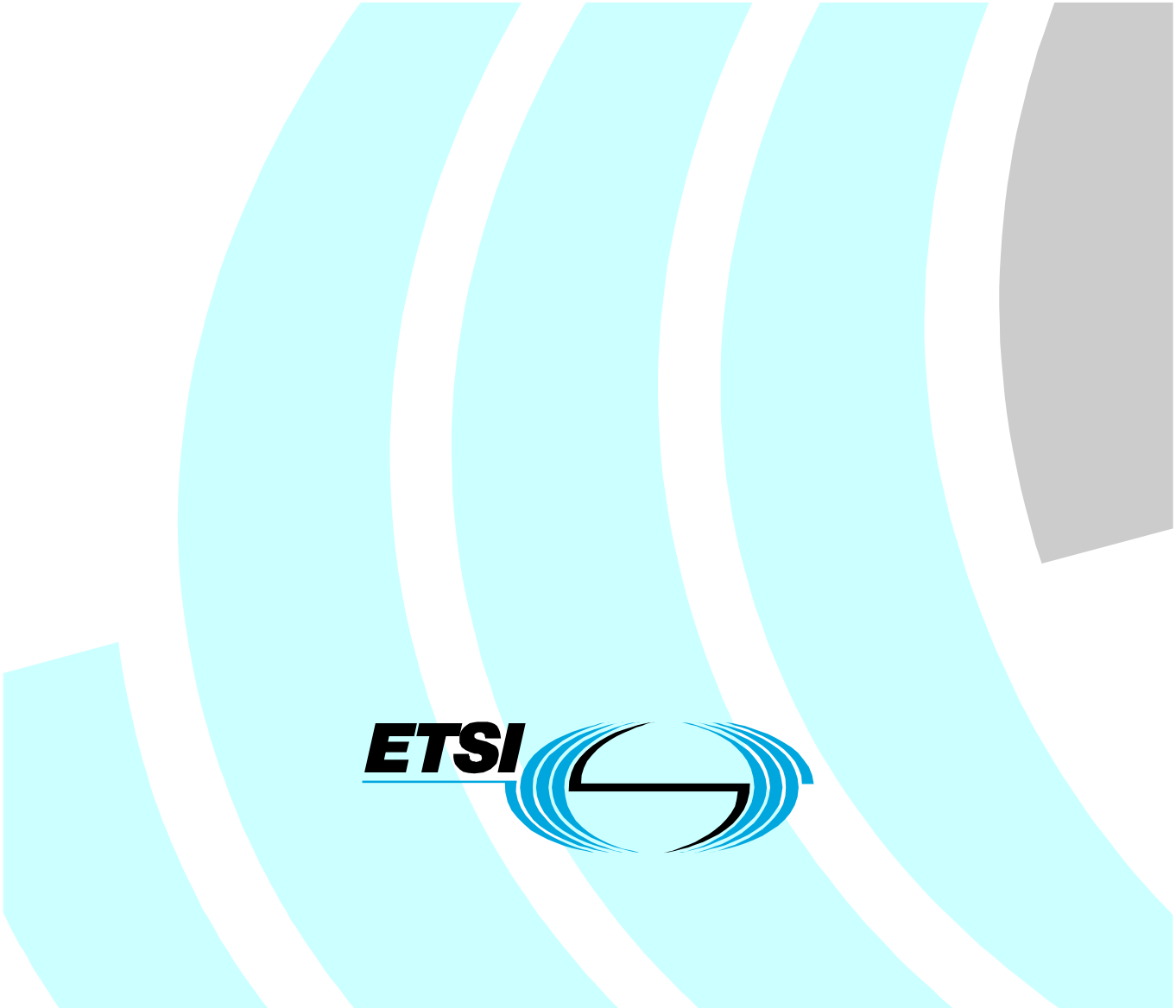
ETSI EG 202 387 V1.1.1 (2005-04)

ETSI Guide

Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Security Design Guide; Method for application of Common Criteria to ETSI deliverables



Reference
DEG/TISPAN-07005-Tech

Keywords
application, IP, methodology, security, VoIP

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Guide (EG) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN).

Introduction

The present document has been prepared with the sponsorship of the eEurope programme as part of the ETSI support to the eEurope action line for a secure information infrastructure (item 3: Society).

A major part of any security specification, and of a security product, is the measure of assurance it provides with respect to the security it offers.

Information security evaluation contributes to the users' trust and confidence in communications products and services. The use of common criteria for evaluation (as defined in ISO/IEC 15408 [20]) has facilitated mutual recognition of results in many European countries and these countries have also entered into an arrangement with the US and Canada for further mutual recognition of IT security certificates.

The present document is part of a set of standards and guidelines which show how the Common Criteria as identified in ISO/IEC 15408 [20] can be used effectively within the ETSI standardization process. The documents in this set are:

- EG 202 387: Method for application of Common Criteria to ETSI deliverables;
- ES 202 382 [2]: Method and proforma for defining Protection Profiles;
- ES 202 383 [3]: Method and proforma for defining Security Targets.

Between them, these documents identify how standards fit to the Common Criteria and how developers of standards should prepare their standards with a view to support submission for evaluation of product conforming to the standards.

Adoption of Common Criteria objectives in standardization of security countermeasures is also consistent with achieving the objectives and recommendations of the NIS report.

1 Scope

The present document is a guide to the development of standards that allow compliant product to be considered for product evaluation under the Common Criteria scheme [20].

NOTE: Within Europe there is mutual recognition of CC evaluation results for all assurance levels.

The present document gives guidance to standards authors (rapporteurs and contributors) on the scope and application of the Common Criteria for Information Technology Security Evaluation [20] and how ETSI standards may be developed to meet the goals and objectives of the Common Criteria.

The purpose of the present document is to provide developers of security standards with a summary of the requirements of ISO/IEC-15408 [20] in the context of standardization and to give guidance on how formal methods and other engineering techniques can be used to ensure that standards meet, as far as is possible, the requirements of ISO/IEC 15408 [20] and do not prevent an implementation from achieving an appropriate EAL.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

[1] ETSI EN 300 392-7: "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 7: Security".

[2] ETSI ES 202 382: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Security Design Guide; Method and proforma for defining Protection Profiles".

[3] ETSI ES 202 383: "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Security Design Guide; Method and proforma for defining Security Targets".

[4] ETSI TS 102 237-1: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Interoperability test methods and approaches; Part 1: Generic approach to interoperability testing".

[5] ETSI ETS 300 406: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

[6] ETSI ETR 332: "Security Techniques Advisory Group (STAG); Security requirements capture".

[7] ETSI EG 201 383: "Methods for Testing and Specification (MTS); Use of SDL in ETSI deliverables; Guidelines for facilitating validation and the development of conformance tests".

[8] ETSI EG 201 872: "Methods for Testing and Specification (MTS); Methodological approach to the use of object-orientation in the standards making process".

[9] ETSI EG 202 106: "Methods for Testing and Specification (MTS); Guidelines for the use of formal SDL as a descriptive tool".

[10] ITU-T Recommendation I.130: "Method for the characterization of telecommunication services supported by an ISDN and network capabilities of an ISDN".

[11] ETSI EG 201 015: "Methods for Testing and Specification (MTS); Specification of protocols and services; Validation methodology for standards using Specification and Description Language (SDL); Handbook".

[12] ETSI EG 201 058: "Methods for Testing and Specification (MTS); Implementation Conformance Statement (ICS) proforma style guide".

[13] ETSI EG 202 107: "Methods for Testing and Specification (MTS); Planning for validation and testing in the standards-making process".

[14] ETSI ETR 184: "Methods for Testing and Specification (MTS); Overview of validation techniques for European Telecommunication Standards (ETSs) containing SDL".

[15] ETSI SR 001 262: "ETSI Drafting rules".

[16] ISO/IEC 13335 (parts 1 to 5): "Information technology - Guidelines for the Management of IT Security (GMITS)".

[17] ISO/IEC 15408-1: "Information technology - Security techniques - Evaluation criteria for IT security - Part 1: Introduction and general model".

[18] ISO/IEC 15408-2: "Information technology - Security techniques - Evaluation criteria for IT security - Part 2: Security functional requirements".

[19] ISO/IEC 15408-3: "Information technology - Security techniques - Evaluation criteria for IT security - Part 3: Security assurance requirements".

[20] ISO/IEC 15408: "Information technology - Security techniques - Evaluation criteria for IT security".

NOTE: When referring to all parts of ISO/IEC 15408 the reference above is used.

[21] ISO/IEC 9646 (parts 1 to 7): "Information technology - Open Systems Interconnection - Conformance testing methodology and framework".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

asset: information or resource of a Target of Evaluation (TOE) to be protected by the countermeasures

conformance testing: testing the extent to which an Implementation Under Test (IUT) satisfies both static and dynamic conformance requirements

NOTE: That is, the purpose of conformance testing is to determine to what extent a single implementation of a particular standard conforms to the individual requirements of that standard.

interoperability testing: activity of proving that end-to-end functionality between (at least) two communicating systems is as required by the base standard(s) on which those systems are based

Protection Profile (PP): implementation-independent set of security requirements that meets specific consumer needs for a category of TOEs

Security Target (ST): set of security requirements and specifications to be used as the basis for evaluation of an identified TOE

Target Of Evaluation (TOE): IT product, an IT system, or a Protection Profile and its associated administrator and user guidance documentation that is the subject of an evaluation

vulnerability: weakness of an asset or group of assets, which can be exploited by one or more threats (source: ISO/IEC 13335 [16])

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ASN.1	Abstract Syntax Notation 1
CC	Common Criteria

NOTE: For Information Technology Security Evaluation.

CM	Change Management
EAL	Evaluation Assurance Level
FE	Functional Entity
ICS	Implementation Conformance Statement
IDL	Interface Description Language
IETF	Internet Engineering Task Force
IT	Information Technology
LOTOS	Language Of Temporal Ordering Specification
MSC	Message Sequence Chart
PDF	Portable Document Format
PICS	Protocol Implementation Conformance Statement
PP	Protection Profile
ST	Security Target
SDL	Specification and Description Language
TOE	Target of Evaluation
TSF	TOE Security Functions
UML	Unified Modelling Language
XML	eXtensible Markup Language

4 Security in standardization

4.1 Communications security model

In the context of the present document, security means to be assured that the risk of a weakness being exploited either intentionally or unintentionally is low.

Many standards include aspects of security, such as:

- confidentiality;
- integrity;
- availability.

The goals of security and of evaluation are:

- to provide product owners with confidence that countermeasures bring the risk to assets to an acceptable level;
- to implement assurance techniques which give confidence that countermeasures bring the risk to assets to an acceptable level;
- to ensure that evaluation provides evidence of assurance giving confidence that countermeasures bring the risk to assets to an acceptable level.

The standardization process plays a significant role in achieving these objectives. Firstly, in order to ensure that the requirements identified in a standard are expressed accurately, clearly and unambiguously, a standard is critically reviewed by its potential implementors. Such review, along with other validation techniques, helps to provide the assurance that any specified countermeasures will, in fact, minimize risk. Secondly, a protocol standard is accompanied by a conformance test specification which can be used in the evaluation process to provide evidence that any countermeasures required by the protocol standard have been implemented correctly in a product.

4.2 Standards review and evaluation

In the ETSI environment the technical content of standards is not formally reviewed although the editorial structure is mechanically checked against the ETSI drafting rules [15] (primarily to ensure that the ETSI stylesheet is properly applied to ETSI deliverables). The review of standards is an essential element of quality assurance and whilst the editorial level review is an important aspect of this the review of technical content is similarly important. Standards need to be reviewed and the review or evaluation process should be standardized and repeatable in order to give some level of trust in the standard.

In the evaluation process described by ISO/IEC 15408 [20] the review is expected to be formal and to follow a set of evaluation rules. The input to the evaluation process is also expected to be in a prescribed format so that the rules and guidance given to evaluators can be applied. The present document gives guidance to standards developers in preparing standards in a format that can be assessed by evaluators using the Common Criteria.

In the ISO/IEC 15408 [20] evaluation process there are a number of evaluation levels. Each increasing level of evaluation, of which there are 7, requires that the developer provides more information on the system for review and that the reviewer reviews to a greater depth. It should be noted that passing a review at a higher evaluation level does not increase the overall security of an evaluated system. It only indicates that the security aspects of the system have been evaluated with more rigour and that the developer has submitted more evidence to support the evaluation.

4.3 Overall development process

Prior to defining the detailed security requirements for a new standard, it is essential to identify:

- the purpose of a system implementing the standard;
- what level of risk is acceptable to the users of such a system;
- how claims for the security of such a system will be evaluated; and
- any specific evaluation and assurance requirement required (or likely to be required) by the end users.

EXAMPLE: It is a requirement that security products supplied for Government use are designed and evaluated according to Evaluation Assurance Level 5 of ISO/IEC 15408 [20].

As shown in figure 1, all of these aspects contribute to the definition of the security requirements to be specified in the standard and met by a product implementing it.

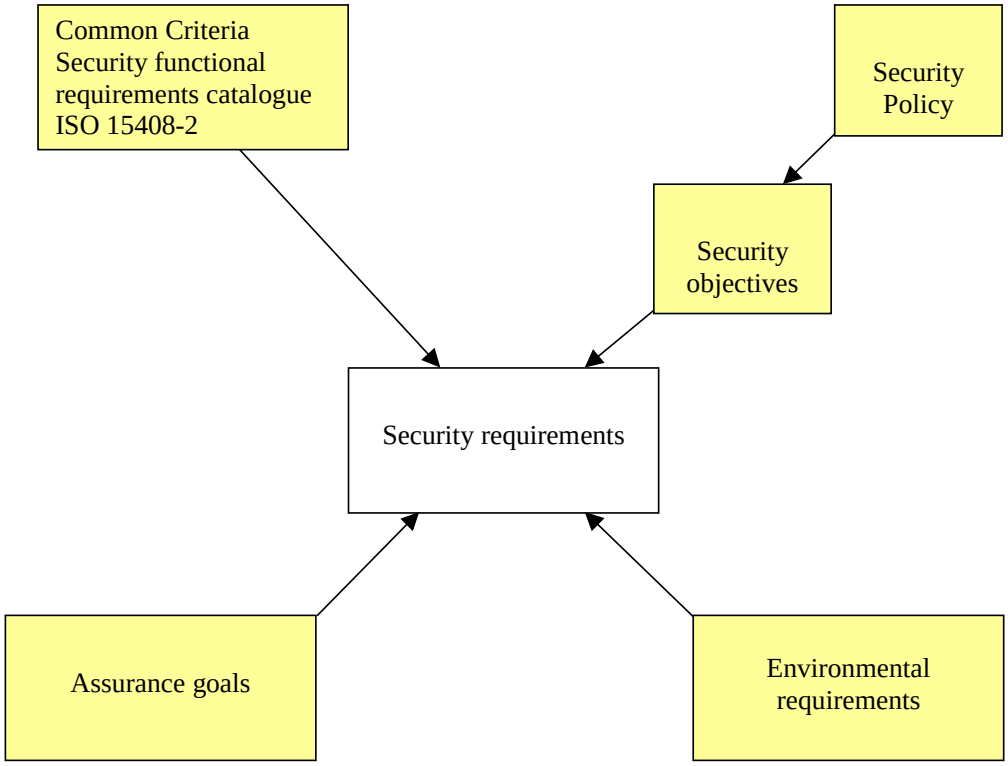


Figure 1: Composition of security requirements

ETSI security standards that may become the subject of evaluation should:

- state technical security requirements in terms of the security functions defined in ISO/IEC 15408-2 [18];
- be documented as required by the ISO/IEC 15408-3 [19] assurance level required by the project.

NOTE: The documentation requirements of each assurance level that can be addressed by ETSI are outlined in the present document.

The provision of a vulnerability analysis is a core requirement of ISO/IEC 15408 [20] as a means of ensuring that the implemented security solution fits the security context. Vulnerability analyses should be developed and documented according to the guidelines described in ETR 332 [6] which is closely aligned to the process defined in ISO/IEC 15408-1 [17]. Figure 2 illustrates the relationships between TOE development activities and the information associated with each of these activities. It shows that a vulnerability analysis continues throughout the overall TOE development process. At each stage of the analysis, the input information (objectives, requirements, design) is modified if necessary and control either passes on to the next activity or back to an earlier activity where the analysis indicates that further development is required.

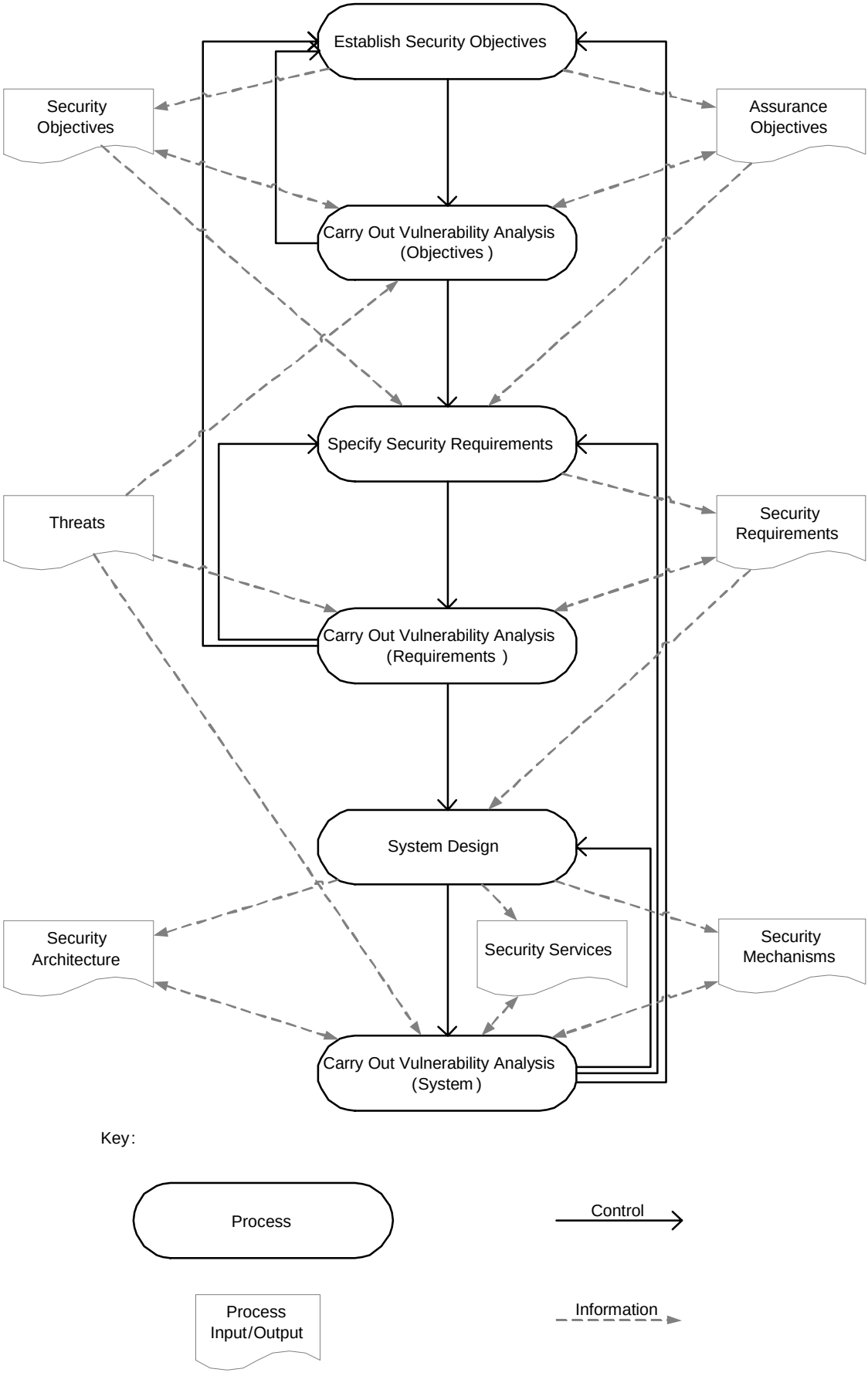


Figure 2: Structure of security analysis and development in standards documents

For the purposes of analysis, all assets should be considered to have weaknesses.

Figure 3 shows how the vulnerabilities that surround a system may be attacked by known threats and may be countered by known security countermeasures. In some instances a residual vulnerability may exist even after application of the countermeasure.

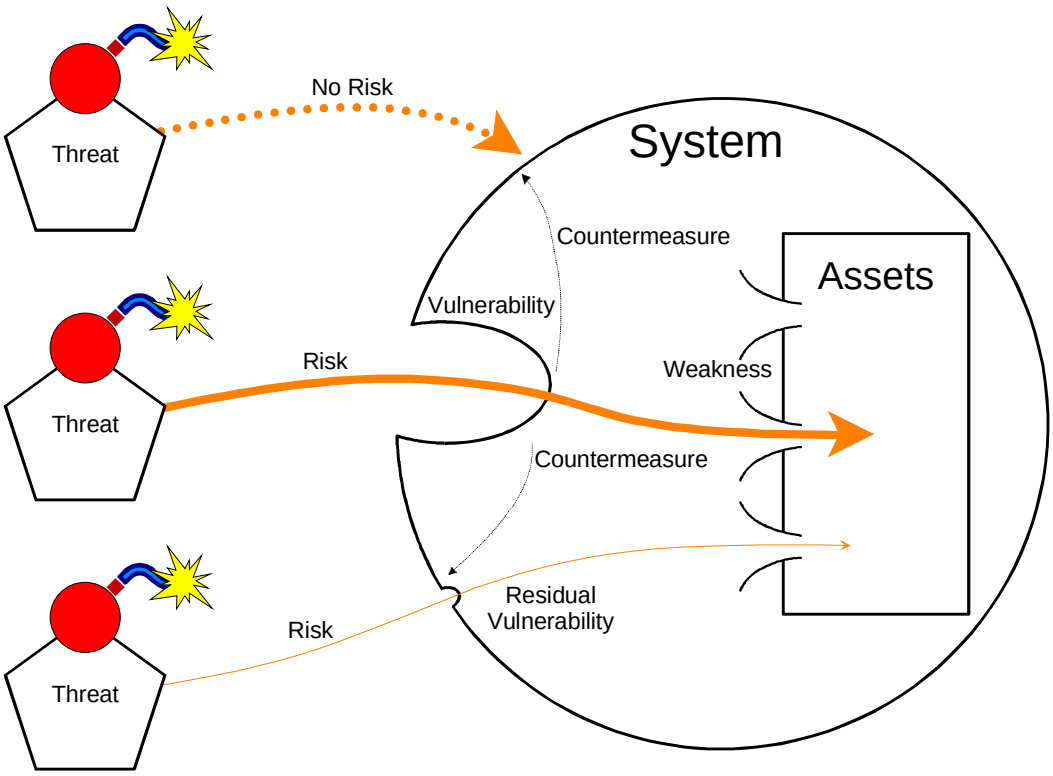


Figure 3: Threats, risks, vulnerabilities, countermeasures and residual vulnerability

A potential threat is able to cause harm only if there is a corresponding weakness or vulnerability in the system which can be exploited. Thus it is necessary to evaluate threats and to characterize them according to both the likelihood of their occurrence and of the impact the attack has.

4.4 Protocol standards containing security-related requirements

A Standard which does not directly specify security requirements may still contain some security-related aspects. These implicit security requirements should be identified and analysed early in the development of the standard so that they can be taken into account in the specification. Adding security requirements at a later stage can be costly and inefficient. In addition, the means of evaluating any security requirements against the CC should also be considered as this, too, could have an impact on the specification of the standard.

NOTE 1: For the purposes of this guide it is assumed that evaluation is performed according to ISO-15408-3 [19].

The security considerations which are the results of this analysis should be summarized in an annex to the standard.

NOTE 2: Current Internet-drafts and IETF RFCs have such a section as mandatory content.

The "Security considerations" annex should also include the results of the vulnerability analysis.

5 Overview of ISO/IEC 15408

5.1 Introduction to the Common Criteria (CC)

The primary purpose of the Common Criteria for Information Technology Security Evaluation [20], usually referred to simply as the Common Criteria (CC), is to harmonize the security evaluation of products by:

- defining a common set of terms related to the evaluation of security requirements;
- defining a set of procedures to be followed by both product developers and security evaluation authorities.

A product or service that is to be evaluated under the CC is referred to as a Target Of Evaluation (TOE) and it is the developer's responsibility to provide evidence that the security provisions for a TOE have been designed and implemented to meet the requirements of the CC.

There are 2 forms of TOE:

- Protection profile (PP);
- Security target (ST) and its corresponding product.

A standard can be used to form part of a PP and can be referred to in the construction of an ST.

5.1.1 Contents of a Protection Profile (PP)

Annex B of ISO/IEC 15408-1 [17] defines the outline structure of a PP as follows:

- PP Introduction:
  - PP identification;
  - PP overview.
- Target Of Evaluation description;
- TOE security environment:
  - Assumptions;
  - Threats;
  - Organizational security policies.
- Security objectives:
  - Security objectives for the TOE;
  - Security objectives for the environment.
- IT security requirements:
  - TOE security requirements:
    - TOE security functional requirements;
    - TOE security assurance requirements.
  - Security requirements for the IT environment (OPTIONAL).
- Application notes (OPTIONAL);

- Rationale:
  - Security objectives rationale;
  - Security requirements rationale.

ES 202 382 [2] defines a method for preparing a PP based on an ETSI standard.

5.1.2 Contents of a Security Target (ST)

Annex C of ISO/IEC 15408-1 [17] defines the outline structure of an ST as follows:

- ST Introduction:
  - ST identification;
  - ST overview;
  - CC conformance claim.
- Target Of Evaluation description;
- TOE security environment:
  - Assumptions;
  - Threats;
  - Organizational security policies.
- Security objectives:
  - Security objectives for the TOE;
  - Security objectives for the environment.
- IT security requirements:
  - TOE security requirements:
    - TOE security functional requirements;
    - TOE security assurance requirements.
  - Security requirements for the IT environment (OPTIONAL).
- TOE summary specification:
  - Statement of TOE security specifications;
  - Statement of assurance measures.
- PP claims:
  - PP reference;
  - PP tailoring;
  - PP additions.

- Rationale:
  - Security objectives rationale;
  - Security requirements rationale;
  - TOE summary specification rationale;
  - PP claims rationale.

ES 202 383 [3] defines a method for preparing an ETSI ST.

5.1.3 Common Criteria relationships

Figure 4 shows a simplified view of the relationships between security Protection Profiles (PP), Security Targets (ST) and Targets Of Evaluation (TOE).

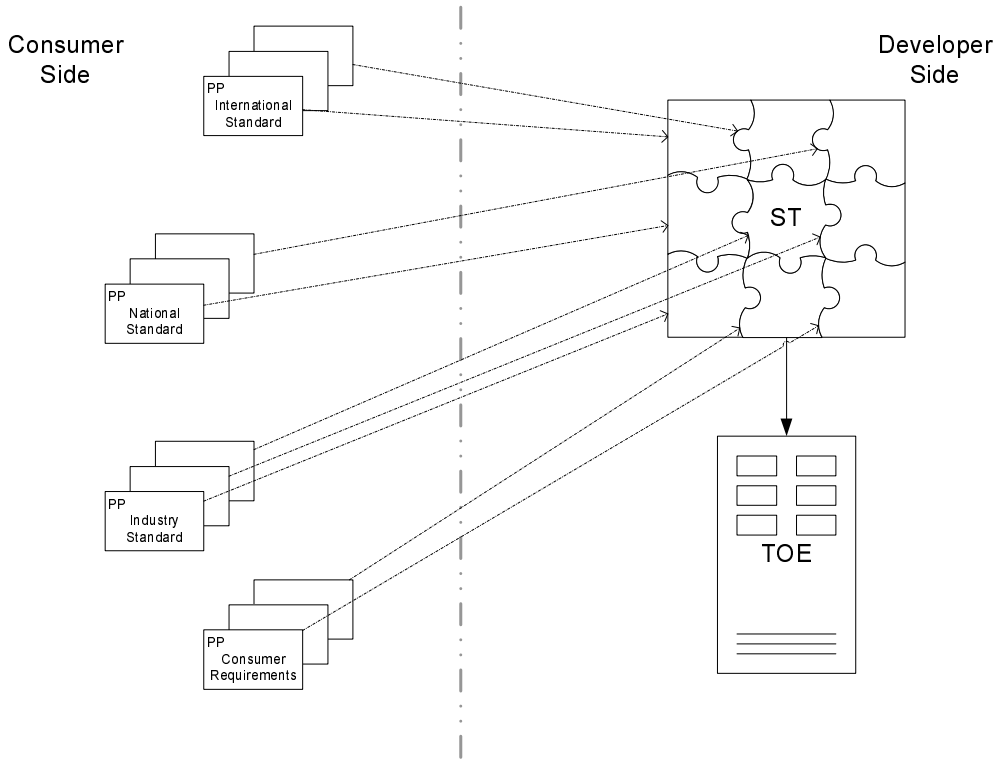


Figure 4: Relationship between PPs, STs and TOEs

5.1.4 Evaluation Assurance Levels

ISO/IEC 15408-3 [19] defines a set of Evaluation Assurance Levels (EAL) which identify various degrees of assurance that can be achieved by a TOE during assessment of its security functions. The full range of EALs is shown in table 1.

Table 1: Evaluation Assurance Levels

Assurance level	Outline definition
EAL0 (note)	No evaluation undertaken
EAL1	Functionally tested
EAL2	Structurally tested
EAL3	Methodically tested and checked
EAL4	Methodically designed, tested and reviewed
EAL5	Semi formally designed and tested
EAL6	Semi formally verified designed and tested
EAL7	Formally verified design and tested
NOTE: EAL0 is generally not recognized as no evaluation is undertaken.	

5.2 Overview of CC documents

5.2.1 ISO/IEC 15408-1: Introduction and general model

ISO/IEC-15408-1 [17] provides a general overview of the Common Criteria evaluation model. It defines a range of security terms within the CC context, describes the model itself, identifies what results are expected from an evaluation and, in its normative annexes, specifies the content of a PP and an ST.

5.2.2 ISO/IEC 15408-2: Security functional requirements

ISO/IEC 15408-2 [18] specifies a formal set of security functional requirements which together describe the security behaviour expected of a Target Of Evaluation (TOE). These requirements, summarized in annex A, are defined as a catalogue of components and classes which can be extended and specialized to suit particular TOE applications.

5.2.3 ISO/IEC 15408-3: Security assurance requirements

ISO/IEC 15408-3 [19] describes the evaluation process by defining a set of evaluation classes and specifying the actions of an evaluator. Guidance on meeting the CC evaluation requirements is given in clause 6 of the present document.

5.3 ETSI standards in the evaluation of CC

Evaluation in the context of ISO/IEC-15408 [20] is a test of the product against a set of defined security criteria. It is not a test that the product functions completely and correctly but a test that the product meets the security-related claims made for it. As such, this type of testing is similar to the conformance testing which is normally specified for all protocol standards. However, it is possible that interoperability testing (or a combination of conformance and interoperability testing) may be a more appropriate means of evaluating security criteria. Further details of the methods involved in the development of test specifications for both conformance and interoperability can be found in ISO/IEC 9646 [21] and TS 102 237-1 [4].

6 Evaluation components in ISO/IEC-15408-3

6.1 Introduction

This clause gives guidance on how standards developers should read and interpret the evaluation components in ISO/IEC 15408-3 [19]. The guidance will assist standards developers to prepare standards that meet the evaluation requirements of the standards group.

ISO/IEC 15408-3 [19] defines seven Evaluation Assurance Levels (EAL) where each level extends the previous level with additional requirements for proof that the design has been carried out rigorously. Table 6.1 of ISO/IEC 15408-3 [19] (duplicated here in table 2) details the assurance components that apply at each EAL.

Figure 5 shows the hierarchical structure of classes as families and components defined in ISO/IEC 15408-3 [19]. Each EAL (see clause 5.1.4) identifies the classes, families and components that will be selected by the evaluator.

NOTE: A higher EAL does not mean the PP or TOE represents a more secure product but only that the evaluation of the security claim has been made with more rigour and that the developer has submitted more proof to support the claim.

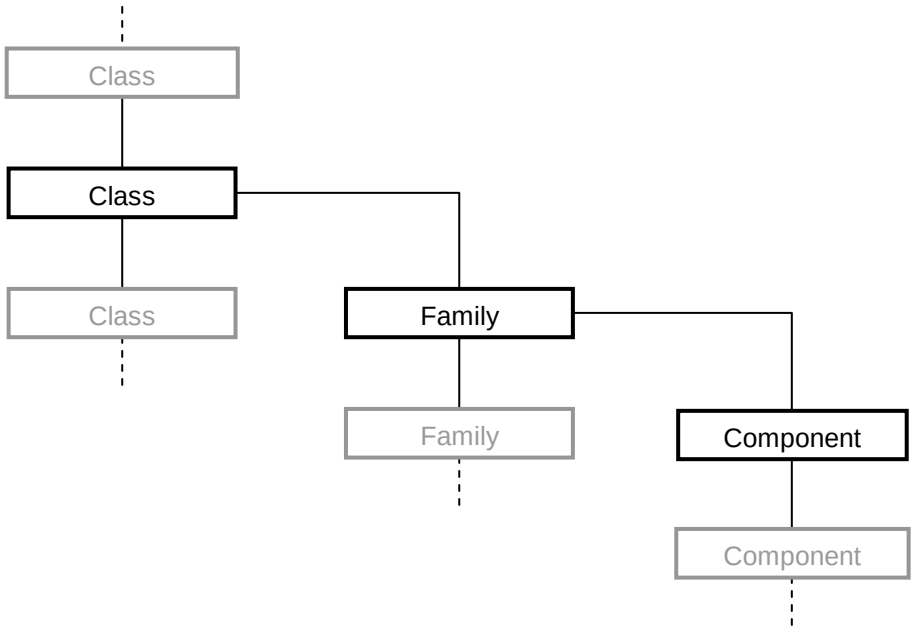


Figure 5: Hierarchical structure of assurance evaluation classes

Table 2: Evaluation service level summary as specified in ISO/IEC 15408-3 [19]

Assurance class	Assurance family	Assurance components by EAL						
		1	**2**	**3**	**4**	**5**	**6**	**7**
Configuration management	CM automation				1	1	2	2
	CM capabilities	1	2	3	4	4	5	5
	CM scope			1	2	3	3	3
Delivery and operation	Delivery		1	1	2	2	2	3
	Installation, generation and startup	1	1	1	1	1	1	1
Development	Functional specification	1	1	1	2	3	3	4
	High level design		1	2	2	3	4	5
	Implementation representation				1	2	3	3
	TSF internals					1	2	3
	Low level design				1	1	2	2
	Representation correspondence	1	1	1	1	2	2	3
	Security policy modelling				1	3	3	3
Guidance documents	Administrator guidance	1	1	1	1	1	1	1
	User guidance	1	1	1	1	1	1	1
Life cycle support	Development security			1	1	1	2	2
	Flaw remediation							
	Life cycle definition				1	2	2	3
	Tools and techniques				1	2	3	3
Tests	Coverage		1	2	2	2	3	3
	Depth			1	1	2	2	3
	Functional tests		1	1	1	1	2	2
	Independent testing	1	2	2	2	2	2	3
Vulnerability assessment	Covert channel analysis					1	2	2
	Misuse			1	2	2	3	3
	Strength of TOE security functions		1	1	1	1	1	1
	Vulnerability analysis		1	1	2	3	4	4

6.2 Configuration management

6.2.1 Class description

The purpose of this evaluation class is to ensure that the integrity of the TOE is preserved by requiring discipline and control in any process used to refine and maintain the various capabilities (options) of the TOE.

The families of this class cover three aspects of configuration management:

1) The degree to which configuration management is automated;

2) The capabilities of the developer's configuration management system; and

3) The extent to which configuration management is effectively used in the development and maintenance of the TOE.

6.2.2 Implications for the standardization process

The standards development process is primarily concerned with the production of specification documents rather than hardware or software products. Consequently, the control of changes to documents during development is informal and based upon the use of change marking within the text itself and peer-reviews to determine the validity of proposed changes. The use of change management systems as part of the maintenance of published standards is addressed in clause 6.6.

6.2.3 Families and components

The components of each of the "Configuration management" evaluation families cover a wide range of activities from supplying version numbers through to complete automation of the configuration of the TOE. The relationship between meeting the requirements of this class and achieving a particular EAL is summarized in table 3.

Table 3: Configuration management" family evaluation levels

Evaluation component		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Automation								
ACM_AUT.1	Partial CM automation				✓	✓		
ACM_AUT.2	Complete CM automation						✓	✓
Capability								
ACM_CAP.1	Version numbers	✓						
ACM_CAP.2	Configuration items		✓					
ACM_CAP.3	Authorization controls			✓				
ACM_CAP.4	Generation support and acceptance procedures				✓	✓		
ACM_CAP.5	Advanced support						✓	✓
Scope								
ACM_SCP.1	TOE CM coverage			✓				
ACM_SCP.2	Problem tracking CM coverage				✓			
ACM_SCP.3	Development tools CM coverage					✓	✓	✓

Although a rigorous version numbering and identification scheme is applied to all ETSI deliverables, there is no configuration management system available as part of the standards development process. Consequently, no further definition of the ACM families and components is given in the present document.

6.3 Delivery and operation

6.3.1 Class description

The objective of the assurance class "Delivery and operation" (ADO) is to ensure the integrity of the TOE when it is arrives at a customer's site. The TOE manufacturer is expected to have processes in place which ensure that the delivered TOE corresponds exactly to the current master copy.

6.3.2 Implications for the standardization process

Completed ETSI deliverables are supplied as either PDF or Microsoft Word files where the master copy is always retained at ETSI. The PDF files are offered as read-only. ETSI deliverables do not need to be installed, generated or started-up.

There is no impact on the ETSI standardization process of this assurance class.

6.3.3 Families and components

The components of each of the "Delivery and operation" evaluation families cover a wide range of activities from documenting the delivery procedures through to documenting how the TOE is to be installed and started. The relationship between meeting the requirements of these components and achieving particular evaluation assessment levels is summarized in table 4.

Table 4: "Delivery and operation" family evaluation levels

Evaluation component		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Delivery								
ADO_DEL.1	Delivery procedures		✓	✓				
ADO_DEL.2	Detection of modification				✓	✓	✓	
ADO_DEL.3	Prevention of modification							✓
Installation, generation and start-up								
ADO_IGS.1	Installation, generation and startup procedures	✓	✓	✓	✓	✓	✓	✓
ADO_IGS.2	Generation log							

6.4 Development

6.4.1 Class description

The assurance class, "Development" (ADV) implies requirements for the methods used during the development of a TSF. There are 7 families described in this class and these are as follows:

- Functional specification (ADV_FSP).
- High-level design (ADV_HLD).
- Implementation representation (ADV_IMP).
- Internals (ADV_INT).
- Low-level design (ADV_LLD).
- Representation correspondence (ADV_RCR).
- Security policy modelling (ADV_SPM).

In the context of CC evaluation, the development process is expected to show that a stepwise refinement from a summary specification down to the actual specification has been used throughout the development process. Each of the resulting representations should provide information to help the evaluator determine whether the functional requirements have been met.

NOTE: Modern development processes do not always follow a stepwise refinement where each step is fully signed-off before the next step commences. Advances in product development technology have made it possible to implement a process of overlapping, parallel activities.

6.4.2 Implications for the standardization process

The development families together imply a functional specification process with the following sequential activities:

- decomposition of the system into subsystems;
- decomposition of the subsystems into modules;
- description of the behaviour of the modules; and
- demonstration of correspondence between all decompositions that are provided as evidence.

If a communications standard is to be successfully evaluated against the requirements for a PP specified in annex B of ISO/IEC 15408-1 [17], it is necessary for its development process to follow strict guidelines. These involve top-down decomposition of the specification, the use of specification languages such as SDL or UML, planned validation of the specification and the careful recording of design decisions and validation results. In addition, it is also necessary for a vulnerability analysis to be undertaken for all standards specifying security-related aspects. This analysis will provide essential information which can be used in the development of security services and requirements.

Although the development process implied in ISO/IEC 15408-3 [19] is not identical to that generally assumed for communication standards, the mapping of activities between them is straightforward. Figure 6 shows how the component families within the CC Development class relate to specifications produced during the standardization of a communication protocol. Both the vulnerability analysis and the specification of functional requirements (equivalent to a stage 1 protocol specification) are expected to take place prior to the start of the development process itself. The Development class then describes the CC evaluation requirements for the process of producing stage 2 and stage 3 standards (or their equivalents). The Implementation Representation family (ADV_IMP) applies only to physical products and can be ignored for standardization purposes.

The families described in the Development class of ISO/IEC 15408-3 [19] are primarily concerned with the provision of evidence for evaluation and, consequently, specify criteria for the representation of designs at particular levels of detail rather than for the underlying development process itself. However, protocol standards that follow the method specified in ITU-T Recommendation I.130 [10] and include correctly defined formal language specifications such as SDL or UML are likely to satisfy the evaluation requirements of this class. Guidance on the use of such languages in standards can be found on the ETSI "Making Better Standards" web site at http://portal.etsi.org/mbs.

NOTE: ISO/IEC 15408 [20] uses the terms "formal", "semiformal" and "informal" to distinguish between mathematically-based specification languages, those that have a defined semantics and those that have no specified semantics at all. Mathematically-based languages are rarely used in the specification of communications standards and, as a result, the term "formal specifications" has been used to refer to those developed using SDL, UML, MSC and ASN.1 tools while "informal specifications" are those that contain diagrams (often representations of SDL and MSC) produced within a drawing tool and free text. The term "semiformal" is not used in this context.

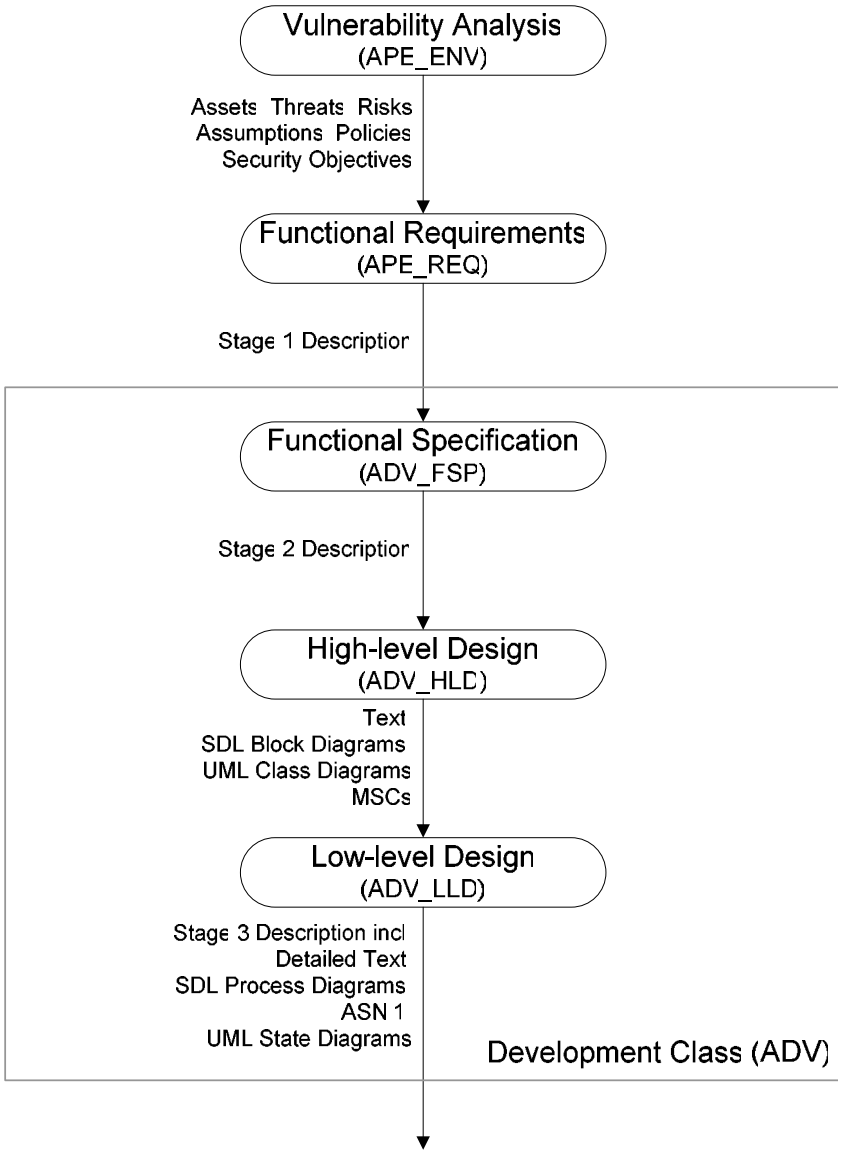


Figure 6: Relationships between CC development activities and the standardization process

6.4.3 Families and components

6.4.3.1 Development class evaluation levels

The components of each of the "Development" evaluation families cover a wide range of activities from informal functional specification through to formal low-level design and security policy modelling. The relationship between meeting the requirements of these components and achieving particular evaluation assessment levels is summarized in table 5.

Table 5: "Development" family evaluation levels

Evaluation component		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Functional specification								
ADV_FSP.1	Informal functional specification	✓	✓	✓				
ADV_FSP.2	Fully defined external interfaces				✓			
ADV_FSP.3	Semiformal functional specification					✓	✓	
ADV_FSP.4	Formal functional specification							✓
High-level design								
ADV_HLD.1	Descriptive high-level design		✓					
ADV_HLD.2	Security enforcing high-level design			✓	✓			
ADV_HLD.3	Semiformal high-level design					✓		
ADV_HLD.4	Semiformal high-level explanation						✓	
ADV_HLD.5	Formal high-level design							✓
Implementation representation								
ADV_IMP.1	Subset of the implementation of the TSF				✓			
ADV_IMP.2	Implementation of the TSF					✓		
ADV_IMP.3	Structured implementation of the TSF						✓	✓
TSF internals								
ADV_INT.1	Modularity					✓		
ADV_INT.2	Reduction of complexity						✓	
ADV_INT.3	Minimization of complexity							✓
Low-level design								
ADV_LLD.1	Descriptive low-level design				✓	✓		
ADV_LLD.2	Semiformal low-level design						✓	
ADV_LLD.3	Formal low-level design							✓
Representation correspondence								
ADV_RCR.1	Informal correspondence demonstration	✓	✓	✓	✓			
ADV_RCR.2	Semiformal correspondence demonstration					✓	✓	
ADV_RCR.3	Formal correspondence demonstration							✓
Security policy modelling								
ADV_SPM.1	Informal security policy model				✓			
ADV_SPM.2	Semiformal security policy model					✓	✓	✓
ADV_SPM.3	Formal security policy model					✓	✓	✓

A security-related protocol which is standardized using the three-stage approach described in ITU-T Recommendation I.130 [10] and which follows the guidelines found on ETSI's "Making Better Standards" web site (http://portal.etsi.org/mbs) is likely to meet the broad requirements expressed in the Development class. Standards developed in this way should meet the evaluation requirements at EAL4 or above.

6.4.3.2 Functional specification family (ADV_FSP)

The functional specification is a high-level description of the required behaviour of systems that implement the standard. The information required as evidence for the evaluation of this family can be found in a stage 1 specification (service description) of a service as described in ITU-T Recommendation I.130 [10].

6.4.3.2.1 Informal functional specification (ADV_FSP.1)

Applicable to: EAL1, EAL2 and EAL3

Requirements should be expressed, as far as possible, entirely from the user's perspective (although the "user" may be a terminal or network application acting on behalf of a human user) using free text and some informal diagrams. At this stage of the specification, there should be no need to consider the possible physical architecture of any system implementing the requirements.

The component ADV_FSP.1 requires that the "purpose and method of use of external TSF interfaces" is described giving details of normal events, exceptions and error messages. This description is not required to be complete and, for this reason, should not be used as the basis for a stage 1 specification. Instead, a stage 1 should conform to the requirements specified for either the component ADV_FSP.2 (see clause 6.4.3.2.2) or ADV_FSP.3 (see clause 6.4.3.2.3).

6.4.3.2.2 Fully defined external interfaces (ADV_FSP.2)

Applicable to: EAL4

To achieve the evaluation level set by the "Fully defined external interfaces" component, it is first necessary to meet the requirements of the component ADV_FSP.1 (see clause 6.4.3.2.1). In addition, a complete description of all normal and exceptional behaviour in relation to external interfaces should be provided with a definition of any error messages. A fully specified stage 1 standard should include descriptions of both normal and exceptional behaviour but not in relation to external interfaces. Nor will it include the definition of error messages. External interfaces and message details are generally specified in the stage 3 standard which is considered within the Low-Level Design family of components (see clause 6.4.3.6).

ADV_FSP.2 also expects the functional specification to include an explanation of why the defined user requirements completely represent the TOE Security Functions. This rationale will only become evident once a vulnerability analysis has been undertaken and should be included in the documentary output of this activity.

6.4.3.2.3 Semiformal functional specification (ADV_FSP.3)

Applicable to: EAL5 and EAL6

Evaluation to the level set by the "Semiformal functional specification" component requires the same evidence identified in ADV_FSP.2 (see clause 6.4.3.2.2) but the functional specification itself should be presented in a semiformal language supported by explanatory text. ISO/IEC 15408-3 [19] uses the term "semiformal" to refer to specification languages and notations such as SDL, UML and MSC. Semiformal functional specifications should also include explanatory text to assist in the interpretation of the elaborated requirements.

6.4.3.2.4 Formal functional specification (ADV_FSP.4)

Applicable to: EAL7

Communications protocol standards rarely use specification languages which have mathematically defined semantics. Thus, it is unlikely that such standards could successfully achieve evaluation to the "Formal functional specification" component. However, standards which use languages such as LOTOS and Z to specify behaviour or which define security algorithms for encryption and authentication purposes could comply with the requirements of this component if they also meet the requirements of ADV_FSP.1 (see clause 6.4.3.2.1).

6.4.3.3 High-level design family (ADV_HLD)

A high-level design is equivalent to a stage 2 specification [10]. It provides a description of the functions in terms of subsystems and relates these units to the functions that they provide (see also clause 6.4.3.5). The high-level design refines the functional specification of clause 6.4.3.2 into Functional Entities (FEs). The high-level design should describe purpose and function of each FE and identify the specific security functions contained within it. The high-level design should identify the relationships that exist between FEs and the required flow of information across them.

6.4.3.3.1 Descriptive high-level design (ADV_HLD.1)

Applicable to: EAL2

To achieve the evaluation level set by the "Descriptive high-level design" component, the high-level design should describe the architecture of functional entities and the functionality provided by each entity. The functional entity model in a stage 2 protocol specification adequately describes the functional architecture of a protocol system as shown in the example in figure 7.

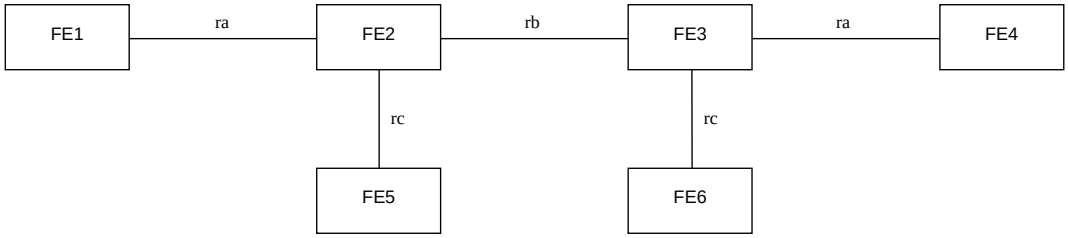


Figure 7: Example informal functional entity model

The role of each of the FEs should be described in text with particular attention paid to the security-related functions provided.

EXAMPLE: FE1 acts on behalf of the calling user to provide authentication data when requested by FE2.

FE5 computes the user authentication function from information provided by FE2.

The descriptive high-level design component requires the specification of neither the information flows between FEs nor the behaviour of each FE and, consequently, should not be used as the basis for the development of a stage 2 specification or its equivalent. Instead, a stage 2 standard should meet at least the requirements of ADV_HLD.2.

6.4.3.3.2 Security enforcing high-level design (ADV_HLD.2)

Applicable to: EAL3 and EAL4

In order to achieve the evaluation level set by the "Security enforcing high-level design" component, it is first necessary to meet the requirements of ADV_HLD.1 (see clause 6.4.3.3.1). Then the contents of all information which needs to flow across the relationships between functional entities must be specified along with the behaviour of each FE in processing this information. Descriptions of significant normal and exceptional behaviour as well as the reporting of important error conditions should be included as appropriate. These can be specified using free text, tables (for information flow contents) and informal flow sequence diagrams as shown in figure 8.

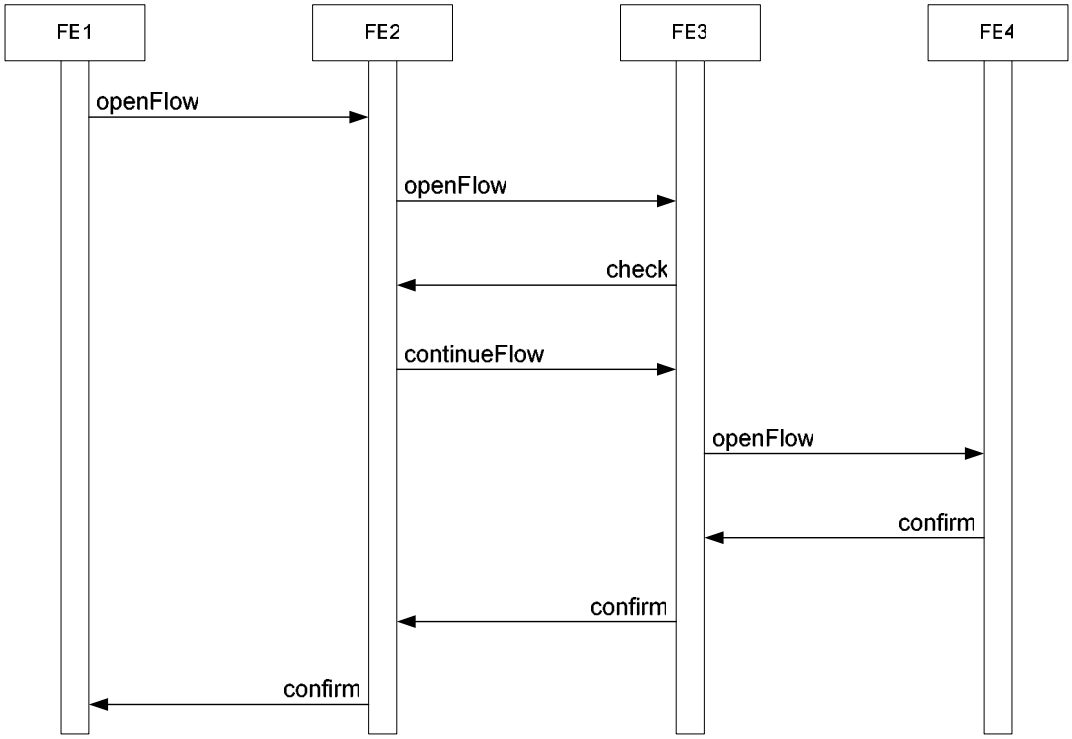


Figure 8: Example informal information flow diagram

A stage 2 specification for a protocol or service which provides security functionality should also identify those FEs that realize the security functions.

6.4.3.3.3 Semiformal high-level design (ADV_HLD.3)

Applicable to: EAL5

Evaluation to the level set by the "Semiformal high-level design" component requires the same evidence identified in ADV_HLD.2 (see clause 6.4.3.3.2) but the specification itself should be presented in a semiformal language supported by explanatory text and should include details of all normal and exceptional behaviour as well as the reporting of all error conditions. Instead of the informal diagrams shown in figures 7 and 8, graphical languages such as UML, SDL and MSC should be used to describe functional architectures, behaviour and information flow sequences (see figures 9 and 10). Guidelines on the use of these languages for this purpose can be found in EG 201 872 [8] and EG 202 106 [9] and on ETSI's "Making Better Standards" web site at http://portal.etsi.org/mbs.

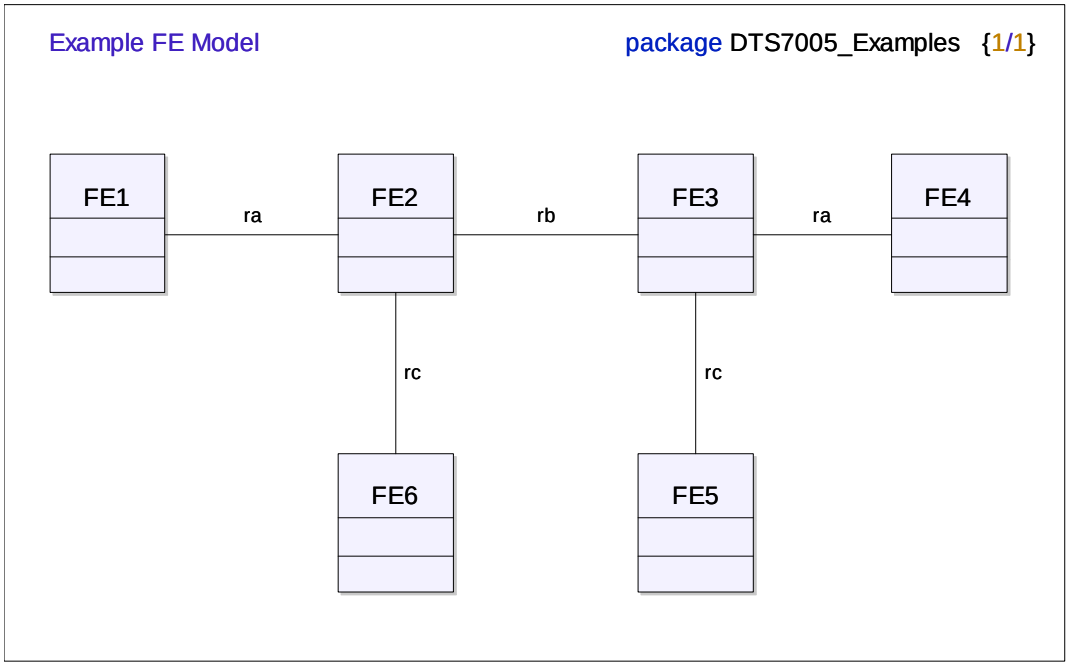


Figure 9: Example UML class diagram as a functional entity model

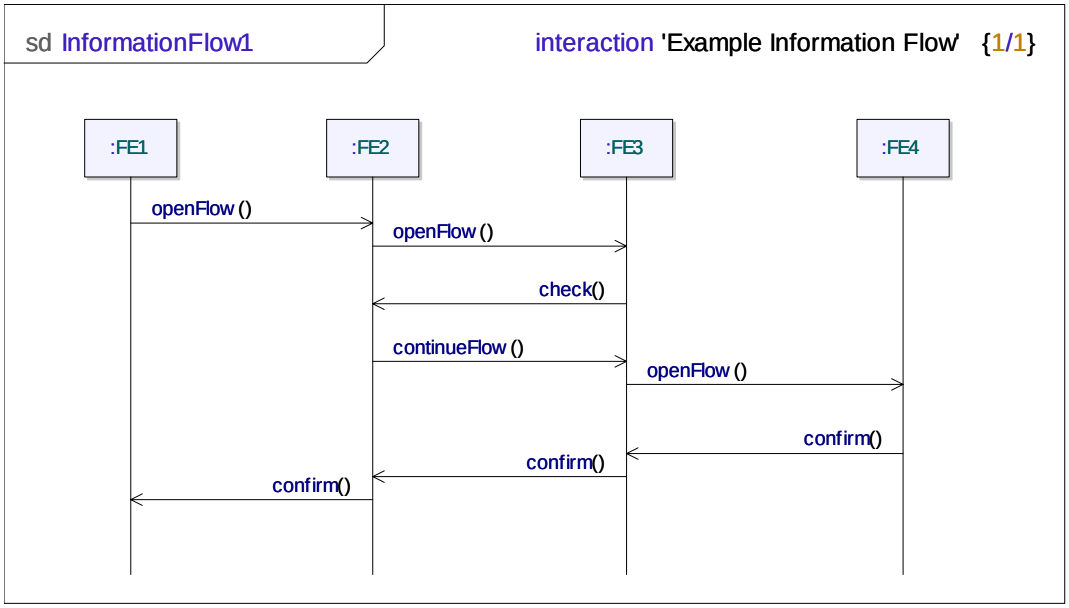


Figure 10: Example UML information flow diagram

6.4.3.3.4 Semiformal high-level explanation (ADV_HLD.4)

Applicable to: EAL6

Evaluation to the level set by the "Semiformal high-level explanation" component requires the same evidence identified in ADV_HLD.3 (see clause 6.4.3.6.3) with the addition of justifications of why:

- the specified means of separating security functions from non-security functions is likely to be effective in an implementation of the standard; and
- the functional model implements the security functions identified in the ADV_FSP family of components (see clause 6.4.3.2).

6.4.3.3.5 Formal high-level design (ADV_HLD.5)

Applicable to: EAL7

Evaluation to the level set by the "Formal high-level design" component requires the same evidence identified in ADV_HLD.4 (see clause 6.4.3.3.4) with the exception that the specification should be based on formal design languages such as LOTOS and Z. As stated in clause 6.4.3.2.4, such languages are rarely used in communications standards.

6.4.3.4 Implementation representation family (ADV_IMP)

ISO/IEC 15408-3 [19] requires that design documentation includes an implementation representation in the form of source code, firmware, hardware drawings, etc. which specifies the detailed internal workings of the TOE. Communications standards do not generally provide such detail except in the specification of the data structures that constitute protocol messages. Behaviour is specified in abstract terms which places requirements on external (and, therefore, visible) interfaces. The means of achieving this behaviour in an implementation is entirely a matter for the implementor. One significant exception to this is the specification of algorithms for encryption and authentication purposes which are often provided as segments of source code written in a programming language such as C++ or Java.

6.4.3.4.1 Subset of the implementation of the TSF (ADV_IMP.1)

Applicable to: EAL4

In order to achieve evaluation to the "Subset of the implementation of the TSF" component, it is necessary to specify at least part of the standardized security functions as source code. In a stage 3 specification [10] of a protocol, the data structures which are used to construct protocol messages should be specified in a notation which has standardized or implicit encoding rules. Such notations include ASN.1 (http://portal.etsi.org/mbs/Languages/ASN.1/asn1.asp), bit-tables, IDL and XML.

6.4.3.4.2 Implementation of the TSF (ADV_IMP.2)

Applicable to: EAL5

Successful evaluation to the "Implementation of the TSF" component requires all behaviour as well as the data structures to be included in the specification as source code which should be implemented unchanged in any conformant product. This is impractical in a communications standard and can be ignored.

6.4.3.4.3 Structured implementation of the TSF (ADV_IMP.3)

Applicable to: EAL6 and EAL7

Successful evaluation to the "Structured implementation of the TSF" component requires all behaviour as well as the data structures to be included in the specification as source code which should be implemented unchanged in any conformant product. This is impractical in a communications standard and can be ignored.

6.4.3.5 Standard internals family (ADV_INT)

The "Standard internals" family addresses the internal structure of a specification. An architectural description is requested and requirements are defined for modularity, layering (to separate levels of abstraction and minimize circular dependencies), minimization of the complexity of policy enforcement mechanisms, and the minimization of the amount of non-policy-enforcing functionality within the modules relevant for security. In a product implementing a standard, this is likely to result in security functions that are simple enough to be analysed. However, such requirements are not generally applicable to communications standards.

6.4.3.5.1 Modularity and layering (ADV_INT.1)

Applicable to: EAL5

To achieve the evaluation level set by the "Modularity and layering" component, a standard should identify the modules that are relevant for the security including the purpose, the interfaces, all parameters, and the effects of each module of the security functions. A standard which follows the general guidelines for editing and presentation of the technical content is likely to meet these requirements.

6.4.3.5.2 Reduction of complexity (ADV_INT.2)

Applicable to: EAL6

The evaluation requirements of the "Reduction of complexity" component are not applicable to the development of communications standards.

6.4.3.5.3 Minimization of complexity (ADV_INT.3)

Applicable to: EAL7

The evaluation requirements of the "Minimization of complexity" component are not applicable to the development of communications standards.

6.4.3.6 Low-level design family (ADV_LLD)

A low-level design is equivalent to a stage 3 specification [10] and should provide a description of the internal workings in terms of physical items of equipment, their interrelationships and dependencies. It should refine the requirements specified in the associated high-level design (stage 2) described in clause 6.4.3.3 by defining the detailed behaviour expected and both the form and content of any data items such as protocol messages.

6.4.3.6.1 Descriptive low-level design (ADV_LLD.1)

Applicable to: EAL4 and EAL5

To achieve the evaluation level set by the "Descriptive low-level design" component a protocol standard (stage 3) may be specified entirely informally using text and drawings. SDL process charts and MSCs produced with a drawing tool (as opposed to an SDL or MSC tool) would be considered to be informal in this context. Significant normal and exceptional behaviour as well as the messages associated with important error conditions should be specified as appropriate. The parts of the standard which specify security-related aspects should be clearly identified.

6.4.3.6.2 Semiformal low-level design (ADV_LLD.2)

Applicable to: EAL6

Evaluation to the level set by the "Semiformal low-level design" component requires the same evidence identified in ADV_LLD.1 (see clause 6.4.3.6.1) but the specification itself should be presented in a semiformal language supported by explanatory text and should include details of all normal and exceptional behaviour as well as the messages associated with all error conditions. The use of semiformal languages such as SDL and UML should be supported by software tools capable of checking both syntax and semantics. These tools should be used to develop complete models of the specification which can be validated by simulation. Guidelines on the use of these languages for this purpose can be found in EG 201 383 [7], EG 201 872 [8] and EG 202 106 [9] and on ETSI's "Making Better Standards" website at http://portal.etsi.org/mbs.

6.4.3.6.3 Formal low-level design (ADV_LLD.3)

Applicable to: EAL7

Evaluation to the level set by the "Formal low-level design" component requires the same evidence identified in ADV_LLD.2 (see clause 6.4.3.6.2) with the exception that the specification should be based on formal design languages such as LOTOS and Z. As stated in clause 6.4.3.2.4, such languages are rarely used in communications standards.

6.4.3.7 Representation correspondence family (ADV_RCR)

The "Representation correspondence" family of components addresses verification of the consistency between the different levels of specification and design. This includes the design correlation between a stage 2 specification and the corresponding stage 1 and between a stage 3 standard and the corresponding stage 2.

6.4.3.7.1 Informal correspondence demonstration (ADV_RCR.1)

Applicable to: EAL1 to EAL4

To achieve the evaluation level set by the "Informal correspondence demonstration" component an informal analysis should demonstrate that all functionality specified in a stage 1 (or equivalent) is correctly and completely refined in the corresponding stage 2 and that a similar correspondence exists between the stage 2 and stage 3. The analysis can take the form of a series of design reviews where particular attention was placed on the correlation between the different levels of specification. Guidelines on conducting a design review (walk-through) can be found on ETSI's "Making Better Standards" web site at http://portal.etsi.org/mbs/Validation/walk_through.asp.

6.4.3.7.2 Semiformal correspondence demonstration (ADV_RCR.2)

Applicable to: EAL5 and EAL6

Evaluation to the level set by the "Semiformal correspondence demonstration" component requires the same evidence identified in ADV_RCR.1 (see clause 6.4.3.7.1) except that the correspondence is required to be demonstrated using semiformal methods. Any protocol or service specification which uses UML to define functional requirements, information flows and physical behaviour (i.e., throughout all three stages) will be able to show that the required correspondence is intrinsic. It is likely that the correlation between semiformal specifications which are not refined from a common source using a single software-development tool can only be established by design reviews as described in clause 6.4.3.7.1.

6.4.3.7.3 Formal correspondence demonstration (ADV_RCR.3)

Applicable to: EAL7

Evaluation to the level set by the "Formal correspondence demonstration" component requires the same evidence identified in ADV_RCR.2 (see clause 6.4.3.7.2) with the exception that correspondence should be established using formal design methods based on languages such as LOTOS and Z. As stated in clause 6.4.3.2.4, such languages are rarely used in communications standards.

6.4.3.8 Security policy modelling family (ADV_SPM)

It is the objective of the "Security policy modelling" family to provide additional assurance that the security functions in the functional specification enforce the security policies. Communications standards provide general sets of requirements which can be applied in a wide range of specific applications and, consequently, do not define security policies. Instead, they provide a range of security functions, if security is likely to be applicable, and leave it to the users of products which conform to the standards to select the functions which implement their own security policies best. The modelling of security policies is, therefore, inapplicable to the standards-making process.

6.5 Guidance documents

6.5.1 Class description

The evaluation class, "Guidance documents" (AGD) specifies requirements for user and administrator documentation. There are 2 families described in this class and these are as follows:

- Administrator guidance (AGD_ADM);
- User guidance (AGD_USR).

In the context of CC evaluation these documents represent the guidance given to administrators to ensure secure administration of the TOE and the guidance given to users to assure secure usage of the TOE is evaluated.

6.5.2 Implications for the standardization process

ETSI does not produce specific administrator or user documents. Consequently, the Guidance documents family has no impact on the standardization process.

6.5.3 Families and components

6.5.3.1 Guidance documents class evaluation levels

The components of each of the "Guidance documents" evaluation families cover activities for the evaluation of administrator guidance and user guidance. The requirements of these components are included in all evaluation assessment levels as summarized in table 6.

Table 6: "Guidance Documents" family evaluation levels

Evaluation component		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Administrator guidance								
AGD_ADM.1	Administrator guidance	✓	✓	✓	✓	✓	✓	✓
User guidance								
AGD_USR.1	User guidance	✓	✓	✓	✓	✓	✓	✓

6.5.3.2 Administrator guidance family (AGD_ADM)

Applicable to: EAL1 to EAL7.

The intention for the developer is to prepare guidance that advises how the TOE is meant to work and to ensure that the administrator cannot be given guidance that makes the system insecure. This is closely tied to the delivery class and thus there exists no impact on the ETSI standardization process.

6.5.3.3 User guidance family (AGD_USR)

Applicable to: EAL1 to EAL7.

The intention for the developer is to prepare guidance that advises how the TOE is meant to work and to ensure that the user cannot be given guidance that makes the system insecure.

In terms of the ETSI standardization process this family does not apply. However the set of guidance documents identified in table 8 are intending to give guidance to the developer of standards over the product lifecycle.

6.6 Life cycle support

6.6.1 Class description

The Life Cycle Support class specifies requirements which are intended to establish discipline and control within the overall process of security product management from inception to withdrawal. The following activities are included in this class:

- the management of security during the development phase:
  - physical;
  - procedural;
  - personnel.
- the implementation of an effective change control system during the maintenance phase;
- the preparation of a written specification of the life cycle itself;
- the specification of methods and the selection of automated tools for use in the development and testing of security functions.

6.6.2 Implications for the standardization process

The development of communications standards is generally undertaken by voluntary resource which cannot be controlled to the same degree as employed staff. Consequently, although some controls are implemented within the standards-making process, it is almost impossible to meet the full requirements of the Life cycle support class. For this reason, the following clauses briefly describe the activities associated with life cycle support within ETSI but do not attempt to show how the various levels of evaluation assurance can be met.

6.6.3 Families and components

6.6.3.1 Life cycle support class evaluation levels

The components of each of the "Life cycle support" evaluation families cover a wide range of activities from the specification of a life-cycle model through to the collection and processing of product fault reports. The relationship between meeting the requirements of these components and achieving particular evaluation assessment levels is summarized in table 7.

Table 7: "Life cycle support" family evaluation levels

Evaluation component		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Development security								
ALC_DVS.1	Identification of security measures			✓	✓	✓		
ALC_DVS.2	Sufficiency of security measures						✓	✓
Flaw remediation								
ALC_FLR.1	Basic flaw remediation							
ALC_FLR.2	Flaw reporting procedures							
ALC_FLR.3	Systematic flaw remediation							
Life cycle definition								
ALC_LCD.1	Develop defined life-cycle model				✓			
ALC_LCD.2	Standardized life-cycle model					✓	✓	
ALC_LCD.3	Measurable life-cycle model							✓
Tools and techniques								
ALC_TAT.1	Well-defined development tools				✓			
ALC_TAT.2	Compliance with implementation standards					✓		
ALC_TAT.3	Compliance with implementation standards – all parts						✓	✓

6.6.3.2 Development security (ALC_DVS)

6.6.3.2.1 Family description

Assessment of the development security family of components is intended to establish that the TOE development environment is, itself, secure from potential threats. Evaluation considers measures related to:

- the physical security of the development site(s);
- procedures related to the protection of secure information;
- the selection of development personnel.

Within the ETSI development environment, the only standards containing information which requires protection from threats are those that specify algorithms for encryption and authentication. Such standards are handled by the Security Algorithms Group of Experts (SAGE) which has a controlled membership and which operates within the secure environment provided by its members.

6.6.3.3 Flaw remediation (ALC_FLR)

6.6.3.3.1 Family description

ISO/IEC 15408-3 [19] does not identify flaw remediation as mandatory for any EAL. Nevertheless, there are a number of approaches taken to identifying and correcting mistakes in communications standards. When an error (technical or editorial) is identified in a published standard, a new edition can be prepared. The method used for the reporting and management of errors in published security-related standards should follow a defined process. An example of such a process can be found on the ETSI PTCC web site at http://www.etsi.org/ptcc/TTCN-3%20CR/ptccttcn3cr.htm.

6.6.3.4 Life cycle definition (ALC_LCD)

The life cycle definition family of components require a developer to manage the specification, design, development and maintenance of a TOE according to a defined or even standardized life cycle model. Most such models are based on a series of phases which typically include:

- strategy and planning;
- requirements gathering and analysis;
- product definition;
- product development;
- product launch;
- product design maintenance;
- product withdrawal.

There is no single definition of the standards development life cycle but, as shown in table 8, many of the phases are described in published recommendations, guides and reports.

Table 8: Standards development life cycle documentation

Life cycle phase	Method guidance
strategy and planning	ETSI Directives
requirements gathering and analysis	ITU-T Recommendation I.130 [10], EG 201 872 [8]
product definition	EG 201 872 [8], EG 202 107 [13], ITU-T Recommendation I.130 [10]
product development	EG 201 872 [8], EG 202 107 [13], ITU-T Recommendation I.130 [10]
product launch	ETSI Directives
product design maintenance	(see note)
product withdrawal	ETSI Directives
NOTE: Whilst ETSI does not provide explicit guidance on methods for the maintenance of standards, the TTCN-3 Change Request process described at http://www.etsi.org/ptcc/TTCN-3%20CR/ptccttcn3cr.htm may be useful as the basis for other similar maintenance processes.	

6.6.3.5 Tools and techniques (ALC_TAT)

6.6.3.5.1 Family description

The components of the tools and techniques family are intended to assure that appropriate automatic tools and development methodologies are used and that their selection is based on careful analysis of the related requirements. Methodologies and Tools supporting the use of specification languages such as SDL and UML are used throughout the requirements analysis, product definition and product development phases of the standards-making process. Details of the guidance documentation that is available for these methods and tools can be found in clauses 6.4 and 6.7.

6.7 Tests

6.7.1 Class description

The evaluation class, "Tests" specifies requirements for the testing of a TSF as part of its overall development. There are 4 families described in this class and these are as follows:

- Coverage (ATE_COV);
- Depth (ATE_DEP);
- Functional tests (ATE_FUN);
- Independent testing (ATE_IND).

In the context of CC evaluation, the purpose of testing activities is to confirm that the TSF operates according to its specification. These activities includes both positive testing to ensure that the TSF meets its functional requirements and negative testing to ensure that the TSF displays no undesirable behaviour.

6.7.2 Implications for the standardization process

Although communications standards usually specify the characteristics or behaviour to be found in products implementing the standards, they rarely specify how those characteristics and behaviour are to be realized in the product. Consequently, it may not be possible for a standard to meet all of the evaluation requirements specified in ISO/IEC 15408-3 [19].

Communications standards containing security-related requirements generally fall into two groups; those that specify cryptographic and authentication algorithms and those that specify protocols. In both these cases it is general practice to produce a formal specification of the tests required to establish conformance. The production of such test specifications is an essential activity to support implementations in achieving successful CC evaluation even at EAL1.

As communications security standards are considered to be Protection Profiles in the context of CC, the ATE class has a dual impact on the way that such standards are developed, thus:

- the means of validating the standards themselves will be subject to evaluation and efforts should be made to ensure that the requirements for achieving EAL5 or EAL6 for class ATE are met;

NOTE: It is unlikely that a standard could meet the requirements of EAL7, particularly in the ATE_DPT family (see clause 6.7.3.3) where the "Testing: implementation representation" component (ATE_DPT.3 in clause 6.7.3.3.3), which relates primarily to TSF implementation, can only be mapped to the standardization process artificially.

- the formal test suites developed for protocol and algorithm standards are ideal tools to be used by implementors and evaluators in the achievement and assessment of assurance levels because they define a range of formal and, therefore, repeatable tests.

6.7.3 Families and components

6.7.3.1 Tests class evaluation levels

The components of each of the "Tests" evaluation families cover a wide range of activities from simple developer testing right through to comprehensive independent testing. The relationship between meeting the requirements of these components and achieving particular evaluation assessment levels is summarized in table 9.

Table 9: "Tests" family evaluation levels

Evaluation component		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Coverage								
ATE_COV.1	Evidence of coverage		✓					
ATE_COV.2	Analysis of coverage			✓	✓	✓		
ATE_COV.3	Rigorous analysis of coverage						✓	✓
Depth								
ATE_DPT.1	Testing: high-level design			✓	✓			
ATE_DPT.2	Testing: low-level design					✓	✓	
ATE_DPT.3	Testing: implementation representation							✓
Functional tests								
ATE_FUN.1	Functional testing		✓	✓	✓	✓		
ATE_FUN.2	Ordered functional testing						✓	✓
Independent testing								
ATE_IND.1	Independent testing – conformance	✓						
ATE_IND.2	Independent testing – sample		✓	✓	✓	✓	✓	
ATE_IND.3	Independent testing – complete							✓

6.7.3.2 Coverage family (ATE_COV)

Assessment of test coverage is intended to provide an indication of the extent to which a TSF has been tested. An evaluation of a standard as a PP will establish whether validation methods have been used to ensure that the standard meets the functional requirements specified for it.

6.7.3.2.1 Evidence of coverage (ATE_COV.1)

Applicable to: EAL2

To achieve the evaluation level set by the "Evidence of coverage" component, the security-related requirements defined in a standard specifying behaviour (for example, a protocol stage 3 standard) should be reviewed against the associated functional specification (stage 1 and stage 2 standards, for example) to ensure that each of the security functional requirements have been fully realized in the detailed specification (see also clause 6.4.3.7). Such a review should be carried out and documented according to the "walk-through" method described in EG 202 107 [13].

6.7.3.2.2 Analysis of coverage (ATE_COV.2)

Applicable to: EAL3, EAL4 and EAL5

To achieve the evaluation level set by the "Analysis of coverage" component it is first necessary to meet the requirements of component ATE_COV.1 (see clause 6.7.3.2.1). Additionally, the specified behaviour should be formally modelled using a specification language such as SDL or UML. Once modelled, all functions associated with the provision of security (which may include some which are not explicitly security functions) should be simulated using software tools to demonstrate the expected normal (non-error) behaviour and a representative sample of error cases. The use of simulation techniques to validate formal models is described in EG 201 015 [11], EG 202 107 [13], ETR 184 [14] and on the "Making Better Standards" web site (http://portal.etsi.org/mbs/Validation/FormalMethods/simultation.asp).

The output of a simulation exercise should be a graphical or textual report of the actual behaviour. This report could, for example, be in the form of a set of UML Sequence Diagrams or Message Sequence Charts (MSC) as shown in figure 11. The report of observed behaviour should be annotated to indicate where in the report each of the functional requirements is tested.

It is unlikely that a purely textual specification of security functions could meet the requirements of ATE.COV.2.

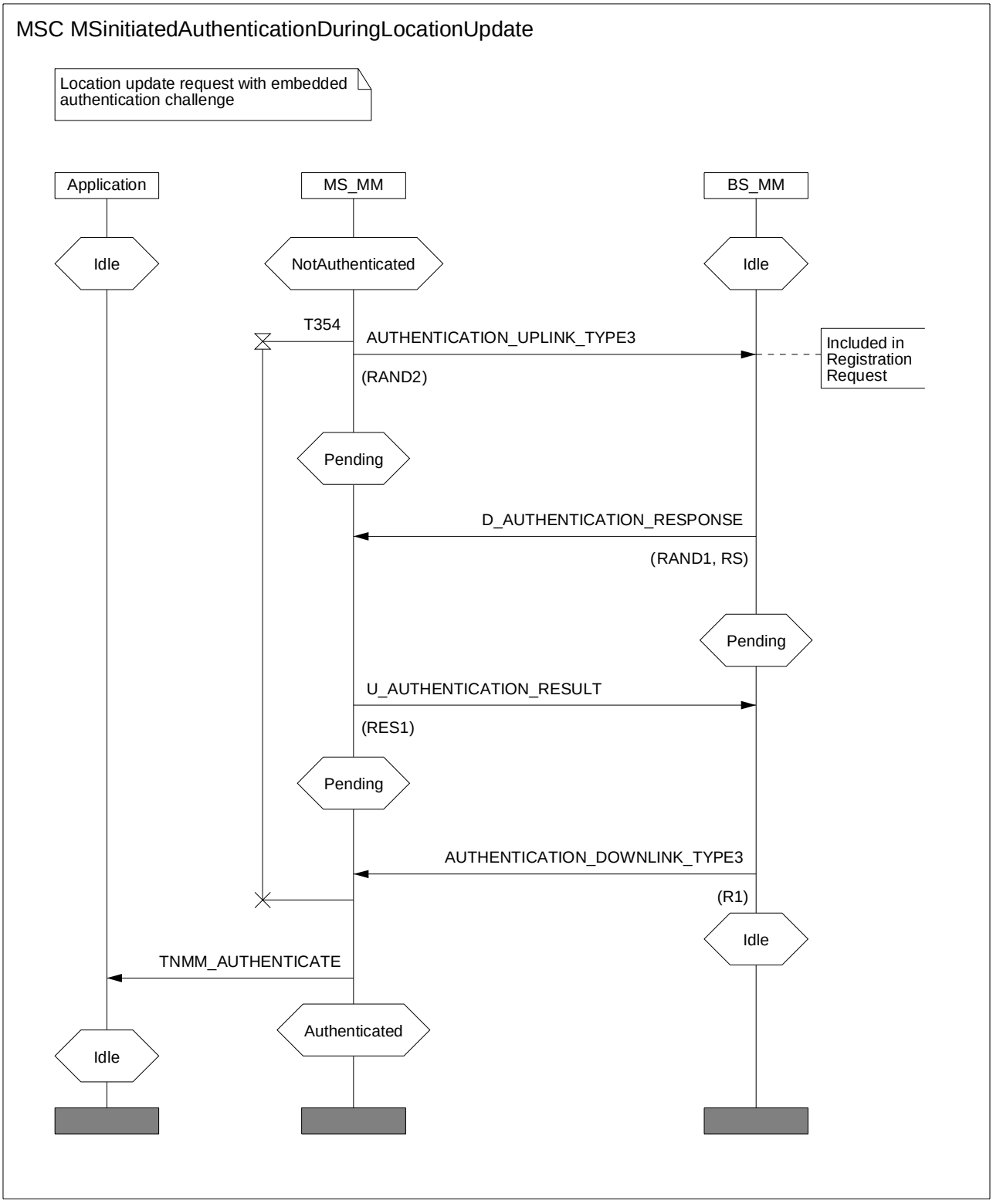


Figure 11: Example MSC

6.7.3.2.3 Rigorous analysis of coverage (ATE_COV.3)

Applicable to: EAL6 and EAL7

The "Rigorous analysis of coverage" component requires the same type of testing as ATE_COV.2 (see clause 6.7.3.2.2) but the scenarios tested need to include the full range of exceptional behaviour.

6.7.3.3 Depth family (ATE_DPT)

The evaluation of the depth of testing is intended to determine whether sufficient and appropriate testing has been carried out to ensure that the internal components of a TOE (related to both security functions and non-security functions) operate correctly. Such testing should detect malicious code, if it exists, and identify other vulnerabilities so that they may be countered.

6.7.3.3.1 Testing: high-level design (ATE_DPT.1)

Applicable to: EAL3 and EAL4

It is not possible to review a standard at a high-level of design as the documents are, by their nature, monolithic. Evaluation to the low-level design component (ATE_DPT.2) should, therefore, be expected for EAL3 and EAL4.

6.7.3.3.2 Testing: low-level design (ATE_DPT.2)

Applicable to: EAL5 and EAL6

The detailed content of all standards should be reviewed using the structured walk-through method described in EG 202 107 [13] to ensure that:

- the objectives set for the standard have been met in full by the specified requirements;
- the requirements expressed in the standard are consistent with each other;
- the requirements expressed in the standard are consistent those specified in other related standards;
- where requirements are expressed in more than one format (for example, textually and graphically) the different forms do not conflict (see also clause 6.4.3.7).

A walk-through is no more complex than the review of a draft standard within a technical committee or working group. However, it is important to ensure that the following aspects are taken into account when organizing such a review:

- the attendees are collectively able to assess the correctness of the whole standard under review;
- the standard is reviewed extensively and completely.

Notes are taken during the review to indicate what changes are required to the reviewed text as well as other significant decisions and actions.

6.7.3.3.3 Testing: implementation representation (APT_DPT.3)

Applicable to: EAL7

A standard that has used formal specification methods to describe a protocol can be validated using automatic tools to analyse the structure and integrity of the specification. Validation based on state-space exploration is described in EG 201 015 [11] and can identify vulnerable areas of the design such as:

- implicit consumption of signals:

  signals that can arrive in a state where there is no processing specified for such a signal;

- unreachable code:

  areas of the specification which cannot be reached as a result of any explicit stimulus;

- deadlocks and live-locks:

  areas of the specification where processing will stall or cease altogether.

If automatic validation tools are not available, a formal specification can also be validated by means of a structured walk-through (see EG 202 107 [13]) but this is time-consuming, labour-intensive and less accurate than the tool-based approach.

It is unlikely that a purely textual specification of security functions could meet the requirements of this component.

6.7.3.4 Functional tests family (ATE_FUN)

Evaluation of functional testing is primarily concerned with the approach taken to testing rather than what is tested. It specifies requirements for the presentation of all test plans, procedures and results. The range of functions tested and the rigor with which they are tested are evaluated within the ATE_COV (see clause 6.7.3.2) and ATE_DPT (see clause 6.7.3.3) families of components.

As its name implies, functional testing can only be applicable to standards which specify behaviour. Such standards include specifications of protocols, services and physical interfaces. For a standard to be functionally tested, it is necessary that its required functions are modelled in:

- a formal specification using a language such as SDL or UML; or
- a physical prototype.

Guidance on the use of formal modelling and prototyping methods can be found in EG 201 015 [11], EG 202 107 [13], ETR 184 [14] and on the "Making Better Standards" web site which can be found at http://portal.etsi.org/mbs/Validation/validationMethods.asp.

6.7.3.4.1 Functional testing (ATE_FUN.1)

Applicable to: EAL2, EAL3, EAL4 and EAL5

Successful assessment to the ATE_FUN.1 component requires considerable planning and effort prior to and during the validation of a security-related standard. The following documentation should be produced for each standard:

- a test plan identifying:
  - the individual security functional requirements to be validated;
  - an objective for the validation of each of these requirements;
- a test procedure identifying:
  - the individual validation activities that should be performed;
  - the stimuli and parameter values that are to be used to initiate each validation test;
- expected test results.

Once the standard has been validated, the results should be documented, comparing the actual results with the expected results in order to demonstrate that each security functional requirement is met. EG 202 107 [13] provides guidance on planning for validation activities.

Although ATE_FUN.1 expects tests to be carried out in a logical and ordered fashion, it does not require that this ordering is based on formal documented analysis and so there is no guarantee that the sequence of tests can be exactly repeated.

6.7.3.4.2 Ordered functional testing (ATE_FUN.2)

Applicable to: EAL6 and EAL7

The evaluation requirements for the ATE_FUN.2 component are identical to those for ATE_FUN.1 (see clause 6.7.3.4.1) except that validation activities should be ordered on the basis of a thorough, documented analysis of the dependency relationships between testing tasks (i.e., which tasks must have been successfully completed before starting the current test).

6.7.3.5 Independent testing (ATE_IND)

The evaluation of the use independent testing as part of the development process is intended to demonstrate the degree to which a third party (other than the developer) has been able to test the functional capabilities of a TOE.

Although often carried out by an external laboratory or test-house, independent testing in this context is always performed on behalf of the evaluator rather than the developer and the tests performed are chosen and specified by the evaluator. The nature of the standardization process makes independent testing of a PP derived from a standard difficult unless a formal model or prototype is developed for evaluation in the functional testing family (see clause6.7.3.4). However, independent testing is an important part of the evaluation of an ST implementing the security requirements expressed in a standard.

6.7.3.5.1 Independent testing - conformance (ATE_IND.1)

Applicable to: EAL1

Evaluation to the ATE_IND.1 component requires only that sufficient information is provided to enable an independent assessor to devise and perform a series of tests to ensure that the TOE security functional requirements are met. There is no requirement for the review of any test results obtained during the development of the TOE.

6.7.3.5.2 Independent testing - sample (ATE_IND.2)

Applicable to: EAL2, EAL3, EAL4, EAL5 and EAL6

In addition to the requirements of ATE_IND.1 (see clause 6.7.3.5.1), evaluation to ATE_IND.2 will include repeating a sample of tests already performed during development of the TOE. The selection of tests to be included in this sample will be made by the independent assessor who will expect full development test results to be made available.

6.7.3.5.3 Independent testing - complete (ATE_IND.3)

Applicable to: EAL7

The evaluation requirements for the ATE_IND.3 component are the same as those for ATE_IND.2 (see clause 6.7.3.5.2) except that the independent assessor will repeat the full range of tests performed during the development of the TOE.

6.8 Vulnerability assessment

6.8.1 Class description

The evaluation class, "Vulnerability assessment" (AVA) specifies requirements for identifying weaknesses that could be exploited in a TOE design. There are 4 families described in this class and these are as follows:

- Covert channel analysis (AVA_CCA);
- Misuse (AVA_MSU);
- Strength of TOE security functions (AVA_SOF);
- Vulnerability analysis (AVA_VLA).

In the context of CC evaluation the purpose of this class is to ensure that appropriate steps have been taken within the TOE development process to locate any of the following potential vulnerabilities which could be exploited by threat agents:

- exploitable covert channels;
- the possibility of misuse or insecure configuration of the TOE;
- the resilience of statistical security mechanisms;
- design weaknesses.

6.8.2 Implications for the standardization process

It is recognized that without an understanding of the threats to the system that appropriate selection of countermeasures cannot be made. Within ETSI there is a general guideline on the preparation of a threat analysis published as ETR 332 [6] which identifies risk to the system based upon the product of the likelihood of an attack, and the impact that such an attack will have on the system. The purpose of ETR 332 [6] is to perform a cost-benefit-analysis on the system when viewed by an attacker. The structure of the assurance class for vulnerability analysis is slightly different as it does not address the impact of an attack on the system but addresses the resistance to attack of the system.

Within the context of a PP (see ES 202 382 [2]) the bulk of the descriptive text will be derived from the system threat analysis document: Security objectives; Security requirements; Rationale. It is also shown in clause 4.3 of the present document that a vulnerability analysis is required throughout the specification process.

The depth of the vulnerability analysis changes as the system design becomes more detailed. A vulnerability analysis working from the system objectives will identify at a very coarse level the required security functionality to ensure that the objectives can be met without damage to the system. The vulnerability analysis assurance family seems to assume that the system design is complete whereas the purpose of the vulnerability analysis exercise in ETSI is to be able to identify vulnerabilities that require the provision of countermeasures, and then to assess the vulnerabilities that exist in the system with the countermeasures applied.

6.8.3 Families and components

6.8.3.1 Vulnerability assessment class evaluation levels

The components of each of the "Vulnerability assessment" evaluation families cover a wide range of activities from assessing the possibility that control information could enter or leave the system undetected through to an analysis of the weaknesses in the TOE design. The relationship between meeting the requirements of these components and achieving particular evaluation assessment levels is summarized in table 10.

Table 10: "Vulnerability assessment" family evaluation levels

Evaluation component		EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
Covert channel analysis								
AVA_CCA.1	Covert channel analysis					✓		
AVA_CCA.2	Systematic covert channel analysis						✓	✓
AVA_CCA.3	Exhaustive covert channel analysis							
Misuse								
AVA_MSU.1	Examination of guidance			✓				
AVA_MSU.2	Validation of analysis				✓	✓		
AVA_MSU.3	Analysis and testing for in secure states						✓	✓
Strength of TOE security functions								
AVA_SOF.1	Strength of TOE security functional evaluation		✓	✓	✓	✓	✓	✓
Vulnerability analysis								
AVA_VLA.1	Developer vulnerability analysis		✓	✓				
AVA_VLA.2	Independent vulnerability analysis				✓			
AVA_VLA.3	Moderately resistant					✓		
AVA_VLA.4	Highly resistant						✓	✓

6.8.3.2 Covert channel analysis family (AVA_CCA)

The purpose of this family in the context of vulnerability analysis is to detect specific weaknesses in the design of the system that allow unintended signalling channels to propagate. The application notes in ISO/IEC 15408-3 [19] indicate that this assurance class only applies when information flow control security function policies are present in the ST. However notwithstanding the application note, and noting that ETSI works primarily in the PP domain rather than the ST domain, the design goal should be to ensure that the PP (and hence any ST claiming to be PP conformant) does not allow for the introduction of covert control channels.

6.8.3.2.1 Covert channel analysis

Applicable to: EAL5

The objective of covert channel analysis is to identify possible covert channels in a TOE design and to estimate the capacity of each. This means that the designer of an information channel should ensure that only those signals intended for that channel are accepted and processed. At this level of evaluation, an analysis only needs to make use of informal searching methods such as walk-through [13] and visual inspection.

To avoid creating a potential covert channel, the state machine associated with a standardized protocol should specify general exceptional behaviour including the capture of unexpected signals.

Abstract Syntax Notation 1 (ASN.1) is frequently used to define the structure of standardized protocol messages. Compatibility between versions is achieved by means of an extension marker which allows all information following the marker (e.g. Version 2) to be ignore by an earlier version (e.g. Version 1). Such a mechanism could easily be exploited as a covert channel and should be avoided if possible.

6.8.3.2.2 Systematic covert channel analysis

Applicable to: EAL6

The evaluation requirements for systematic covert channel analysis are almost identical to those specified for covert channel analysis (see clause 6.8.3.2.1) with the exception that the analysis is expected to be systematic rather than informal. ISO/IEC 15408-3 [19] gives no guidance on the meaning of the term "systematic" other than to say that covert channels should be identified in a structured and repeatable way rather than in an ad-hoc fashion. An example of a systematic approach to covert channel analysis would be the automatic processing of all ASN.1 code (with a text editor) to locate the use of extension markers.

6.8.3.2.3 Exhaustive covert channel analysis

Applicable to: EAL7

The evaluation requirements for systematic covert channel analysis are almost identical to those specified for covert channel analysis (see clause 6.8.3.2.1) with the exception that the analysis is expected to be both systematic and exhaustive. Consequently, there should be a high probability that the analysis identifies all possible covert channels.

6.8.3.3 Misuse family (AVA_MSU)

This family is used to determine if the TOE can be configured to be insecure where the user or administrator would reasonably believe it to be secure. In this respect it is a test of the guidance given to the administrator or user in documentation (as defined by family Guidance documents (see clause 6.3.4)).

In general ETSI does not provide standards for user or administrator guidance over and above the outline specifications. Communications standards are not generally supplied with user or administrator manuals. However, guidance is given to implementors of the standards in the form of a Protocol Implementation Conformance Statement (PICS) which summarizes the requirements of the base standard and identifies permissible combinations of options. It is the responsibility of the standard writer to ensure that none of these legitimate combinations fail to meet the security objectives of the system. A brief description of a PICS can be found in annex B.

6.8.3.3.1 Strength of TOE security functions family (AVA_SOF)

NOTE: In the updates to ISO/IEC-15408-3 [19] being undertaken during the period 2003-2005 this element has been taken out of class AVA and therefore may not appear in any update of ISO/IEC-15408-3 [19].

The purpose of the AVA_SOF family is to determine if the strength of the TOE security function is sufficient to counter the threats or attacks it is intended to deter.

6.8.3.3.2 Strength of TOE security function evaluation

There is an underlying assumption that all security functions are breakable in some form. The requirement placed on the developer is to identify both the means to measure the strength of a function and to evaluate the actual strength of the security function compared to the requirement.

6.8.3.4 Vulnerability analysis family (AVA_VLA)

NOTE: The component levelling in this family suggests a linear levelling, however it may also be viewed that components 3 and 4 have a dependency on components 1 and 2 whereby the levelling may not be linear.

The purpose of the vulnerability analysis is to determine how open to attack the system, or components of the system are. One method of addressing the attack potential is to consider a number of factors that will enable the attack as shown in tables 11 and 12.

Table 11: Attack potential

Factor	Range	Value
Time to mount the attack (1 point per week)	≤1 day	0
	≤ 1 week	1
	≤ 1 month	4
	≤ 3 months	13
	≤ 6 months	26
	> 6 months	See note 1
Expertise	Layman	0
	Proficient	2
	Expert	5
Knowledge of TOE	Public	0
	Restricted	1
	Sensitive	4
	Critical	10
Access to launch the attack	Unnecessary / unlimited access	0
	Easy	1
	Moderate	4
	Difficult	12
	None	See note 2
Equipment	Standard	0
	Specialized	3
	Bespoke	7
NOTE 1: Attack potential is beyond high. NOTE 2: Attack path is not exploitable.		

Each of these attack factors are summed (i.e. Elapsed time + Expertise + Knowledge of TOE + Window of opportunity + Equipment) to give an overall vulnerability rating as shown in table 12. The vulnerability rating is then mapped to the Occurrence likelihood as shown in table 13.

Table 12: Vulnerability rating

Range of values	Resistant to attacker with attack potential of:
0 to 2	No rating
3 to 6	Basic
7 to 14	Moderate
15 to 26	High
> 26	Beyond high

The method for threat analysis defined in ETR 332 [6] combines the likelihood with the impact of the attack in determining if a countermeasure should be applied. The form of countermeasures can include redesign of the at risk element in the system to remove the vulnerability that is to be attacked, and application of a defensive system component that masks the vulnerability.

Table 13: Mapping of vulnerability rating to likelihood

Vulnerability rating	Likelihood
Beyond high	Unlikely
High	
Moderate	Possible
Basic	Likely
No rating	
NOTE: Motivation is not considered explicitly in the vulnerability rating.	

6.8.3.4.1 Developer vulnerability analysis

Applies from EAL2

The objective here is to ensure the developer has ascertained the presence of obvious vulnerabilities in the system and to confirm that they cannot be exploited.

6.8.3.4.2 Independent vulnerability analysis

Applies from EAL4

6.8.3.4.3 Moderately resistant

Applies from EAL5

The objective in this group is to be able to show that all threats where the attack potential falls in class "moderate" cannot be exploited. The residual impact is that only those threats where the attacker has high attack potential remain in the system (i.e. only attacks requiring a combination of expert knowledge, specialized hardware and good access to the system will succeed).

6.8.3.4.4 Highly resistant

Applies from EAL6

The objective in this group is to be able to show that all threats where the attack potential falls in class "high" cannot be exploited. The residual impact is that only those threats where the attacker has infeasible attack potential remain in the system (i.e. no reasonable (evaluated) attack will succeed).

6.9 Maintenance of assurance

6.9.1 Class description

The evaluation class, "Maintenance of Assurance" (AMA) is intended to be applied after evaluation and certification in order to give confidence that the TOE continues to meet its security goals in the face of changes to the environment and the TOE. Whilst the simplest method of maintenance may appear to be re-evaluation this may be impractical and this class, which is not required as part of any EAL, is intended to identify alternatives methods of assurance maintenance to re-evaluation.

There are 4 families described in this class and these are as follows:

- Assurance maintenance plan (AMA_AMP);
- TOE component categorization report (AMA_CAT);
- Evidence of assurance maintenance (AMA_EVD);
- Security impact analysis (AMA_SIA).

6.9.2 Implications for the standardization process

ETSI standards are maintained in the market and evolve over time. In clauses 6.8 (Life Cycle Support), 6.6 (Development) and 6.7 (Test) methods are identified in support of the standards development process that maintain the integrity of the standard against its security objectives. When the advice given in the preceding clauses is followed the major impact of the AMA class is in ensuring that the user of the TOE is party to the maintenance of the standard and has an assurance maintenance plan in place.

Annex A (normative): Functional components in ISO/IEC-15408-2 [18]

A.1 Introduction

When preparing a security document for evaluation (either PP or ST) the developer is requested to identify security functionality from the range of components defined in ISO/IEC 15408-2 [18]. This annex reviews these components and identifies additional levels of standardization that need to be provided to satisfy the evidential proof of their operation.

NOTE: ISO/IEC-15408 [20] refers to aspects of both assurance and functionality as "classes". These should not be confused with the same term commonly used in Object Oriented Design and Analysis.

ISO/IEC 15408-2 [18] identifies a set of functional components which cover the major elements of any security product or process and these are defined in the following classes (ISO/IEC 15408-2 [18] component name in brackets):

- Security audit (FAU);
- Communication (FCO);
- Cryptographic support (FCS);
- User data protection (FDP);
- Identification and authentication (FIA);
- Security management (FMT);
- Privacy (FPR);
- Protection of the Target of Evaluation Security Functions (FPT);
- Resource utilization (FRU);
- Target of Evaluation access (FTA);
- Trusted path/channels (FTP).

The components can be used in the development of requirements at both an abstract level and at the detail development level.

The developer needs to be aware of the functional components and to report their use.

EXAMPLE: A countermeasure to prevent masquerade may require that the identity is presented and validated, then authenticated, prior to system access. To implement this countermeasure will require a design that includes components "User identification before action" and "User authentication before action" (FIA_UID.2 and FIA_UAU.2 respectively in ISO/IEC-15408-2 [18].

A.2 Security audit

Functional class "Security audit " (FAU) is required to ensure that information related to security relevant activities is recognized, recorded, stored, and analysed. It is also required to assure that the resulting audit records can be examined to determine which security relevant activities took place and whom (which user) is responsible for them. The class provides 6 families of behaviour, thus:

- Security audit automatic response (FAU_ARP):
  - Security alarms (FAU_ARP.1)
    actions are taken in case of a detected potential security violation.

- Security audit data generation (FAU_GEN):
  - Audit data generation (FAU_GEN.1)
    the level of auditable events are defined and the list of data that shall be recorded in each record is specified;
  - User identity association (FAU_GEN.2)
    auditable events are associated to individual user identities.
- Security audit analysis (FAU_SAA):
  - Potential violation analysis (FAU_SAA.1)
    basic threshold detection on the basis of a fixed rule set is required;
  - Profile based anomaly detection (FAU_SAA.2)
    individual profiles of system usage are maintained, where a profile represents the historical patterns of usage performed by members of the profile target group and a profile target group refers to a group of one or more individuals;
  - Simple attack heuristics (FAU_SAA.3)
    the occurrence of signature events that represent a significant threat to TSP enforcement is detected;
  - Complex attack heuristics (FAU_SAA.4)
    the ability to compare system events against event sequences known to represent entire intrusion scenarios and the ability to indicate when an event sequence is found that indicates a potential violation of the TSP.
- Security audit review (FAU_SAR):
  - Audit review (FAU_SAR.1)
    provides the capability to read information from the audit records;
  - Restricted audit review (FAU_SAR.2)
    requires that there are no other users except those that have been identified that can read the information;
  - Selectable audit review (FAU_SAR.3)
    requires audit review tools to select the audit data to be reviewed based on criteria.
- Security audit event selection (FAU_SEL):
  - Selective audit (FAU_SEL.1)
    requires the ability to include or exclude events from the set of audited events based upon attributes to be specified.
- Security audit event storage (FAU_STG):
  - Protected audit trail storage (FAU_STG.1)
    requirements are placed on the audit trail to protect it from unauthorized deletion and/or modification;
  - Guarantees of audit data availability (FAU_STG.2)
    guarantees that even given the occurrence of an undesired condition the audit data are maintained;
  - Action in case of possible audit data loss (FAU_STG.3)
    specifies actions to be taken if a threshold on the audit trail is exceeded;
  - Prevention of audit data loss (FAU_STG.4)
    specifies actions in case the audit trail is full.

A.3 Communication

Functional class "Communication" (FCO) is required to ensure that the originator of a message cannot deny having sent it (proof of origin) and the receiver of a message cannot deny having received it (proof of receipt). The class provides 2 families of behaviour, thus:

- Non-repudiation of origin (FCO_NRO):
  - Selective proof (FCO_NRO.1)
    evidence of the origin of information is provided on request;
  - Enforced proof (FCO_NRO.2)
    evidence of the origin of information is provided in every case.
- Non-repudiation of receipt (FCO_NRR):
  - Selective proof (FCO_NRR.1)
    evidence of the receipt of information is provided on request;
  - Enforced proof (FCO_NRR.2)
    evidence of the receipt of information is provided in every case.

A.4 Cryptographic support

Functional class "Cryptographic support" (FCS) is required to provide essential management capabilities and to support the general use of cryptographic keys. The class provides 2 families of behaviour, thus:

- Cryptographic key management (FCS_CKM):
  - Cryptographic key generation (FCS_CKM.1)
    cryptographic keys are generated using a specified algorithm and key size;
  - Cryptographic key distribution (FCS_CKM.2)
    the distribution of cryptographic keys is limited to those users and applications that are suitably authorized;
  - Cryptographic key access (FCS_CKM.3)
    access to cryptographic keys is limited to those users and applications that are suitably authorized;
  - Cryptographic key destruction (FCS_CKM.4)
    only suitably authorized users and applications are permitted to order the destruction of cryptographic keys.
- Cryptographic operation (FCS_COP):
  - Cryptographic operation (FCS_COP.1)
    where cryptographic operations are required then the specified algorithms and keys are used.

A.5 User data protection

Functional class "User data protection" (FDP) is required to ensure that user data is protected during use, transport and storage. It is also required to ensure that access to user data is adequately controlled. The class provides 13 families of behaviour, thus:

- Access control policy (FDP_ACC):
  - Subset access control (FDP_ACC.1)
    access control policies are implemented and enforced for a defined subset of the operations within a TOE;

  - Complete access control (FDP_ACC.2)
    access control policies are implemented and enforced for all objects and their associated operations within a TOE.

- Access control functions (FDP_ACF):

  - Security attribute based access control (FDP_ACF.1)
    access to security objects is restricted to users and applications that are suitably authorized.

- Data authentication (FDP_DAU):

  - Basic data authentication (FDP_DAU.1)
    it is possible to guarantee the authenticity of the information content of an object;

  - Data authentication with identity of guarantor (FDP_DAU.2)
    it is possible to guarantee the authenticity of the information content of an object and identify its guarantor.

- Export outside TSF control (FDP_ETC):

  - Export of user data without security attributes (FDP_ETC.1)
    policies are implemented and enforced to control the export of user data from the TSF although the security attributes associated with the user data are not exported;

  - Export of user data with security attributes (FDP_ETC.2)
    policies are implemented and enforced to control the export of user data and associated security attributes from the TSF.

- Information flow control policy (FDP_IFC):

  - Subset information control (FDP_IFC.1)
    information flow control policies are implemented and enforced for a defined subset of the operations within a TOE;

  - Complete information flow control (FDP_IFC.2)
    information flow control policies are implemented and enforced for all objects and their associated operations within a TOE.

- Information flow control functions (FDP_IFF):

  - Simple security attributes (FDP_IFF.1)
    security attributes are derived for and assigned to items of secure information as well as the senders and receivers of such information;

  - Hierarchical security attributes (FDP_IFF.2)
    hierarchical security attributes are derived for and assigned to items of secure information as well as the senders and receivers of such information;

  - Limited illicit information flows (FDP_IFF.3)
    information flow control policies are implemented that detect but do not necessarily eliminate illicit information flows;

  - Partial elimination of illicit information flows (FDP_IFF.4)
    information flow control policies are implemented that detect illicit information flows and eliminate some, but necessarily all of them;

  - No illicit information flows (FDP_IFF.5)
    information flow control policies are implemented that detect and eliminate all illicit information flows;

  - Illicit information flow monitoring (FDP_IFF.6)
    the flow rate of illicit information is monitored so that action can be taken on flows that exceed predefined thresholds.

- Import from outside TSF control (FDP_ITC):
  - Import of user data without security attributes (FDP_ITC.1)
    security attributes are generated for user data which is imported without reliable security attributes of its own. Generating security attributes may mean acquiring them from a trusted path or channel;
  - Import of user data with security attributes (FDP_ITC.2)
    security attributes associated with imported user data are used if these attributes can be considered to be reliable.
- Internal TOE transfer (FDP_ITT):
  - Basic internal transfer protection (FDP_ITT.1)
    access control and information control procedures are used to protect transmitted user data from disclosure, modification and/or loss;
  - Transmission separation by attributes (FDP_ITT.2)
    individual items of user data is transmitted separately if the associated security attributes require such separation;
  - Integrity monitoring (FDP_ITT.3)
    user data transmitted between TOE parts is constantly monitored in order to detect errors in the integrity of the data. The form of monitoring is unrelated to the contents of the security attributes of the user data;
  - Attribute-based integrity monitoring (FDP_ITT.4)
    user data transmitted between TOE parts is constantly monitored in order to detect errors in the integrity of the data. The form of monitoring is dependent on the contents of the security attributes of the user data.
- Residual information protection (FDP_RIP):
  - Subset residual information protection (FDP_RIP.1)
    residual user data from a defined group of applications is unavailable to other applications after the resources associated with the data have been deallocated;
  - Full residual information protection (FDP_RIP.2)
    residual user data from all applications is unavailable to all other applications after the resources associated with the data have been deallocated.
- Rollback (FDP_ROL):
  - Basic rollback (FDP_ROL.1)
    a limited number of operations can be undone without compromising the integrity of any associated user data;
  - Advanced rollback (FDP_ROL.2)
    all operation can be undone without compromising the integrity of any associated user data.
- Stored data integrity (FDP_SDI):
  - Stored data integrity monitoring (FDP_SDI.1)
    stored user data is continuously monitored in order to detect any errors in its integrity;
  - Stored data integrity monitoring and action (FDP_SDI.2)
    stored data is continuously monitored in order to detect any errors in its integrity. In the event that errors are found, predefined actions can be taken.
- Inter-TSF user data confidentiality transfer protection (FDP_UCT):
  - Basic data exchange confidentiality (FDP_UCT.1)
    user data is protected against disclosure while in transit.

- Inter-TSF user data integrity transfer protection (FDP_UIT):

  - Data exchange integrity (FDP_UIT.1)
    any modifications to user data during transmission are detected;

  - Source data exchange Recovery (FDP_UIT.2)
    when modifications to user data are detected during transmission, these are corrected by the receiver with help from the source;

  - Destination data exchange recovery (FDP_UIT.3)
    when modifications to user data are detected during transmission, these are corrected by the receiver without needing help from the source.

A.6 Identification and authentication

Functional class "Identification and authentication" (FIA) is required to ensure that users are associated with valid and meaningful security attributes such as identity, role and integrity level. It provides the basic capabilities for the implementation of access control. The class provides 6 families of behaviour, thus:

- Authentication Failure (FIA_AFL):

  - Authentication failure handling (FIA_AFL.1)
    session establishment is terminated after a predetermined number of unsuccessful attempts by the user to be authenticated and the user account is then disabled.

- User attribute definition (FIA_ATD):

  - User attribute definition (FIA_ATD.1)
    a set of security attributes is established and maintained for each user.

- Specification of secrets (FIA_SOS):

  - Verification of secrets (FIA_SOS.1)
    items of data that are considered to be secret are evaluated to ensure that they comply with predefined constraints related to secrets;

  - TSF generation of secrets (FIA_SOS.2)
    items of data that are considered to be secret are generated using predefined constraints related to secrets.

- User authentication (FIA_UAU):

  - Timing of authentication (FIA_UAU.1)
    some predefined actions by the user are permitted prior to successful authentication of the user;

  - User authentication before any action (FIA_UAU.2)
    the user is not permitted to perform any action prior to successful authentication of the user;

  - Unforgeable authentication (FIA_UAU.3)
    authentication procedures ensure that forged or copied authentication data is detected and excluded from use;

  - Single-use authentication mechanisms (FIA_UAU.4)
    authentication is based on authentication data which is used once and then discarded (e.g. random number pass-key generated uniquely for each access attempt);

  - Multiple authentication mechanisms (FIA_UAU.5)
    more than one authentication mechanism is supported and, for each application requiring authentication, the appropriate mechanism is identified;

  - Re-authentication (FIA_UAU.6)
    under certain predefined circumstances it is possible for the user to be re-authenticated even though successful authentication has already been achieved;

  - Protected authentication feedback (FIA_UAU.7)
    information which might compromise security is not passed back to the user during authentication (e.g. the characters in a password are replaced with a non significant character such as "*").

- User identification (FIA_UID):

  - Timing of identification (FIA_UID.1)
    some predefined actions by the user are permitted prior to successful identification of the user;

  - User identification before any action (FIA_UID.2)
    the user is not permitted to perform any action prior to successful identification of the user.

- User-subject binding (FIA_USB):

  - User-subject binding (FIA_USB.1)
    a user's security attributes are associated with a subject (application) while the subject is acting on the user's behalf.

A.7 Security management

Functional class "Security management" (FMT) is required to specify the management of security attributes, TSF data and TSF functions. The class provides 6 families of behaviour, thus:

- Management of functions in the TSF (FMT_MOF):

  - Management of security functions behaviour (FMT_MOF.1)
    suitably authorized users (referred to as "roles") can manage the behaviour of security functions by, for example, enabling and disabling them.

- Management of security attributes (FMT_MSA):

  - Management of security attributes (FMT_MSA.1)
    suitably authorized users can manage a specified range of security attributes such as user-group membership and access rights;

  - Secure security attributes (FMT_MSA.2)
    security attributes are evaluated when created to determine whether they contain values appropriate (long enough, complex enough, etc.) to ensure that they remain secure during their lifetime;

  - Static attribute initialization (FMT_MSA.3)
    default values of security attributes are provided and are set to values which are appropriate to their use.

- Management of TSF data (FMT_MTD):

  - Management of TSF data (FMT_MTD.1)
    suitably authorized users (referred to as "roles") can manage the values of TSF data, for example, setting and resetting the system time;

  - Management of limits on TSF data (FMT_MTD.2)
    suitably authorized users can specify actions to be taken if TSF data reaches and exceeds predefined limits;

  - Secure TSF data (FMT_MTD.3)
    TSF data items are constantly evaluated to determine that they contain values which will ensure that the TOE remains secure.

- Revocation (FMT_REV):
  - Revocation (FMT_REV.1)
    assigned security attributes are reviewed according to a set of predefined rules and, if the specified criteria are met (i.e., the rules are broken), the security attributes are revoked.
- Security attribute expiration (FMT_SAE):
  - Time-limited authorization (FMT_SAE.1)
    authorization of a user can be limited to a specific time period after which authorization is removed.
- Security management roles (FMT_SMR):
  - Security roles (FMT_SMR.1)
    the security roles of particular users can be established and maintained;
  - Restrictions on security roles (FMT_SMR.2)
    the security roles of particular users and the relationships between roles can be established and maintained;
  - Assuming roles (FMT_SMR.3)
    certain key roles such as that of administrator, can only be assumed by a user following a specific request.

A.8 Privacy

Functional class "Privacy" (FPR) is required to provide protection against discovery and misuse of a user's identity by other users. The class provides 4 families of behaviour, thus:

- Anonymity (FPR_ANO):
  - Anonymity (FPR_ANO.1)
    the identity of a user cannot be determined by other users except, possibly, those that are suitably authorized;
  - Anonymity without soliciting information (FPR_ANO.2)
    in specific instances, the true identity of a user can be protected by preventing the TOE from requesting the information.
- Pseudonymity (FPR_PSE):
  - Pseudonymity (FPR_PSE.1)
    a user can be associated directly with an application (or group of applications) for accounting purposes but other users or applications are unable to determine that user's true identity;
  - Reversible pseudonymity (FPR_PSE.2)
    Under predefined conditions it is possible for a suitably authorized user to determine the true identity of a user from that user's pseudonym;
  - Alias pseudonymity (FPR_PSE.3)
    a user can be known to the TOE by an alternative identity (alias) in order to avoid revealing the user's true identity.
- Unlinkability (FPR_UNL):
  - Unlinkability (FPR_UNL.1)
    it is not possible to determine the identity of a user of an application through links to other applications being accessed by the same user.

- Unobservability (FPR_UNO):

  - Unobservability (FPR_UNO.1)
    use of an application by a user is not visible to other users or applications;

  - Allocation of information affecting unobservability (FPR_UNO.2)
    use of an application by a user is not visible to other users or applications. Information relating to the specific user's use of the application is distributed to reduce the risk of compromise in the event of a successful attack;

  - Unobservability without soliciting information (FPR_UNO.3)
    in specific instances, a user's privacy-related information can be protected from observation by other users or applications by preventing the TOE from requesting the information;

  - Authorized user observability (FPR_UNO.4)
    use of an application by a user is visible on request to suitably authorized other users.

A.9 Protection of the TSF

Functional class "Protection of the TSF" (FPT) is required to ensure the management and integrity of security functions provided by the TOE and their associated data. The class provides 16 families of behaviour, thus:

- Underlying abstract machine test (FPT_AMT):

  - Abstract machine testing (FPT_AMT.1)
    the characteristics of the underlying hardware or hardware/software combination upon which the functions of the TOE depend are tested periodically.

- Fail secure (FPT_FLS):

  - Failure with preservation of secure state (FPT_FLS.1)
    in the event of specified failures within the TOE, the TSF data are maintained in a consistent and known state and the TOE Security Policy (TSP) continues to be enforced.

- Availability of exported TSF data (FPT_ITA):

  - Inter-TSF availability within a defined availability metric (FPT_ITA.1)
    TSF data transmitted to or from a remote trusted IT product is protected, within specified limits, against loss of availability.

- Confidentiality of exported TSF data (FPT_ITC):

  - Inter-TSF confidentiality during transmission (FPT_ITC.1)
    TSF data transmitted to or from a remote trusted IT product is protected against unauthorized disclosure.

- Integrity of exported data (FPT_ITI):

  - Inter-TSF detection of modification (FPT_ITI.1)
    any modification to TSF data during transmission between the TOE and a remote trusted IT product is detected and reported;

  - Inter-TSF detection and correction of modification (FPT_ITI.2)
    any modification to TSF data during transmission between the TOE and a remote trusted IT product is detected and corrected.

- Internal TOE TSF data transfer (FPT_ITT):
  - Basic internal TSF data transfer protection (FPT_ITT.1)
    TSF data is protected against modification and/or disclosure during transmission between the TSF and another part of the TOE;
  - TSF data transfer separation (FPT_ITT.2)
    user data and TSF data are either physically or logically separated during transmission between the TSF and another part of the TOE;
  - TSF data integrity monitoring (FPT_ITT.3)
    TSF data is monitored for errors in its integrity during transmission between the TSF and another part of the TOE.
- TSF physical protection (FPT_PHP):
  - Passive detection of physical attack (FPT_PHP.1)
    unauthorized attempts to interfere with any TSF data are detected and recorded but not automatically reported;
  - Notification of physical attack (FPT_PHP.2)
    unauthorized attempts to interfere with any TSF data are detected, recorded and automatically reported;
  - Resistance to physical attack (FPT_PHP.3)
    unauthorized attempts to interfere with any TSF data are detected and actively resisted.
- Trusted recovery (FPT_RCV):
  - Manual recovery (FPT_RCV.1)
    authorized human users have procedures available to them for returning the TOE to a known secure state following an interruption in operation of the TOE;
  - Automated recovery (FPT_RCV.2)
    after at least one type of interruption to service, the TOE returns to a known secure state. Other type of service interruption may require the intervention of a suitably authorized human user;
  - Automated recovery without undue loss (FPT_RCV.3)
    after at least one type of interruption to service, the TOE returns to a known secure state with essential TSF data and objects available and with the same contents as before the interruption;
  - Function recovery (FPT_RCV.4)
    in the event of an interruption of operation, specific security functions are able to continue to successful completion or their associated TSF data is returned to a secure state.
- Replay detection (FPT_RPL):
  - Replay detection (FPT_RPL.1)
    attempts to resend encrypted data with all cryptographic parameters unchanged (i.e., identical data after encryption) are detected.
- Reference mediation (FPT_RVM):
  - Non-bypassability of the TSP (FPT_RVM.1)
    security policy is enforced by a function which is tamperproof (FPT_SEP), permanently active (FPT_RVM) and conceptually simple in design (ADV_INT).

- Domain separation (FPT_SEP):
  - TSF domain separation (FPT_SEP.1)
    The TSF executes in a part of the TOE separate from other functions and protected against external interference and tampering;
  - SFP domain separation (FPT_SEP.2)
    each TSF sub-function executes in a part of the TSF which is distinct and separate from other TSF and non-TSF functional parts;
  - Complete reference monitor (FPT_SEP.3)
    each security function, security policy and non security function executes in a part of the TOE which is distinct and separate from each other.
- State synchrony protocol (FPT_SSP):
  - Simple trusted acknowledgement (FPT_SSP.1)
    when data is transmitted between distributed security functions, the receiving TSF indicates with an acknowledgement message to the sending TSF that the data has been received intact and uncompromised;
  - Mutual trusted acknowledgement (FPT_SSP.2)
    when data is transmitted between distributed security functions, the receiving TSF indicates with an acknowledgement message to the sending TSF that the data has been received intact and uncompromised. The sending TSF responds with a message to the receiving TSF acknowledging receipt of the first acknowledgement message.
- Time stamps (FPT_STM):
  - Reliable time stamps (FPT_STM.1)
    actions within a security function can be reliably time stamped for audit and other purposes.
- Inter-TSF TSF data consistency (FPT_TDC):
  - Inter-TFS basic TSF data consistency (FPT_TDC.1)
    the consistency of security attributes shared by a TSF with other trusted IT products is assured.
- Internal TOE TSF data replication consistency (FPT_TRC):
  - Internal TSF consistency (FPT_TRC.1)
    the consistency of TSF data which is replicated and used in other parts of the TOE is assured.
- TSF self test (FPT_TST):
  - TSF testing (FPT_TST.1)
    the operation of a TSF and the integrity of TSF data and executable code can be checked by the TOE periodically or on request from a suitably authorized user.

A.10 Resource utilization

Functional class "Resource utilization" (FRU) is required to ensure the availability of TOE resources such as processing capability and storage capacity. The class provides 3 families of behaviour, thus:

- Fault tolerance (FRU_FLT):
  - Degraded fault tolerance (FRU_FLT.1)
    the correct operation of predefined TSF capabilities continues in the event of one of a range of specified failures;
  - Limited fault tolerance (FRU_FLT.2)
    the correct operation of all TSF capabilities continues in the event of one of a range of specified failures.

- Priority of service (FRU_PRS):
  - Limited priority of service (FRU_PRS.1)
    priorities can be assigned to users and applications to allow them greater or lesser use specific system resources in the event that requests for resource exceed the availability;
  - Full priority of service (FRU_PRS.2)
    priorities can be assigned to users and applications to allow them greater or lesser use all system resources in the event that requests for resource exceed the availability.
- Resource allocation (FRU_RSA):
  - Maximum quotas (FRU_RSA.1)
    mechanisms exist to ensure that any user or application is unable to completely monopolize controlled system resources;
  - Minimum and maximum quotas (FRU_RSA.2)
    mechanisms exist to ensure that any user or application always has a defined minimum level of a specific system resource available to it but is unable to completely monopolize controlled system resources.

A.11 TOE Access

Functional class "TOE Access" (FTA) is required for controlling the establishment of user sessions. The class provides 6 families of behaviour, thus:

- Limitation of scope of selectable attributes (FTA_LSA):
  - Limitation of scope of selectable attributes (FTA_LSA.1)
    the security attributes and capabilities available to a user can be configured according to the method used to access the TOE, the location of the access point and/or the date and time of access.
- Limitation on multiple concurrent sessions (FTA_MCS):
  - Basic limitation on multiple concurrent session (FTA_MCS.1)
    the number of sessions that a single user can be involved in concurrently is limited to the same value for all users;
  - Per-user attribute limitation on multiple concurrent sessions (FTA_MCS.2)
    the number of sessions that a single user can be involved in concurrently is limited according the user's identity and other security attributes.
- Session locking (FTA_SSL):
  - TSF-initiated session locking (FTA_SSL.1)
    the TSF is able to suspend an active session after, for example, a specified period of inactivity. The sequence of events required to re-activate the session are also specified;
  - User-initiated locking (FTA_SSL.2)
    a suitable authorized user is able to suspend and re-activate a current session;
  - TSF-initiated termination (FTA_SSL.3)
    the TSF is able to suspend an active session after, for example, a specified period of inactivity.
- TOE access banners (FTA_TAB):
  - Default TOE access banners (FTA_TAB.1)
    an advisory notice related to the unauthorized use of the TOE is presented to the user prior to the establishment of a session.

- TOE access history (FTA_TAH):
  - TOE access history (FTA_TAH.1)
    details of all unsuccessful attempts to access a user's account are presented to the user on successful session establishment.
- TOE Session establishment (FTA_TSE):
  - TOE session establishment (FTA_TSE.1)
    establishment of a user session can be prevented on the basis of a range of predefined parameters such as user identity, clearance level and time-of-day.

A.12 Trusted path/channels

Functional class "Trusted path/channels" (FTP) is required for (a) establishing trusted communication paths between users and the TSF and (b) establishing trusted communication channels between the TSF and other trusted IT products.

NOTE 1: A trusted communication path provides the means for a user to interact securely with the TSF.

NOTE 2: A trusted communication channel provides a non-reputable means for the TSF and another IT product to determine each other's identity prior to establishing communication.

The class provides 2 families of behaviour, thus:

- Inter-TSF trusted channel (FTP_ITC):
  - Inter-TSF trusted channel (FTP_ITC.1)
    the TSF is able to provide a trusted communication channel between itself and another trusted IT product.
- Trusted path (FTP_TRP):
  - Trusted path (FTP_TRP.1)
    a trusted path is established between the TSF and the user for a range of predefined, security-related interactions.

Annex B (normative): Protocol Implementation Conformance Statement (PICS)

The tool for selection of options recommended in communication standards is the Protocol Implementation Conformance Statement (PICS). EG 201 058 [12] offers guidance to standards writers on the preparation of a PICS and the role of the PICS in standards development is defined in ETS 300 406 [5].

In the context of ISO/IEC 15408-3 [19] the PICS fulfils the following tasks:

- Provides an overview of the capabilities supported by the implementation;
- Explicitly identifies the options available within the standards and the static consequences of their selection;
- May be used to statically check the interworking capabilities of two implementations;
- Acts as a standard checklist of the static conformance requirements of the base specification.

A PICS proforma is a set of tables containing questions to be answered by an implementor where there are specified constraints on the possible answers. The questions asked in a PICS are of two types:

- questions to be answered by either "YES" or "NO", related to whether a feature has been implemented or not. The allowed answers reflect the base specification;
- questions on numerical values implemented (for timers, for sizes of messages, etc.) where the legitimate range of variation reflects the base specification.

In order to allow control of the configuration of the TOE it is essential that the developer provides a PICS and that the implementor completes the proforma for the implementation. The constraints applied to any answer in the PICS are generally of the form:

- Mandatory;
- Conditional, where the condition is stated.

An example drawn from TETRA security (EN 300 392-7 [1] and EN 300 396-6) of the use of tables and conditions in a PICS is given below:

Table B.1: V+D Security class supported

Item	Role	Reference	Status	Support
1	Class 1	EN 300 392-7 [1], clause 6.1.1	o.1	
2	Class 2	EN 300 392-7 [1], clause 6.1.1	o.1	
3	Class 3	EN 300 392-7 [1], clause 6.1.1	o.1	
o.1:	It is mandatory to support at least one of these items.			

Table B.2: V+D Security capabilities supported

Item	Security capability	Reference	Status	Support
1	Authentication	EN 300 392-7 [1], clause 4	c**Error! Reference source not found.**01	
2	OTAR	EN 300 392-7 [1], clause 4	c**Error! Reference source not found.**01	
3	Enable/disable	EN 300 392-7 [1], clause 5	m	
4	AI encryption	EN 300 392-7 [1], clause 6	c**Error! Reference source not found.**02	
5	End-to-end encryption	EN 300 392-7 [1], clause 7	o	
6	TEI delivery	EN 300 392-7 [1], clause 4.1.6	m	
7	ESI	EN 300 392-7 [1], clause 4.2.5	c**Error! Reference source not found.**02	
8	Key change protocol	EN 300 392-7 [1], clause 4.4.6	c**Error! Reference source not found.**03	

c**Error! Reference source not found.**01:	IF B.**Error! Reference source not found.**/3 THEN m ELSE o If security class 3 then mandatory else optional
c**Error! Reference source not found.**02: **Reference source not found.**/2 THEN m ELSE n/a	F B.**Error! Reference source not found.**/3 or B.**Error!** If security class 3, or security class 2 then mandatory else not applicable
c**Error! Reference source not found.**03: **Reference source not found.**/2 THEN m ELSE n/a	F B.**Error! Reference source not found.**/3 or B.**Error!** If security class 3 or if OTAR supported then mandatory else not applicable

A developer should be able to demonstrate that the set of implementations available from a base standard, as evidenced in the PICS, maintain conformance to the security objectives and requirements of the standard.

Annex C (informative): Bibliography

ETSI TS 102 165-1: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Protocol Framework Definition; Methods and Protocols for Security; Part 1: Threat Analysis".

ETSI TS 102 165-2: "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON) Release 4; Protocol Framework Definition; Methods and Protocols for Security; Part 2: Counter Measures".

ETSI TR 101 052: "Security Algorithms Group of Experts (SAGE); Rules for the management of the TETRA standard authentication and key management algorithm set TAA1".

ETSI ETR 232: "Security Techniques Advisory Group (STAG); Glossary of security terminology".

ETSI ETR 237 (1996): "Security Techniques Advisory Group (STAG); Baseline security standards; Features and mechanisms".

ISO/IEC 10181-4: "Information technology - Open systems interconnection - Security frameworks for open systems: Non-repudiation framework".

ISO/IEC 9798-1: "Information technology - Security techniques -Entity authentication - Part 1: Entity authentication mechanisms".

ISO/IEC 9798-2: "Information technology - Security techniques - Entity authentication - Part 2: Mechanisms using symmetric encipherment algorithms".

ISO/IEC 9798-3: "Information technology - Security techniques - Entity authentication - Part 3: Entity authentication using a public key algorithm".

CORAS: "UML profile for security assessment", Mass Soldal Lund, Ida Hogganvik, Fredrik Seehusen, Ketil Stølen. SINTEF Telecom and Informatics, December 2003 (http://coras.sourceforge.com).

Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications - OJ L 201, 31.07.2002).

ETSI EN 300 396-6: "Terrestrial Trunked Radio (TETRA); Direct Mode Operation (DMO); Part 6: Security".

ITU-T Recommendation I.210: "Principles of telecommunication services supported by an ISDN and the means to describe them".

History

Document history			
V1.1.1	February 2005	Membership Approval Procedure	MV 20050401: 2005-02-01 to 2005-04-01
V1.1.1	April 2005	Publication	