This document provides the specialized requirements to demonstrate knowledge, skills and effectiveness requirements of individuals in performing security testing projects in accordance with ISO/IEC 19790 and ISO/IEC 24759. ISO/IEC 19790 provides the specification of security requirements for cryptographic modules. Many certification, validation schemes and recognition arrangements have been developed using it as a basis. ISO/IEC 19790 permits comparability between the results of independent security testing projects. ISO/IEC 24759 supports this by providing a common set of testing requirements for testing a cryptographic module for conformance with ISO/IEC 19790.

One important factor in assuring comparability of the results of such validations or certifications is the knowledge, skills and effectiveness requirements of the individual testers responsible for performing testing projects.

ISO/IEC 17025, which is often specified as a standard to which testing facilities conform, states in 5.2.1 that "Personnel performing specific tasks shall be qualified on the basis of appropriate education, training, experience and/or demonstrated skills".

The audience for this document includes validation and certification authorities, laboratory testing accreditation bodies, testing projects schemes, testing facilities, testers and organizations offering professional credentials and recognitions.

This document establishes a baseline for the knowledge, skills and effectiveness requirements of ISO/IEC 19790 testers with the goal of establishing conformity in the requirements for the training of ISO/IEC 19790 testing professionals associated with cryptographic module conformance testing programs.

Annex D illustrates the usefulness of this document by validators within a validation program.

IT security techniques — Competence requirements for information security testers and evaluators —

Part 2:
Knowledge, skills and effectiveness requirements for ISO/IEC 19790 testers

1 Scope

This document provides the minimum requirements for the knowledge, skills and effectiveness requirements of individuals performing testing activities for a conformance scheme using ISO/IEC 19790 and ISO/IEC 24759.

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 17025, *General requirements for the competence of testing and calibration laboratories*

ISO/IEC 17825, *Information technology — Security techniques — Testing methods for the mitigation of non-invasive attack classes against cryptographic modules*

ISO/IEC 18367, *Information technology — Security techniques — Cryptographic algorithms and security mechanisms conformance testing*

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

ISO/IEC 19896-1, *IT security techniques — Competence requirements for information security testers and evaluators — Part 1: Introduction, concepts and general requirements*

ISO/IEC 20085-1, *Information technology — Security techniques — Test tool requirements and test tool calibration methods for use in testing non-invasive attack mitigation techniques in cryptographic modules — Part 1: Test tools and techniques*

ISO/IEC 20085-2, *Information technology — Security techniques — Test tool requirements and test tool calibration methods for use in testing non-invasive attack mitigation techniques in cryptographic modules — Part: 2 Test calibration methods and apparatus*

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

ISO/IEC 24759, *Information technology — Security techniques — Test requirements for cryptographic modules*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19896-1 and ISO/IEC 19790 apply.

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

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

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

4 Abbreviated terms

AES advanced encryption standard

HDD hard disk drive

RSA rivest-shamir-adleman

SHA secure hash algorithm

SSD solid state drive

5 Structure of this document

This document is divided into the following clauses: Knowledge (Clause 5), Skills (Clause 6), Experience (Clause 7), Education (Clause 8) and Effectiveness (Clause 9). Each clause corresponds to an aspect of the knowledge, skills, experience, education and effectiveness requirements of individuals performing testing activities as introduced in ISO/IEC 19896-1 for a conformance scheme using ISO/IEC 19790 and ISO/IEC 24759.

6 Knowledge

6.1 General

Knowledge is what a tester knows and can describe. Clauses 6 to 9 address education requirements and knowledge areas that are specifically needed for conformance testing to ISO/IEC 19790 and ISO/IEC 24759.

6.2 Tertiary education

6.2.1 General

Testers shall have educational qualifications such as an associate, bachelor, or higher degree that is relevant to the security requirements addressed in ISO/IEC 19790 and the test requirements in ISO/IEC 24759. The testers shall at a minimum demonstrate they have either:

a) successfully completed appropriate tertiary education with at least 3 years of study in disciplines related to IT or IT security; or

b) experience equivalent to the tertiary education in disciplines related to IT, IT security or IT system administration.

6.2.2 Technical specialities

In addition to the minimum level of educational requirements in 6.2.1, testers shall have educational qualifications such as an associate, bachelor, or higher degree that addresses the specific technical specialities. Examples of specific technical specialities include:

— cryptographic concepts;

— engineering technology;

— electrical engineering;

— mechanical engineering;

— material engineering;

— chemical engineering;

— computer information technology;

— computer engineering;

— computer science;

— computer networks;

— cybersecurity;

— information systems;

— laboratory management;

— software development and security; or

— software engineering.

6.2.3 Speciality topics

ISO/IEC 19790:2012 and the test requirements in ISO/IEC 24759 address the following specific speciality knowledge topics. A tester shall, at a minimum, demonstrate knowledge in at least one specific speciality topic.

A testing laboratory shall have knowledge in all the speciality areas as an aggregate of its technical staff.

ISO/IEC 19790:2012 and ISO/IEC 24759 specify speciality topics:

a) software and firmware development:

   1) programming languages (e.g. assembler and high-level);

   2) compilers;

   3) debugging tools;

   4) product testing performed by vendor:

      i) unit testing;

      ii) integration testing;

      iii) regression testing;

b) operating systems:

   1) installation;

   2) configuration;

   3) operation;

   4) architecture;

   5) system hardening;

   6) virtual machines;

7) java runtime environment;

c) hardware development:

1) hardware embodiments:

i) single-chip;

ii) multi-chip embedded;

iii) multi-chip standalone;

2) technology:

i) single-chip fabrication;

ii) electrical components and design, schematics and concepts including logic design and HDL representations;

iii) mechanical design and packaging;

3) manufacturing:

i) supply chain integrity;

ii) fabrication methods;

iii) initialization of parameters;

iv) packing and shipping;

v) testing and characterization;

4) hardware security features;

d) operational environments:

1) boot loader;

2) loading;

3) linking;

4) memory management and protection;

5) inter-process communication;

6) discretionary access control;

7) role-based access control;

8) executable forms;

9) audit mechanisms;

e) cryptographic algorithms, mechanisms and techniques:

1) cryptographic algorithms and security functions:

i) symmetric key;

ii) asymmetric key;

iii) hashing;

iv) random bit generators;

v) message authentication;

vi) entropy;

vii) modes of operation;

2) sensitive security parameter management:

i) sensitive security parameter generation;

ii) sensitive security parameter establishment;

I) automated SSP transport or SSP agreement;

II) manual SSP entry or output via direct or electronic;

iii) sensitive security parameter entry and output;

iv) sensitive security parameter storage;

v) sensitive security parameter zeroization;

f) identification and authentication mechanisms:

1) identity-based authentication;

2) role-based authentication;

3) multi-factor–based authentication;

g) best practices in design and development:

1) design assurance such as configuration management, delivery, operation and development;

2) design by contract;

h) informal modelling;

1) finite state model;

i) non-invasive security;

1) non-invasive attacks:

i) DPA/DEMA;

ii) SPA/SEMA;

iii) timing attacks;

2) countermeasures:

i) physical countermeasures;

EXAMPLE 1 Precharge logic, dual-rail logic, current flattening, probe detection, adding noise, random interrupts, jittered clock.

ii) Logical countermeasures;

EXAMPLE 2 Masking, hiding, dummy operation, balanced timing, shuffling, automatic re-keying.

j) self-test mechanisms:

1) pre-operational tests;

2) conditional tests;

k) security mechanisms:

1) zeroization;

2) trusted path;

3) tamper evident devices;

4) epoxies, potting materials and adhesives (including chemical properties);

5) encapsulation enclosures and materials;

6) tamper mechanisms;

7) countermeasures against fault induction attacks;

EXAMPLE 3 Redundancy-based scheme, error detecting code, footprint

8) secure communication protocols (e.g. Secure Sockets Layer, Transport Layer Security, Internet Key Exchange, Secure Socket Shell, Over the Air Rekeying, etc.);

9) security policy attributes;

10) split knowledge procedures;

l) design features:

1) ports and interfaces;

2) approved modes of operation;

3) specification of services;

4) specification of sensitive security parameters;

m) tools and test methods:

1) construction of test jigs (software or hardware);

2) environmental testing methods such as the use of temperature (e.g. heat and cold) and voltage (e.g. changes to input power);

i) temperature chambers (e.g. heating and cooling mechanisms);

ii) variable power supplies;

3) use of hand tools (e.g. saws, drills, prying tools, grinding, variable speed rotary tools, dental picks and mirrors, etc.);

4) use of chemical solvents (e.g. acids and alkaline based);

5) artificial light sources;

6) magnification tools;

7) use of digital storage oscilloscopes or logic analysers;

8) use of volt-ohm-meter or digital multi-meter;

9) digital scanner;

10) digital camera (including near or MACRO focus capabilities);

11) validation program supplied tools.

NOTE Calibration of tools are only required depending on the test method.

Additional information on specific knowledge association with the security of cryptographic modules is specified in Annex C.

6.3 Knowledge of standards

6.3.1 General

The tester shall have knowledge of the normative references specified in Clause 2. The tester shall be able to demonstrate an understanding or familiarity of one or more of the following topics.

6.3.2 ISO/IEC 19790 concepts

The tester shall have knowledge of the concepts in ISO/IEC 19790. ISO/IEC 19790specifies the security requirements for a cryptographic module utilized within a security system protecting sensitive information in computer and telecommunication systems. ISO/IEC 19790 defines four security levels for each of 11 requirement areas with each security level increasing security over the preceding level for cryptographic modules.

6.3.3 ISO/IEC 24759

6.3.3.1 General

ISO/IEC 24759 specifies the test requirements for cryptographic modules to be used by vendors and testing laboratories. ISO/IEC 24759:2017 includes 11 sub-clauses corresponding to the 11 areas of security requirements and six sub-clauses corresponding to ISO/IEC 19790:2012, Annexes A to F. These corresponding security requirements are listed in ISO/IEC 19790:2012, 5.2.2.5 and 5.2.2.6, respectively.

6.3.3.2 Vendor requirements

ISO/IEC 24759 specifies all of the vendor evidence (VE) requirements that vendors provide to testing laboratories, that are applicable to the module under test, as supporting evidence to demonstrate their cryptographic module's conformity to the security requirements specified in ISO/IEC 19790:2012.

The vendor shall also satisfy any modifications, additions, or deletions to the VE evidence that the validation authority has made to ISO/IEC 24759.

The tester shall be familiar with all vendor requirements.

6.3.3.3 Test requirements

ISO/IEC 24759 specifies the tester evidence (TE) requirements, applicable to the module under test, to be used by testing laboratories to test whether the cryptographic module conforms to the requirements specified in ISO/IEC 19790. The methods are developed to provide a high degree of objectivity during the testing process and to ensure consistency across the testing laboratories.

The tester shall also satisfy any modifications, additions, or deletions to the TE evidence that the validation authority has made to ISO/IEC 24759.

The vendor shall be familiar with all test requirements.

6.3.4 Additional ISO/IEC standards

The tester shall be familiar with the following.

— ISO/IEC 17825 specifies the testing methods for the mitigation of non-invasive attack classes against cryptographic modules.

— ISO/IEC 18367 specifies cryptographic algorithms and security mechanisms conformance testing.

— ISO/IEC 20085-1 specifies test tool requirements for use in testing non-invasive attack mitigation techniques in cryptographic modules.

— ISO/IEC 20085-2 specifies test tool calibration methods for use in testing non-invasive attack mitigation techniques in cryptographic modules.

— ISO/IEC 20543 specifies test and analysis methods for random bit generators within ISO/IEC 19790 and ISO/IEC 15408.

6.4 Knowledge of the validation program

6.4.1 Validation program

6.4.1.1 General

Validation programs, which typically operate under the auspices of an accreditation authority, often define aspects of their operation that are specific to the program. This is usually based on applicable legislation and policies, such as national policies, that are applicable to their operation. Testers shall have knowledge of the validation program and any specific aspects such as those listed in 6.4.1.2 to 6.4.1.7.

6.4.1.2 Organization

This aspect concerns the program's organization, and the bodies that are involved in the program's operation.

6.4.1.3 Communications

This aspect concerns how the program communicates relevant information to stakeholders, especially to testing facilities and the associated testers. This should include how communications and information is protected.

6.4.1.4 Legal and regulatory mandates

This aspect concerns the legislative and/or regulatory framework under which the validation program operates.

6.4.1.5 Policies

This aspect concerns specific policies that are applicable to the validation program. These can include process and technical requirement related policies in connection to accepting cryptographic module validation projects. The following are some examples.

a) Testing sufficiency: the tester should have knowledge of what is required in ensuring that a target cryptographic module is tested sufficiently.

b) Disposition of evidence: the process for properly disposing of supporting evidence upon completion of a project.

c) Confidentiality: any requirements for confidentiality (on the part of the tester and the non-disclosure of information obtained during testing projects).

d) Problem resolution: the course of action to be taken if a problem is encountered during the project (whether the work continues once the problem is remedied, or the project ends immediately and the remedied product needs to be re-submitted).

e) Language: any specific (natural) language in which documentation needs to be provided.

f) Requirements for recorded evidence: any recorded evidence documented by the tester that needs to be submitted to the validation program.

g) Additional reporting policies: any specific reports required from the tester such as testing reports.

h) Implementation guidance: a validation authority can provide programmatic or clarification guidance that should be considered by the tester.

i) Reuse: documentation and rationale required by the validation program to support the reuse of testing evidence.

j) Any specific handling of the validation program identifiers, logos, trademarks, etc.

k) Handling and application of validation program interpretations.

l) A list or characterizations of suitable alternative approaches to testing when the validation program's recommended original testing is infeasible for a given target cryptographic module.

m) The policies by which the validation program determines what steps a tester took while testing.

6.4.1.6 Documentation

This aspect concerns the provision and use of any validation program specific documents. These can include forms, templates, training material, and informational material. Validation program specific documents can include documents such as:

— management manuals;

— frequently asked questions;

— implementation or programmatic guidance;

— manuals for program supplied tools.

6.4.1.7 Tools

The validation program can provide specific tools for testing, report generation, delivery or protection (i.e. encryption). Examples include:

— algorithmic test tools;

— generation of test vectors and resultant expected responses;

— documentation of testing activities and reporting;

— encryption tool for protection of test reports transmitted to the validation program;

— specification of particular encryption algorithm and signature methods (e.g. 128-bit AES for encryption and 2048-bit RSA with SHA-2 for digital signature).

6.5 Knowledge of the requirements of ISO/IEC 17025

Since testing facilities are often required to be compliant with ISO/IEC 17025, the tester shall be familiar with the requirements of ISO/IEC 17025 and how these are implemented in the validation facility or facilities with which the tester is associated. If there are additional programmatic accreditation documents associated with ISO/IEC 17025, that form the basis of the laboratories accreditation, then the tester shall be familiar these documents as well.

7 Skills

7.1 General

Training for testers is often obtained through career experience in the IT industry, or during their association with a testing facility, or because of the requirements of professional organizations.

EXAMPLE Professional certifications such as the ISC2™' CISSP™ credential are associated with a requirement for continued professional development.

7.2 Algorithm testing

The tester shall have the ability to install, configure and execute the cryptographic algorithm validation program or user interface driven algorithm test tools.

7.3 Physical security testing

The tester shall have the skills to perform the physical security tests which they are appropriately trained for and skilled at.

7.4 Side channel analysis

The tester shall have the skills to perform the side channel tests which they are appropriately trained for and skilled at.

7.5 Technology types

The skills and techniques required in the cryptographic module testing of different technology types can vary. Testers shall be able to demonstrate that they have the necessary knowledge, skills and techniques related to the technology types of cryptographic modules which they test.

NOTE 1 The validation program addresses cryptographic modules that represent many technology types which are being considered for testing. A list of the technology types most commonly referenced and suggested fundamental knowledge skills and techniques that testers need is given in Annex B.

NOTE 2 Many specialist professional certifications cover the body of knowledge that is needed by testers. Such certifications can be national, regional or global in scope. It is beyond the scope of this document to catalogue all of them, however some of these are listed in the Bibliography.

8 Experience

8.1 General

The tester shall document their training and testing activities in accordance with validation program and or testing facility requirements.

8.2 Demonstration of technical competence to the validation program

8.2.1 Experience with performing testing

The tester shall record all testing activity in a log. The records shall be kept along with the acquired test evidence (see Annex A). The tester should create the testers' log for operational testing so that it clarifies both expected test results and actual test results.

EXAMPLE Expected status indicator.

8.2.2 Experience with particular technology types

The tester shall include in their recorded activity log the technology types that were tested.

9 Education

The tester shall document their education in accordance with validation program and or testing facility requirements.

Education requirements are referenced in 6.2.

10 Effectiveness

The tester shall be able to apply knowledge and skills in a productive manner, characterized by attributes of behaviour such as aptitude, initiative, enthusiasm, willingness, communication skills, team participation, and leadership.

Annex A
(informative)

Example of an ISO/IEC 24759 testers' log

Name		
Designation		
Validation program		Testing facility
Cryptographic module name		Cryptographic module type
Overall Security level		Cert ID (if known)
Sponsor/developer		Dates testing performed
Description of IUT		
AS 01.01	Applicable test requirement as specified in ISO/IEC 24759	
	Description of IUTs design for conformance to the test requirement	
	Description of test method and results	

Annex B
(informative)

Ontology of technology types and associated bodies of knowledge

B.1 General

The validation program addresses cryptographic modules that represent many technology types which are being considered for testing. A list of the technology types most commonly referenced and suggested fundamental knowledge skills and techniques that testers need is presented below.

B.2 Technology types

B.2.1 General

Cryptographic modules can be software, firmware, hardware or a hybrid of software and firmware with hardware.

B.2.2 Software/firmware

Software or firmware can be written in various programming languages and then compiled into different forms of executables. An executable can represent a cryptographic software or firmware module. Debuggers may be used for finding and correcting errors in the implementation.

B.2.2.1 Programming languages

Examples of various software programming languages that may be employed are (This list is neither exhaustive nor complete and for illustrative purposes only):

— Ada;

— APL;

— Assembly language;

— C++;

— dBase ;

— Google Apps Script ;

— Java ;

— JavaScript ;

— Microcode;

— Unix shell;

— Visual Basic;

— VHDL.

B.2.2.2 Compilers

Examples of various open source software compilers that may be employed are (This list is neither exhaustive nor complete and for illustrative purposes only):

— FreeBASIC;

— Clang C/C++/Objective-C Compiler;

— Free Pascal;

— GCC [C, C++, (G++), Java (GCJ) and Ada (GNAT)];

— Local C compiler;

— Open Watcom;

— Open64;

— XPL PL/I;

— C to HDL.

B.2.2.3 Debuggers or Simulators

Examples of various open source debuggers that may be employed are (This list is neither exhaustive nor complete and for illustrative purposes only):

— Firefox JavaScript debugger;

— GDB – the GNU debugger;

— Eclipse debugger;

— Opera Dragonfly;

— Python debugger;

— X64dbg;

— ZeroBUGS;

— VHDL;

— Verilog.

B.2.2.4 Hardware

B.2.2.4.1 General knowledge

Hardware may be implemented in various embodiments and technology types. Below are examples of hardware embodiments and technology types within each. The following lists are neither exhaustive nor complete and are for illustrative purposes only.

B.2.2.4.2 Single-chip modules

B.2.2.4.2.1 General knowledge about single-chip modules

A single-chip cryptographic module is a physical embodiment in which a single integrated circuit (IC) chip can be used as a standalone device or embedded within an enclosure or a product that may not be physically protected.

B.2.2.4.2.2 Single-chip substrate materials

Examples of single-chip substrate materials:

— Gallium arsenide;

— Germanium;

— Monocrystalline silicon.

B.2.2.4.2.3 Single-chip packaging types

Examples of single-chip packaging types:

— Dual in-line package (DIP);

— Pin grid array (PGA);

— Leadless chip carrier (LCC);

— Surface mount;

— Thin small-outline package (TSOP);

— Plastic quad flat pack (PQFP);

— Ball grid array (BGA);

— Flip-chip ball grid array (FCBGA).

B.2.2.4.3 Multi-chip embedded modules

A multiple-chip embedded cryptographic module is a physical embodiment in which two or more integrated circuit chips are interconnected and are embedded within an enclosure or a product that may not be physically protected.

EXAMPLE

— Adapters cards;

— Expansion boards;

— Daughter cards.

B.2.2.4.4 Multi-chip standalone modules

A multi-chip standalone cryptographic module is a physical embodiment in which two or more integrated circuit chips are interconnected and the entire enclosure is physically protected.

EXAMPLE

— Encrypting routers and switches;

— Secure radios;

— USB tokens.

Annex C
(informative)

Specific knowledge associated with the security of cryptographic modules

C.1 General

The security requirements [xx.yy] which appear in this document are specified in ISO/IEC 19790:2012. Each security requirement in ISO/IEC 19790:2012 includes a referencing index [xx.yy] where xx indicates the clause and yy is a numeric index within the clause.

C.2 Cryptographic module specification

C.2.1 General

Figure C.1 illustrates an overview of the cryptographic module specification as related to the other elements consisting of the cryptographic boundary, block diagram, major components of the cryptographic module, interconnections and the types of date passing through the module.

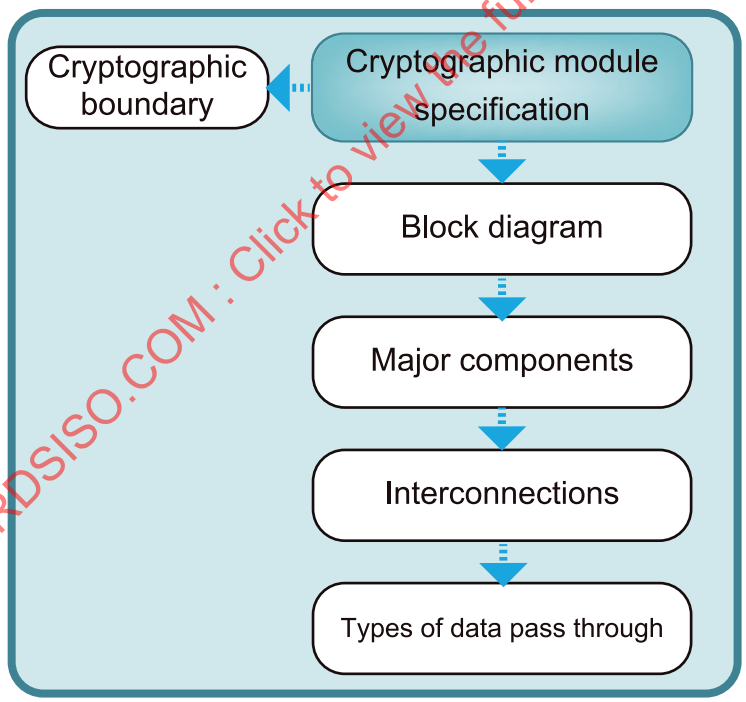

Figure C.1 — Overview of cryptographic module specification

C.2.2 Buffers

There can be data buffers and/or SSP buffers within components (especially for multi-chip embedded module or for multi-chip standalone module). Also, there are multiple levels of data cache in CPU. In such cases, it is a good practice for testers to identify buffers from publicly available information and information provided by the vendor. The information will be the input for cryptographic module testing of zeroization mechanism.

C.2.3 Security relevant components

By reviewing the block diagram, the testers understand the information flow and types of information transferred through each component. By identifying components which transfer/store/read/write/generate/use/zeroise SSPs, the testers can identify security relevant components.

C.2.4 Identification of programmable interfaces, debugging interfaces and covert channels

Commercial grade components can have programmable interfaces, debugging interfaces, and/or covert channels. It is a good practice for testers to search publicly available information and information provided by the vendor, to determine whether such interfaces exist. If there is such an interface, testers verify from the cryptographic module testing that such an interface is not available due to the cryptographic module design.

In the case of software module, especially for dynamic link libraries or shared objects, there are known tools to identify exported APIs. By using such tools, the tester can determine whether undocumented APIs are exported.

If the cryptographic module parses input commands, the tester can determine whether undocumented input commands are supported by inspecting the parser implementation(s). Note that the parser can be implemented in multiple layers.

C.2.5 Identification of approved and non-approved security functions

The testers should identify approved security functions and also non-approved security functions. For an implementation of security function to be considered as approved, the following is required.

a) The implementation of security function has passed a cryptographic algorithm testing.

b) The implementation of security function meets the requirements of conditional cryptographic algorithm self-test.

c) If a SSP is used, then the SSP can be zeroized when operating in approved mode of operation, and

   1) the SSP is generated by an approved SSP generation method by an operator assuming an authorized role when operating in an approved mode of operation, or

   2) the SSP is established by an approved SSP establishment method by an operator assuming an authorized role when operating in an approved mode of operation.

d) If a random number is used, then the random number is generated by an approved RBG when operating in an approved mode operation, unless otherwise specified.

e) If an approved RBG is used, then the entropy security requirements [09.07] and [09.08] are met.

If one or more of the conditions described above are not met, then the implementation of security function is considered non-approved.

C.2.6 Exclusion of components

To verify the justification of exclusion of components, the tester needs the following information:

a) which types of data (e.g. sensitive security parameters) pass through the identified components of the defined boundary of the module;

b) interconnections between components and interfaces;

c) whether there is a programmable interface or a debugging interface that can provide a non-secure channel for the access of sensitive security parameters.

C.2.7 Degraded operation

A cryptographic module can be designed to support degraded functionality if the module enters the error state. Vendors can implement degraded functionality for cryptographic modules to achieve fault tolerance or to maintain survivability. A cryptographic module can inform other cooperating cryptographic modules via status output that the cryptographic module enters a degraded operation.

Note that degraded operation originates from an error detected in a conditional self-test. As stated in the requirement [02.32], it is not allowed to enter degraded operation if the cryptographic module fails the pre-operational self-tests. To exit degraded operation and enter normal operation, the cryptographic module passes without failure all pre-operational self-tests successfully, as stated in the requirement [02.31].

As the contents of pre-operational self-tests are different from those of conditional self-tests, it is not guaranteed for the cryptographic module to pass the conditional self-tests if the module passes the pre-operational self-tests. Figure C.2 shows an example of state transitions for cryptographic module supporting degraded functionality. In Figure C.2, the conditional algorithm self-test for RSA is shown as an example where an error is detected.

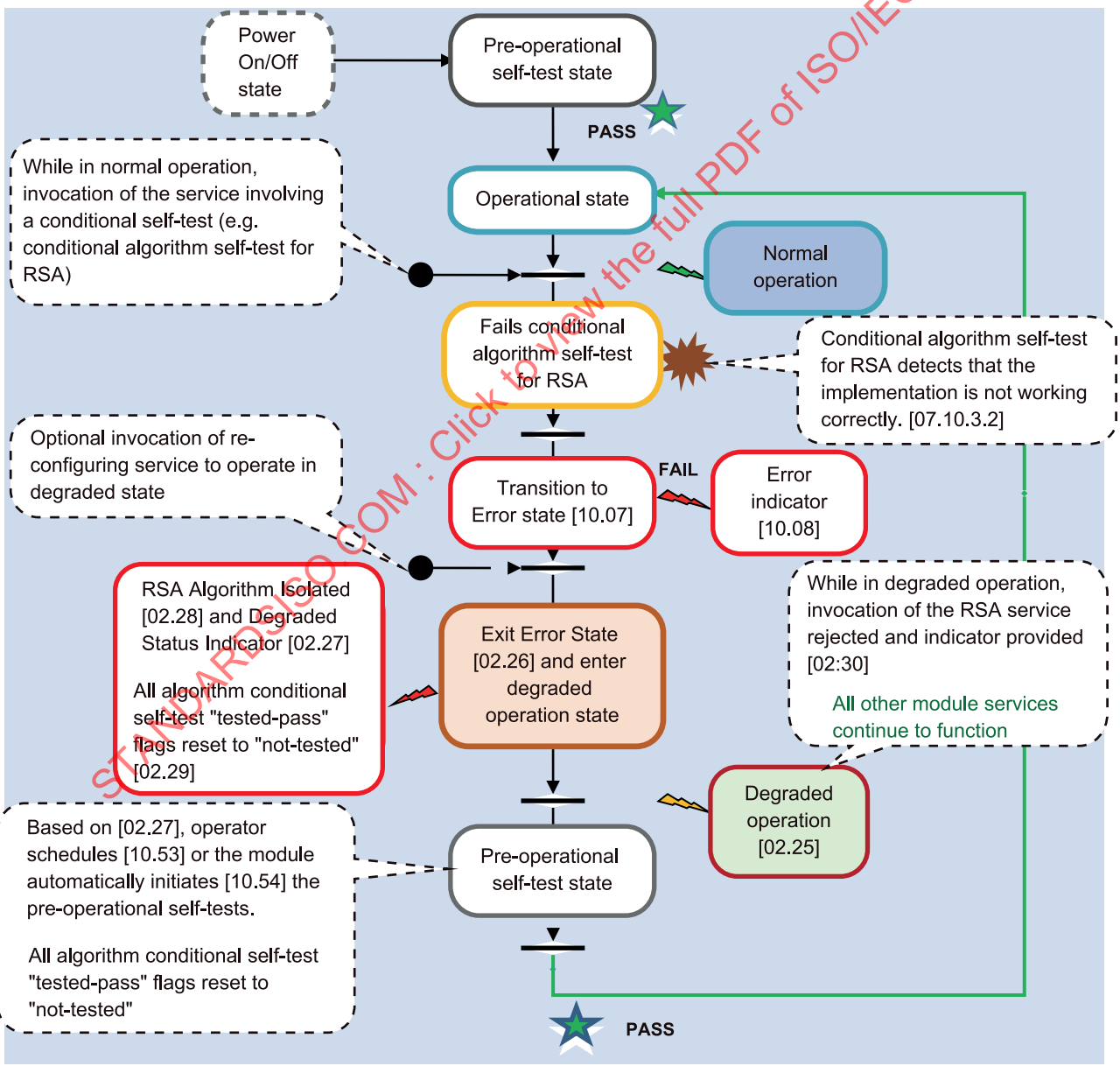

Figure C.2 — Example of state transitions supporting degraded functionality

If an error detected in a conditional self-test is caused by a permanent fault, it will again be found by the conditional test and the cryptographic module will repeat degraded operation. A cryptographic module can be designed to permanently transition to the error state if degraded mode repeated too many times.

C.3 Cryptographic module interfaces

C.3.1 Overview

In ISO/IEC 19790, it is required that software, firmware and hardware modules have a defined cryptographic boundary as part of the requested service and have software, firmware and hardware module interfaces used to request the services of the module, including parameters that enter or leave it. Figure C.3 illustrates the various types of functional interfaces that a module is required to have and controls if applicable.

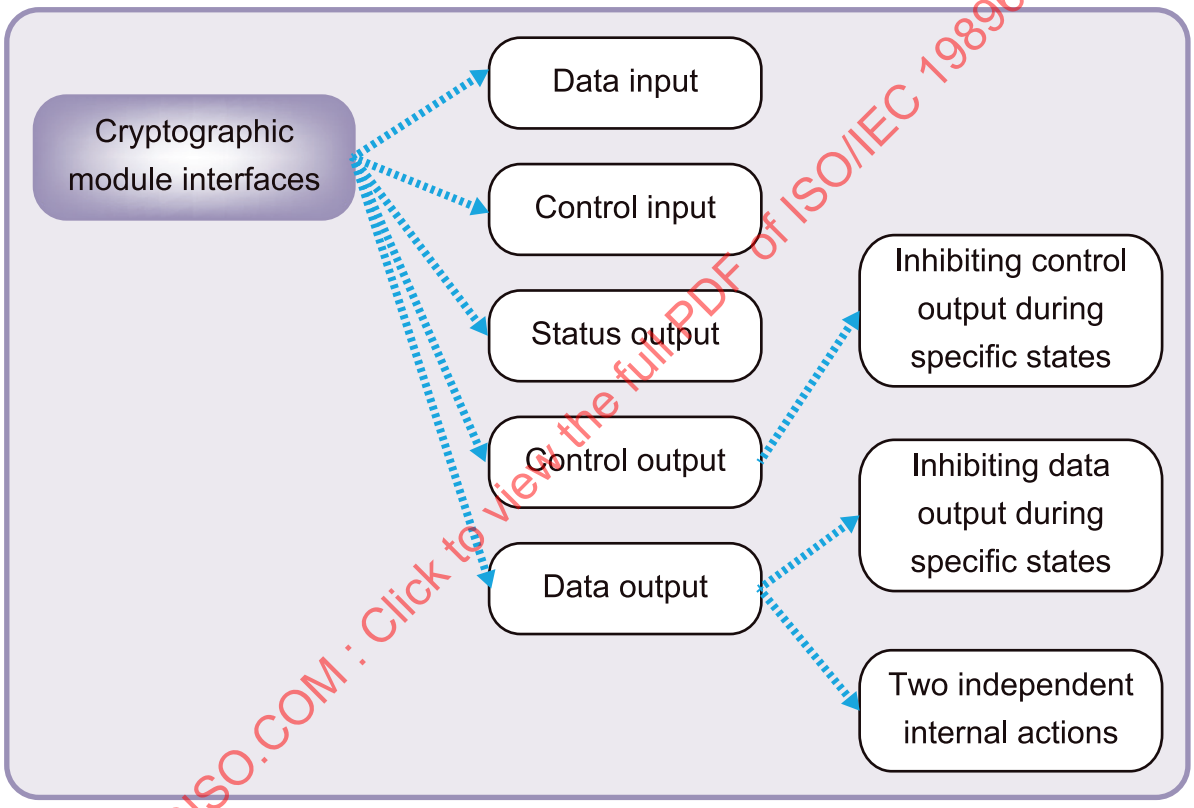

Figure C.3 — Overview of cryptographic module interfaces

C.3.2 Separation of input data from output data

In considering application programming interfaces (APIs), the parameters are divided into a) input parameters, b) output parameters, or c) input/output parameters. The return value of API is considered as output data or output control data. Input/output parameters can be further divided into either input parameters or output parameters in combination with the information of the other parameters and/or return values.

C.3.3 Knowledge of critical security functions, services or security relevant services

In ISO/IEC 19790:2012, 7.2, it is required that the output data path is logically disconnected from the circuitry and processes while performing key generation, manual key entry, or key zeroization. This means that these three services are considered security relevant. Due to a single error or misuse, intermediate key values, manually entered key, or key values during zeroization can be inadvertently output through the output data path.

C.3.4 Trusted channel

A trusted channel provides a safe and secure communication link between the cryptographic module and a sender or receiver to communicate unprotected data. A trusted channel protects against eavesdropping, as well as physical or logical tampering by unwanted operators/entities, processes or other devices, between the module's defined input or output ports and along the communication link with the intended endpoint. Knowledge of the mechanisms that are used to establish and employ the trusted channel is critical to the protection of unprotected data.

C.4 Roles, services, and authentication

C.4.1 General

Figure C.4 illustrates an overview and relationships between roles, services and authentication mechanisms and how an access control policy intersects with the modules pre-defined services.

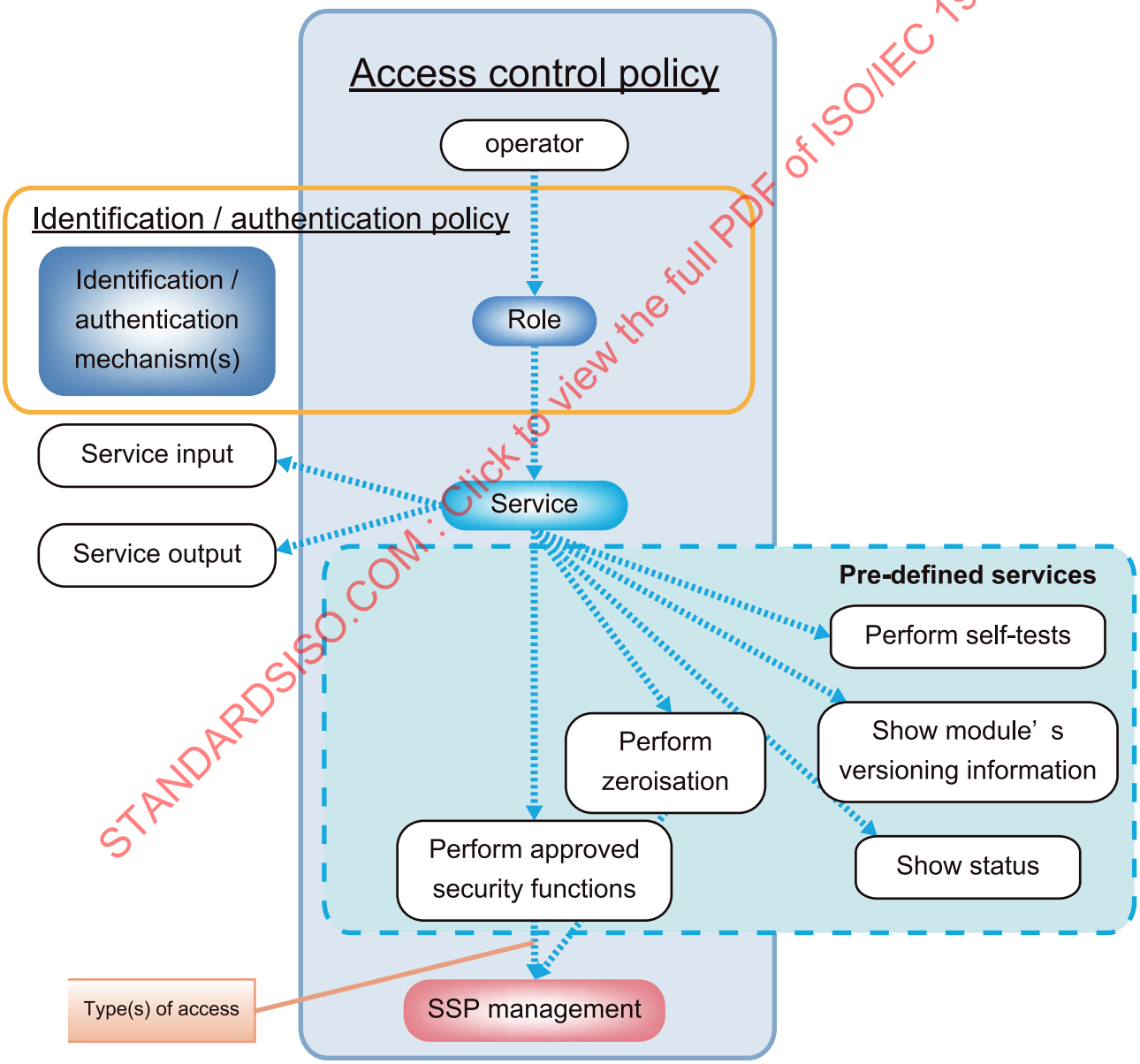

Figure C.4 — Overview of roles, services, and authentication

C.4.2 Services

In ISO/IEC 19790:2012, B.2.4 , the access control policy is required to be sufficiently detailed so as to answer the following question:

— What access does operator X, performing service Y while in role of Z, have to security relevant data item W for every role, service, and security-relevant data item contained in the cryptographic module?

In ISO/IEC 19790:2012, B.2.4 , the following is required:

— For each service, the service name, a concise description of the service purpose and/or use (the service name alone can, in some instances, provide this information), a list of approved security functions (algorithm(s), key management technique(s) or authentication technique(s) used by, or implemented through, the invocation of the service, and a list of the SSPs associated with the service or with the approved security function(s) it uses.

For each operator role authorized to use the service, information describing the individual access rights to all SSPs and information describing the method used to authenticate each role.

Figure C.5 depicts the above required information.

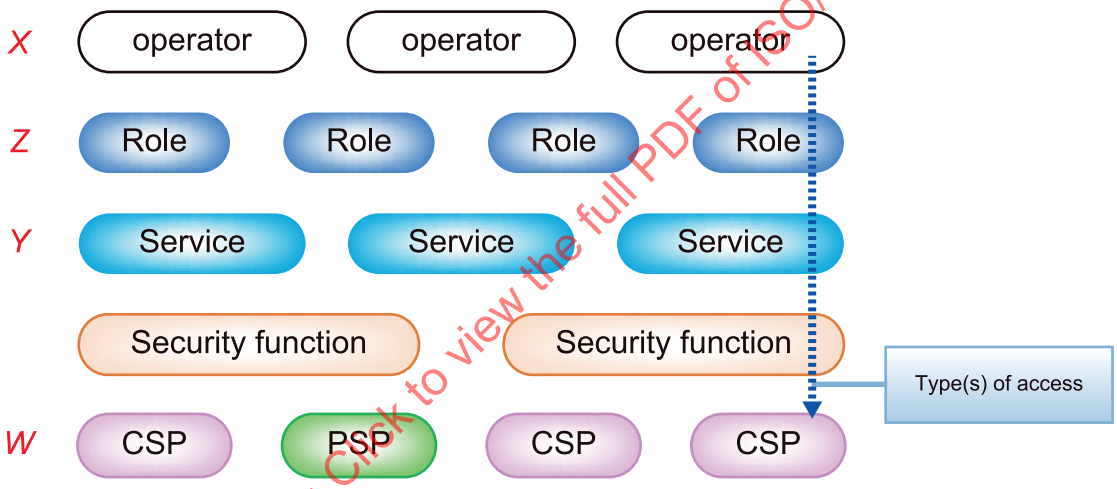

Figure C.5 — Overview of access control policy

C.4.3 Authentication

When the replacement of default authentication data of a cryptographic module is replaced at security level 3 or 4, new operator-specific authentication data needs to be input to the cryptographic module as per the requirements [04.45] and [09.20].

When plaintext authentication data is input to the cryptographic module, an identity-based authentication is required in advance, as per the requirement [03.20].

When encrypted authentication data is input to the cryptographic module, the authentication data shall be encrypted as per the requirement [09.13]. Note that encryptions using global secret keys are not considered “approved”. One can think to generate a secret cryptographic key for the trusted channel, but it is possible that it not be accepted as an exception by a validation authority's specific guidance.

A validation authority may provide guidance when it is assumed that the operator authentication can be potentially attempted multiple times without assuming any authorized roles. If a CSP is used in the operator authentication, there is a risk that the CSP can be released or disclosed, by using fault induction attacks or by using non-invasive attacks. The security of CSPs used for operator authentication can be achieved if they are separated from CSPs used for the other purposes.

C.5 Software/firmware security

Each operating system defines the executable forms of software. Current executable forms have header segments which are parsed in the loading stage. If such content of headers is modified, the operating system can stop to load the software. Executable forms can also have data segments. It is possible that the operating system does not stop to load the software if the content of data segments is modified. If the software/firmware integrity test is implemented by the module itself, then modifying a header segment is not sufficient to test the software/firmware integrity test.

C.6 Operational environment

C.6.1 Process memory management

The requirement [06.06] depends on the process memory management provided by operating systems. The memory protection provided by operating systems is applied to the process level so the thread-level protection is not applied in the commercially available operating systems.

C.6.2 Loading

In considering the software module, an operating system loads the software module and then the software module starts its operation from the entry point. In the case of a shared object or dynamic link library, some operating systems provide options to bypass the entry point of the software module. If the entry point enforces to perform pre-operational software integrity test, then the tester should verify that the guidance documentation specifies not using such options.

C.6.3 Linking

If two or more symbols make a collision, then the final decision is up to the linker. The guidance documentation should address such aspect.

There are software modules which consist of multiple executable files. Currently there are tools to analyse the dependency of executable files. After understanding the dependency of the executable files, the tester should verify that the integrity of all software components is tested in the pre-operational software integrity test.

C.6.4 Virtual memory

Current operating systems support virtual memory, which uses HDDs (or SSDs) as if they are physical memory. For security reasons, if the cryptographic process is allowed to reside on the virtual memory, and the virtual memory is stored on a network drive, then there is a risk that the process is accessed by other operating system. The tester should verify that the guidance documentation does not allow such usage.

C.7 Physical security

A cryptographic module shall [07.01] employ physical security mechanisms in order to restrict unauthorized physical access to the contents of the module and to deter unauthorized use or modification of the module (including substitution of the entire module) when installed. All hardware, software, firmware, data components and SSPs within the cryptographic boundary shall [07.02] be protected.

C.8 Non-invasive security

Non-invasive attacks attempt to compromise a cryptographic module by acquiring knowledge of the module's CSPs without physically modifying or invading the module. Modules may implement various techniques to mitigate against these types of attacks. The test metrics for non-invasive attack mitigation for each of the associated security functions addressed by ISO/IEC 19790:2012, Annex F.

C.9 Sensitive security parameter management

C.9.1 General

A cryptographic module employs many parameters in the execution and performance of its services. These paraments may or may not require security management. Sensitive security parameters need to be managed properly to ensure the security of the paraments. Figure C.6 illustrates the relationship between different security parameters relative to the security function performed.

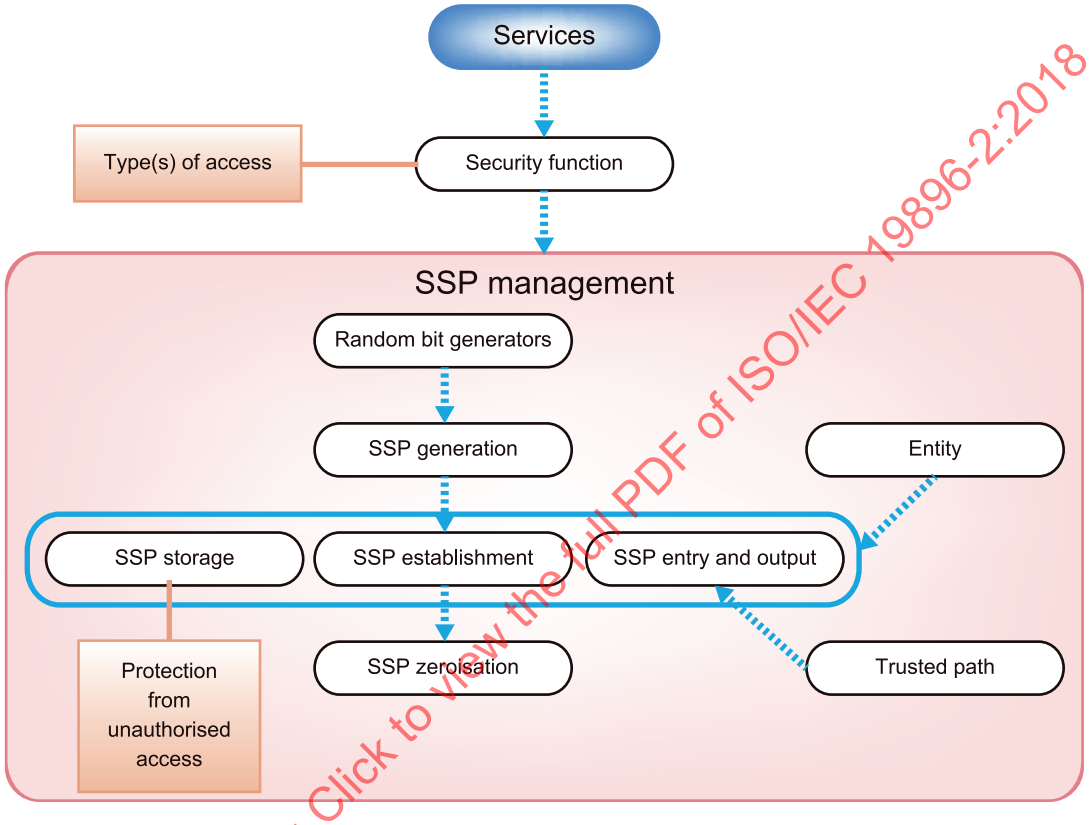

Figure C.6 — Overview of SSP management

C.9.2 Password vs cryptographic key

Generally speaking, information entropy of passwords is much less than that of cryptographic keys. Therefore, not any password can be used as cryptographic key under ISO/IEC 19790, unless explicitly stated in the selected algorithm standard of approved security function.

EXAMPLE NIST/SP 800-132, *Recommendation for Password-Based Key Derivation*.

C.9.3 Entropy vs attackers' knowledge

Under ISO/IEC 19790, encryption using any global secrets is not considered as cryptographically protected. Likewise, if the seed of DRBG algorithm is input at the factory, the manufacturer has the knowledge of the seed and then has the capability to predict the output from the DRBG, in the worst case. It should be understood that the more information attackers obtain, the less entropy remains.

In the case of software module, vendors can claim non-zero entropy for a piece of information (e.g. process ID) which can be available to the other processes running on the operating system. If an attacker has access to the operating system, or control of the other processes, then the piece of information does not add any entropy.

A physical RBG is implemented inside a smart card, and the random numbers from the RBG can be a) used inside the card, and/or b) output from the card (e.g. as a challenge). In the former case, the secrecy is kept when there is no untrusted entity inside the card, and there is no need to consider non-invasive attacks. In the latter case, the secrecy is not kept once the random number is output.

C.9.4 SSP hierarchy

C.9.4.1 General

As stated in requirements [09.03] and [09.25], the cryptographic module shall associate every SSP within the module with an entity. The entity can be a person, group, role, or process.

C.9.4.2 Split knowledge

In ISO/IEC 19790, the term "split knowledge" is defined as a process by which a cryptographic key is split into multiple key components, individually sharing no knowledge of the original key, which can be subsequently input into, or output from, a cryptographic module by separate entities and combined to recreate the original cryptographic key.

EXAMPLE It is not considered as split knowledge to just divide a 128-bit key into former 64-bit and later 64-bit.

As stated in requirement [09.22], for entering or outputting each key component, a different operator shall be involved. For example, operator A is associated with key component 1 and operator B is associated with key component 2. In ISO/IEC 19790, it is not mentioned which entity (or operator) is capable of reconstructing the original key. Based on the design of cryptographic module, the service reconstructing the original key can be assigned to either operator A or B, or another entity or another specific role.

C.9.5 Authorized roles for SSPs management

An authorized role is required before utilizing approved security functions if cryptographic keys and CSPs are created, modified, disclosed, or substituted.

As stated in the requirement [09.20], the trusted channel is required when inputting CSPs and authentication data to or outputting CSPs and authentication data from the cryptographic module for security levels 3 and 4. As stated in requirement [03.20], the identity-based authentication is required when performing services utilizing the trusted channel.

These requirements might seem contradictory. To establish a trusted channel, the authentication data is entered by the operator. However, the inputting authentication data requires the trusted channel. Here, as an exception, approved security function can be used for processes used for authentication (e.g. symmetric algorithm secret sharing, asymmetric algorithms for authentication), before assuming an authorized role.

C.9.6 Zeroization

C.9.6.1 Copies of SSPs

Through the testing of cryptographic module specification, the tester understands the locations where SSPs and their copies reside, temporally and/or permanently. The locations can be key registers, CPU cache, RAM, buffers, or HDD. The tester should verify that SSPs are physically zeroized by inspection of the design of cryptographic modules.

C.9.6.2 Embodiment of storage device

C.9.6.2.1 Flash memory

Current flash memory devices employ “wear levelling” techniques. As a result of “wear levelling”, overwriting specific data does not always mean overwriting physically, i.e. new data value is written to a physical memory address other than the original physical memory address where the original data value is written. If the memory devices employ “wear levelling” technique, the tester should verify that SSPs are physically zeroized by inspection of the design of cryptographic modules.

EXAMPLE The “wear levelling” technique is disabled for specific physical memory address range storing SSPs.

C.9.6.2.2 Hard disk drive

The same holds for HDDs. In order to achieve physical overwriting, reformatting and overwriting should be performed at least once.

C.10 Self-tests

C.10.1 General

ISO/IEC 19790 specifies that a module implement an array of self-tests to insure the correct operation of the modules functions. The nature of cryptography is if an error occurs during a cryptographic operation, such as encryption, the encrypted result may not decrypt properly to recover the plaintext data. This would result in the loss of data. Other examples of the effect an error can cause is the wrong cryptographic keys used for a security service, authentication fails, or the module not respond correctly to an attack (e.g. zeroization of critical security parameters). Figure C.7 illustrates the critical functions employed by the module and the relationship to self-test requirements.

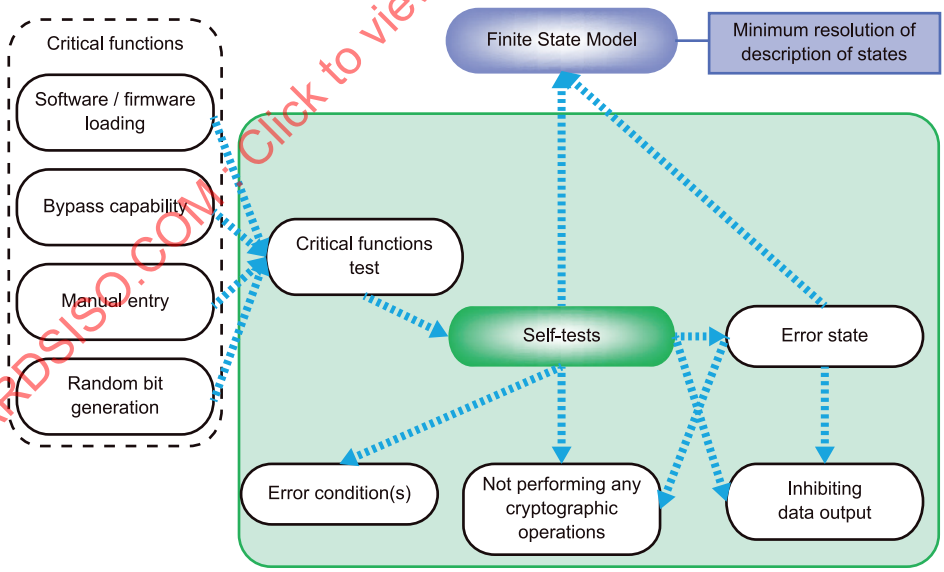

Figure C.7 — Overview of self-tests

C.10.2 Critical functions

C.10.2.1 Notion of critical functions

ISO/IEC 19790 refers to critical functions test. In considering the security of cryptographic module, the following items are considered security relevant:

a) integrity of software/firmware;

b) correctness of cryptographic algorithm implementation;

c) integrity of public and private key pair;

d) integrity of bypass mechanism(s);

e) authenticity of software/firmware to be loaded;

f) integrity of manually entered SSPs.

C.10.2.2 Pre-defined critical functions

In response to the items addressed in C.10.2.1, the following self-tests are addressed in ISO/IEC 19790:

a) pre-operational software/firmware integrity test;

b) conditional cryptographic algorithm self-test;

c) conditional pair-wise consistency test;

d) pre-operational bypass test;

e) conditional bypass test;

f) conditional software/firmware load test;

g) conditional manual entry test.

C.10.2.3 Vendor-defined critical functions

Vendors can claim other critical functions tests in addition to the self-tests already addressed in ISO/IEC 19790.

EXAMPLE 1 If a cryptographic module embeds a physical random bit generator, then the health of entropy source used inside the physical random bit generator will be critical to the secure operation of the cryptographic module.

EXAMPLE 2 As a cyclic redundancy check (CRC) is applied in the software/firmware integrity test, the integrity of CRC coprocessor is tested as a part of pre-operational critical function test.

EXAMPLE 3 As the integrity and availability of the memory content is considered critical, some cryptographic modules implement memory diagnosis test.

C.10.3 Pre-operational software/firmware integrity test

C.10.3.1 Scope of pre-operational software/firmware integrity test

An executable code is loaded to a volatile memory, via a system bus. Due to an environmental failure, fault induced attacks, or other reasons, there is the potential for a modification, substitution in the loaded executable code. The pre-operational software/firmware integrity test helps to protect against or mitigate such modification and/or substitution.

As stated in requirements [02.16] and [02.17], the instantiation of the cryptographic module saved in memory is included inside the cryptographic boundary. Here, all software components and firmware components within the cryptographic boundary are subject to pre-operational software/firmware integrity test, as stated in requirement [10.17]. From these requirements, not only the set of executable file or files but also the instantiation saved in memory are subject to pre-operational software/firmware integrity test.

C.10.3.2 Use of a truncated version of approved message authentication code

A vendor can choose to use a truncated version of approved message authentication code for the pre-operational software/firmware integrity test. In order to claim an approved integrity technique, it is not accepted to use a truncated version unless the truncated version is explicitly approved.

In relation to conditional cryptographic algorithm self-tests, a vendor can choose not to implement a separate cryptographic algorithm self-test for the underlying cryptographic algorithm used for the approved integrity technique if all the cryptographic functions of the underlying cryptographic algorithm are tested.

If a truncated version of approved message authenticated code is used and if the separate cryptographic algorithm self-tests are omitted for the approved message authenticated code and for its underlying cryptographic algorithms, then it can be said that the truncated part is never tested through self-tests. This can become a problem especially for hardware implementation of cryptographic algorithms.

C.10.3.3 Single encompassing message authentication code vs multiple disjoint codes

As stated in requirement [05.09], multiple disjoint message authentication codes or signature can be used in the integrity test. Design of the cryptographic modules internal structures affect whether a single encompassing authentication code or multiple disjoint authentication codes are employed. Examples for single encompassing message authentication code and for multiple disjoint codes are shown below.

EXAMPLE 1 The bootstrap loader and the main firmware providing services are combined into a monolithic file, and the monolithic file is stored in non-volatile memory in the specific memory address range. A single encompassing message authentication code is applied to the monolithic file. First, the bootstrap loader loads the main firmware, and next the main firmware performs the integrity test of the bootstrap loader and the main firmware, based on the file image and the process image. The expected message authentication code is stored in the specific memory address.

EXAMPLE 2 The bootstrap loader and the main firmware providing services are stored separately in different memory address ranges of non-volatile memory. A message authentication code is applied to the bootstrap loader, another message authentication code is applied to the main firmware, and the expected message authentication codes are stored in the specific memory addresses.

C.10.4 Conditional cryptographic algorithm self-tests

The testers should verify that cryptographic algorithm self-tests are properly implemented.

In considering the correct operation of a digital signature verification function, a valid signature should be verified as valid, and an invalid signature should be verified as invalid. A cryptographic algorithm self-test for digital signature verification includes both valid signature and invalid signature as test vectors.

Some cryptographic modules truncate message digests or MAC tags. If both truncated message digests and not truncated message digests are calculated by the cryptographic module, then the self-tests should check original (i.e. not truncated) message digests.

C.10.5 Pair-wise consistency test

In general, there are several conditions to be met by each asymmetric key pair.