Common Criteria
for Information Technology
Security Evaluation

Part 1: Introduction and general model

September 2006

Version 3.1
Revision 1

CCMB-2006-09-001

Foreword

This version of the Common Criteria for Information Technology Security Evaluation (CC v3.1) is the first major revision since being published as CC v2.3 in 2005.

CC v3.1 aims to: eliminate redundant evaluation activities; reduce/eliminate activities that contribute little to the final assurance of a product; clarify CC terminology to reduce misunderstanding; restructure and refocus the evaluation activities to those areas where security assurance is gained; and add new CC requirements if needed.

CC version 3.1 consists of the following parts:

- Part 1: Introduction and general model
- Part 2: Security functional components
- Part 3: Security assurance components

Trademarks:

- UNIX is a registered trademark of The Open Group in the United States and other countries
- Windows is a registered trademark of Microsoft Corporation in the United States and other countries

Legal Notice:

Australia/New Zealand:	*The Defence Signals Directorate and the* *Government Communications Security Bureau respectively;*
Canada:	*Communications Security Establishment;*
France:	*Direction Centrale de la Sécurité des Systèmes d'Information;*
Germany:	*Bundesamt für Sicherheit in der Informationstechnik;*
Japan:	*Information Technology Promotion Agency*
Netherlands:	*Netherlands National Communications Security Agency;*
Spain:	*Ministerio de Administraciones Públicas and* *Centro Criptológico Nacional;*
United Kingdom:	*Communications-Electronics Security Group;*
United States:	*The National Security Agency and the* *National Institute of Standards and Technology.*

Table of Contents

Table of contents

List of figures

List of tables

1 Introduction

1 The CC permits comparability between the results of independent security evaluations. The CC does so by providing a common set of requirements for the security functionality of IT products and for assurance measures applied to these IT products during a security evaluation. These IT products may be implemented in hardware, firmware or software.

2 The evaluation process establishes a level of confidence that the security functionality of these IT products and the assurance measures applied to these IT products meet these requirements. The evaluation results may help consumers to determine whether these IT products fulfil their security needs.

3 The CC is useful as a guide for the development, evaluation and/or procurement of IT products with security functionality.

4 The CC addresses protection of assets from unauthorised disclosure, modification, or loss of use. The categories of protection relating to these three types of failure of security are commonly called confidentiality, integrity, and availability, respectively. The CC may also be applicable to aspects of IT security outside of these three. The CC is applicable to risks arising from human activities (malicious or otherwise) and to risks arising from non-human activities. Apart from IT security, the CC may be applied in other areas of IT, but makes no claim of applicability in these areas.

2 Scope

5 This multi-part standard, the Common Criteria (CC), is meant to be used as the basis for evaluation of security properties of IT products. By establishing such a common criteria base, the results of an IT security evaluation may be meaningful to a wider audience.

6 Certain topics, because they involve specialised techniques or because they are somewhat peripheral to IT security, are considered to be outside the scope of the CC. Some of these are identified below.

- The CC does not contain security evaluation criteria pertaining to administrative security measures not related directly to the IT security functionality. However, it is recognised that significant security can often be achieved through or supported by administrative measures such as organisational, personnel, physical, and procedural controls.

- The evaluation of some technical physical aspects of IT security such as electromagnetic emanation control is not specifically covered, although many of the concepts addressed will be applicable to that area.

- The CC does not address the evaluation methodology under which the criteria should be applied. This methodology is described in the Common Methodology for IT Security Evaluation [CEM].

- The CC does not address the administrative and legal framework under which the criteria may be applied by evaluation authorities. However, it is expected that the CC will be used for evaluation purposes in the context of such a framework.

- The procedures for use of evaluation results in accreditation are outside the scope of the CC. Accreditation is the administrative process whereby authority is granted for the operation of an IT product (or collection thereof) in its full operational environment including all of its non-IT parts. The results of the evaluation process are an input to the accreditation process. However, as other techniques are more appropriate for the assessments of non-IT related properties and their relationship to the IT security parts, accreditors should make separate provisions for those aspects.

- The subject of criteria for the assessment of the inherent qualities of cryptographic algorithms is not covered in the CC. Should independent assessment of mathematical properties of cryptography be required, the evaluation scheme under which the CC is applied must make provision for such assessments.

7 The CC is intentionally flexible, enabling a range of evaluation methods to be applied to a range of security properties of a range of IT products.

Therefore care should be exercised to ensure that this flexibility is not misused. For example, the CC should not be used to apply unsuitable evaluation methods, to irrelevant security properties, of inappropriate IT products, resulting in meaningless evaluation results.

8 Consequently, the fact that an IT product has been evaluated has meaning only in the context of the security properties that were evaluated and the evaluation methods that were used. Evaluation authorities should carefully check the products, properties and methods to determine that an evaluation will provide meaningful results. Additionally, purchasers of evaluated products should carefully consider this context to determine whether the evaluated product is useful and applicable to their specific situation and needs.

3 Normative references

9 The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

CEM Common Methodology for Information Technology Security Evaluation, Version 3.1, revision 1, September 2006.

ISO/IEC ISO/IEC Directives - Part 2: Rules for the structure and drafting of International Standards.

4 Terms and definitions

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

11 This Chapter 4 contains only those terms which are used in a specialised way throughout the CC. Some combinations of common terms used in the CC, while not meriting inclusion in this Chapter 4, are explained for clarity in the context where they are used.

12 **assets** — entities that the owner of the TOE presumably places value upon.

13 **assignment** — the specification of an identified parameter in a component (of the CC) or requirement.

14 **assurance** — grounds for confidence that a TOE meets the SFRs.

15 **attack potential** — a measure of the effort to be expended in attacking a TOE, expressed in terms of an attacker's expertise, resources and motivation.

16 **augmentation** — the addition of one or more requirement(s) to a package.

17 **authentication data** — information used to verify the claimed identity of a user.

18 **authorised user** — a user who may, in accordance with the SFRs, perform an operation.

19 **can** — within normative text, “can” indicates “statements of possibility and capability, whether material, physical or causal” (ISO/IEC).

20 **class** — a grouping of CC families that share a common focus.

21 **coherent** — an entity is logically ordered and has a discernible meaning. For documentation, this addresses both the actual text and the structure of the document, in terms of whether it is understandable by its target audience.

22 **complete** — all necessary parts of an entity have been provided. In terms of documentation, this means that all relevant information is covered in the documentation, at such a level of detail that no further explanation is required at that level of abstraction.

23 **component** — the smallest selectable set of elements on which requirements may be based.

24 **component TOE** — an evaluated TOE that is part of another TOE.

25 **composed assurance package (CAP)** — an assurance package, consisting of requirements drawn from CC Part 3 (predominately from the ACO class), representing a point on the CC predefined composition assurance scale.

26 **confirm** — this term is used to indicate that something needs to be reviewed in detail, and that an independent determination of sufficiency needs to be made. The level of rigour required depends on the nature of the subject matter. This term is only applied to evaluator actions.

27 **connectivity** — the property of the TOE which allows interaction with IT entities external to the TOE. This includes exchange of data by wire or by wireless means, over any distance in any environment or configuration.

28 **consistent** — this term describes a relationship between two or more entities, indicating that there are no apparent contradictions between these entities.

29 **counter (verb)** — this term is typically used when the impact of a particular threat is mitigated but not necessarily eradicated.

30 **demonstrate** — this term refers to an analysis leading to a conclusion, which is less rigorous than a “proof”.

31 **dependency** — a relationship between components such that if a requirement based on the depending component is included in a PP, ST or package, a requirement based on the component that is depended upon must normally also be included in the PP, ST or package.

32 **describe** — this term requires that specific details of an entity be provided.

33 **determine** — this term requires an independent analysis to be made, with the objective of reaching a particular conclusion. The usage of this term differs from “confirm” or “verify”, since these other terms imply that an analysis has already been performed which needs to be reviewed, whereas the usage of “determine” implies a truly independent analysis, usually in the absence of any previous analysis having been performed.

34 **development environment** — the environment in which the TOE is developed.

35 **element** — an indivisible statement of security need.

36 **ensure** — this term, used by itself, implies a strong causal relationship between an action and its consequences. When this term is preceded by the word “helps” it indicates that the consequence is not fully certain, on the basis of that action alone.

37 **evaluation** — assessment of a PP, an ST or a TOE, against defined criteria.

38 **evaluation assurance level (EAL)** — an assurance package, consisting of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale.

39 **evaluation authority** — a body that implements the CC for a specific community by means of an evaluation scheme and thereby sets the standards

and monitors the quality of evaluations conducted by bodies within that community.

40 **evaluation scheme** — the administrative and regulatory framework under which the CC is applied by an evaluation authority within a specific community.

41 **exhaustive** — this term is used in the CC with respect to conducting an analysis or other activity. It is related to “systematic” but is considerably stronger, in that it indicates not only that a methodical approach has been taken to perform the analysis or activity according to an unambiguous plan, but that the plan that was followed is sufficient to ensure that all possible avenues have been exercised.

42 **explain** — this term differs from both “describe” and “demonstrate”. It is intended to answer the question “Why?” without actually attempting to argue that the course of action that was taken was necessarily optimal.

43 **extension** — the addition to an ST or PP of functional requirements not contained in Part 2 and/or assurance requirements not contained in Part 3 of the CC.

44 **external entity** — any entity (human or IT) outside the TOE that interacts (or may interact) with the TOE.

45 **family** — a grouping of components that share a similar goal but may differ in emphasis or rigour.

46 **formal** — expressed in a restricted syntax language with defined semantics based on well-established mathematical concepts.

47 **guidance documentation** — documentation that describes the delivery, preparation, operation, management and/or use of the TOE.

48 **identity** — a representation (e.g. a string) uniquely identifying an authorised user, which can either be the full or abbreviated name of that user or a pseudonym.

49 **informal** — expressed in natural language.

50 **informative** — informative text “provides additional information intended to assist the understanding or use of the document.” (ISO/IEC).

51 **inter-TSF transfers** — communicating data between the TOE and the security functionality of other trusted IT products.

52 **internal communication channel** — a communication channel between separated parts of the TOE.

53 **internal TOE transfer** — communicating data between separated parts of the TOE.

54 **internally consistent** — this term means that there are no apparent contradictions between any aspects of an entity. In terms of documentation, this means that there can be no statements within the documentation that can be taken to contradict each other.

55 **iteration** — the use of the same component to express two or more distinct requirements.

56 **justification** — this term refers to an analysis leading to a conclusion, but is more rigorous than a demonstration. This term requires significant rigour in terms of very carefully and thoroughly explaining every step of a logical argument.

57 **may** — within normative text, “may” indicates “a course of action permissible within the limits of the document” (ISO/IEC).

58 **normative** — normative text “describes the scope of the document, and sets out provisions.” (ISO/IEC). Within normative text, the verbs “shall”, “should”, “may”, and “can” have the ISO standard meanings described in this glossary and the verb “must” is not used. Unless explicitly labelled “informative”, all CC text is normative.

59 **object** — a passive entity in the TOE, that contains or receives information, and upon which subjects perform operations.

60 **operation (on a component of the CC)** — modifying or repeating that component. Allowed operations on components are assignment, iteration, refinement and selection.

61 **operation (on an object)** — a specific type of action performed by a subject on an object.

62 **operational environment** — the environment in which the TOE is operated.

63 **organisational security policy (OSP)** — a set of security rules, procedures, or guidelines imposed (or presumed to be imposed) now and/or in the future by an actual or hypothetical organisation in the operational environment.

64 **package** — a named set of either functional or assurance requirements (e.g. EAL 3).

65 **PP evaluation** — assessment of a PP against defined criteria.

66 **Protection Profile (PP)** — an implementation-independent statement of security needs for a TOE type.

67 **prove** — this term refers to a formal analysis in its mathematical sense. It is completely rigorous in all ways. Typically, “prove” is used when there is a desire to show correspondence between two TSF representations at a high level of rigour.

68 **refinement** — the addition of details to a component.

69 **role** — a predefined set of rules establishing the allowed interactions between a user and the TOE.

70 **secret** — information that must be known only to authorised users and/or the TSF in order to enforce a specific SFP.

71 **secure state** — a state in which the TSF data are consistent and the TSF continues correct enforcement of the SFRs.

72 **security attribute** — a property of subjects, users (including external IT products), objects, information, sessions and/or resources that is used in defining the SFRs and whose values are used in enforcing the SFRs.

73 **security function policy (SFP)** — a set of rules describing specific security behaviour enforced by the TSF and expressible as a set of SFRs.

74 **security objective** — a statement of intent to counter identified threats and/or satisfy identified organisation security policies and/or assumptions.

75 **Security Target (ST)** — an implementation-dependent statement of security needs for a specific identified TOE.

76 **selection** — the specification of one or more items from a list in a component.

77 **semiformal** — expressed in a restricted syntax language with defined semantics.

78 **shall** — within normative text, “shall” indicates “requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.” (ISO/IEC).

79 **should** — within normative text, “should” indicates “that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required.” (ISO/IEC) The CC interprets 'not necessarily required' to mean that the choice of another possibility requires a justification of why the preferred option was not chosen.

80 **specify** — this term is used in the same context as “describe”, but is intended to be more rigorous and precise. It is very similar to “define”.

81 **ST evaluation** — assessment of an ST against defined criteria.

82 **subject** — an active entity in the TOE that performs operations on objects.

83 **target of evaluation (TOE)** — a set of software, firmware and/or hardware possibly accompanied by guidance.

84 **TOE evaluation** — assessment of a TOE against defined criteria.

85 **TOE resource** — anything useable or consumable in the TOE.

86 **TOE Security Functionality (TSF)** — a set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the SFRs.

87 **trace (verb)** — this term is used to indicate that an informal correspondence is required between two entities with only a minimal level of rigour.

88 **transfers outside of the TOE** — TSF mediated communication of data to entities not under control of the TSF.

89 **trusted channel** — a means by which a TSF and a remote trusted IT product can communicate with necessary confidence.

90 **trusted IT product** — an IT product other than the TOE which has its security functional requirements administratively coordinated with the TOE and which is assumed to enforce its security functional requirements correctly (e. g. by being separately evaluated).

91 **trusted path** — a means by which a user and a TSF can communicate with necessary confidence.

92 **TSF data** — data created by and for the TOE, that might affect the operation of the TOE.

93 **TSF interface (TSFI)** — a means by which external entities (or subjects in the TOE but outside of the TSF) supply data to the TSF, receive data from the TSF and invoke services from the TSF.

94 **user** — see **external entity** .

95 **user data** — data created by and for the user, that does not affect the operation of the TSF.

96 **verify** — this term is similar in context to “confirm”, but has more rigorous connotations. This term when used in the context of evaluator actions indicates that an independent effort is required of the evaluator.

4.1 Terms and definitions related to the ADV class

97 The following terms are used in the requirements for software internal structuring. Some of these are derived from the *Institute of Electrical and Electronics Engineers Glossary of software engineering terminology, IEEE Std 610.12-1990.*

98 **administrator** — an entity that has complete trust with respect to all policies implemented by the TSF.

99 **call tree** — a diagram that identifies the modules in a system and shows which modules call one another. All the modules named in a call tree that originates with (i.e., is rooted by) a specific module are the modules that directly or indirectly implement the functions of the originating module.

100 **cohesion (also called module strength)** — the manner and degree to which the tasks performed by a single software module are related to one another; types of cohesion include coincidental, communicational, functional, logical, sequential, and temporal. These types of cohesion are characterised below, listed in the order of decreasing desirability.

101 **coincidental cohesion** — a module with this characteristic performs unrelated, or loosely related, activities.

102 **communicational cohesion** — a module with this characteristic contains functions that produce output for, or use output from, other functions within the module. An example of a communicationally cohesive module is an *access check* module that includes mandatory, discretionary, and capability checks.

103 **complexity** — this is a measure of how difficult software is to understand, and thus to analyse, test, and maintain. Reducing complexity is the ultimate goal for using modular decomposition, layering and minimisation. Controlling coupling and cohesion contributes significantly to this goal.

104 A good deal of effort in the software engineering field has been expended in attempting to develop metrics to measure the complexity of source code. Most of these metrics use easily computed properties of the source code, such as the number of operators and operands, the complexity of the control flow graph (cyclomatic complexity), the number of lines of source code, the ratio of comments to executable code, and similar measures. Coding standards have been found to be a useful tool in generating code that is more readily understood.

105 This TSF internals (ADV_INT) family calls for a *complexity analysis* in all components. It is expected that the developer will provide support for the claims that there has been a sufficient reduction in complexity. This support could include the developer's programming standards, and an indication that all modules meet the standard (or that there are some exceptions that are justified by software engineering arguments). It could include the results of tools used to measure some of the properties of the source code. Or it could include other support that the developer finds appropriate.

106 **coupling** — the manner and degree of interdependence between software modules; types of coupling include call, common and content coupling. These types of coupling are characterised below, listed in the order of decreasing desirability

- *call:* two modules are call coupled if they communicate strictly through the use of their documented function calls; examples of call coupling are data, stamp, and control, which are defined below.

    1. *data:* two modules are data coupled if they communicate strictly through the use of call parameters that represent single data items.

    2. *stamp:* two modules are stamp coupled if they communicate through the use of call parameters that comprise multiple fields or that have meaningful internal structures.

    3. *control:* two modules are control coupled if one passes information that is intended to influence the internal logic of the other.

- *common:* two modules are common coupled if they share a common data area or a common system resource. Global variables indicate that modules using those global variables are common coupled. Common coupling through global variables is generally allowed, but only to a limited degree. For example, variables that are placed into a global area, but are used by only a single module, are inappropriately placed, and should be removed. Other factors that need to be considered in assessing the suitability of global variables are:

    1. The number of modules that modify a global variable: In general, only a single module should be allocated the responsibility for controlling the contents of a global variable, but there may be situations in which a second module may share that responsibility; in such a case, sufficient justification must be provided. It is unacceptable for this responsibility to be shared by more than two modules. (In making this assessment, care should be given to determining the module actually responsible for the contents of the variable; for example, if a single routine is used to modify the variable, but that routine simply performs the modification requested by its caller, it is the calling module that is responsible, and there may be more than one such module). Further, as part of the complexity determination, if two modules are responsible for the contents of a global variable, there should be clear indications of how the modifications are coordinated between them.

    2. The number of modules that reference a global variable: Although there is generally no limit on the number of modules that reference a global variable, cases in which many modules make such a reference should be examined for validity and necessity.

- *content:* two modules are content coupled if one can make direct reference to the internals of the other (e.g. modifying code of, or referencing labels internal to, the other module). The result is that some or all of the content of one module are effectively included in the other. Content coupling can be thought of as using unadvertised module interfaces; this is in contrast to call coupling, which uses only advertised module interfaces.

107 **domain separation** — the security architecture property whereby the TSF defines separate security domains for each user and for the TSF and ensures that no user process can affect the contents of a security domain of another user or of the TSF.

108 **functional cohesion** — a module with this characteristic performs activities related to a single purpose. A functionally cohesive module transforms a single type of input into a single type of output, such as a stack manager or a queue manager.

109 **interaction** — a general communication-based relationship between entities.

110 **interface** — a means of interaction with a component or module.

111 **layering** — the design of software such that separate groups of modules (the *layers*) are hierarchically organised to have separate responsibilities such that one layer depends only on layers below it in the hierarchy for services, and provides its services only to the layers above it. Strict layering adds the constraint that each layer receives services only from the layer immediately beneath it, and provides services only to the layer immediately above it.

112 **logical (or procedural) cohesion** — a module with this characteristic performs similar activities on different data structures. A module exhibits logical cohesion if its functions perform related, but different, operations on different inputs.

113 **modular decomposition** — the process of breaking a system into components to facilitate design and development.

114 **non-bypassability (of the TSF)** — the security architecture property whereby all SFR-related actions are mediated by the TSF.

115 **security domain** — the collection of resources to which an active entity has access.

116 **sequential cohesion** — a module with this characteristic contains functions each of whose output is input for the following function in the module. An example of a sequentially cohesive module is one that contains the functions to write audit records and to maintain a running count of the accumulated number of audit violations of a specified type.

117 **software engineering** — the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software. As with engineering practises in general, some amount of judgement must be used in applying engineering principles. Many factors affect choices, not just the application of measures of modular decomposition, layering, and minimisation. For example, a developer may design a system with future

applications in mind that will not be implemented initially. The developer may choose to include some logic to handle these future applications without fully implementing them; further, the developer may include some calls to as-yet unimplemented modules, leaving call stubs. The developer's justification for such deviations from well-structured programs will have to be assessed using judgement, as well as the application of good software engineering discipline.

118 **temporal cohesion** — a module with this characteristic contains functions that need to be executed at about the same time. Examples of temporally cohesive modules include *initialisation*, *recovery*, and *shutdown* modules.

119 **TSF self-protection** — the security architecture property whereby the TSF cannot be corrupted by non-TSF code or entities.

4.2 Terms and definitions related to the AGD class

120 **installation** — the procedures that the user has to perform normally only once after receipt and acceptance of the TOE to progress it to the secure configuration as described in the ST including the embedding of the TOE in its operational environment. If similar processes have to be performed by the developer they are denoted as “generation” throughout ALC: Life-cycle support. If the TOE requires an initial start-up that does not need to be repeated regularly, this process would be classified as installation here.

121 **operation** — the usage phase of the TOE. This includes “normal usage”, administration and maintenance of the TOE.

122 **operation (of the TOE)** — usage of the TOE after delivery and preparation.

123 **preparation** — the product life-cycle phase comprising the customer's acceptance of the delivered TOE and its installation which may include such things as booting, initialisation, start-up, progressing the TOE to a state ready for operation.

4.3 Terms and definitions related to the ALC class

124 **acceptance criteria** — the criteria to be applied when performing the acceptance procedures (e.g. successful document review, or successful testing in the case of software, firmware or hardware).

125 **acceptance procedures** — the procedures followed in order to accept newly created or modified configuration items as part of the TOE, or to move them to the next step of the life-cycle. These procedures identify the roles or individuals responsible for the acceptance and the criteria to be applied to decide on the acceptance.

126 There are several types of acceptance situations some of which may overlap:

- acceptance of an item into the CM system for the first time, in particular inclusion of software, firmware and hardware components from other manufacturers into the TOE (“integration”);
- progression of configuration items to the next life-cycle phase at each stage of the construction of the TOE (e.g. module, subsystem, quality control of the finished TOE);
- subsequent to transports of configuration items (for example parts of the TOE or preliminary products) between different development sites;
- subsequent to the delivery of the TOE to the consumer.

127 **CM documentation (documentation of the CM system) —** overall term for the following:

- CM output
  - CM list (configuration list)
  - CM system records
- CM plan
- CM usage documentation

128 **CM evidence —** everything that may be used to establish confidence in the correct operation of the CM system, e.g., CM output, rationales provided by the developer, observations, experiments or interviews made by the evaluator during a site visit.

129 **CM item (configuration item) —** object managed by the CM system during the TOE development. These may be either parts of the TOE or objects related to the development of the TOE like evaluation documents or development tools. CM items may be stored in the CM system directly (for example files) or by reference (for example hardware parts) together with their version.

130 **CM list (configuration list) —** a CM output document listing all configuration items for a specific product together with the exact version of each CM item relevant for a specific version of the complete product. This list allows distinguishing the items belonging to the evaluated version of the product from other versions of these items belonging to other versions of the product. The final CM list is a specific document for a specific version of a specific product. (Of course the list can be an electronic document inside of a CM tool. In that case it can be seen as a specific view into the system or a part of the system rather than an output of the system. However, for the practical use in an evaluation the configuration list will probably be delivered as a part of the evaluation documentation.) The configuration list defines the items that are under the CM requirements of ALC_CMC.

131 **CM output** — CM related results produced or enforced by the CM system. These CM related results could occur as documents (for example filled paper forms, CM system records, logging data, hard-copies and electronic output data) as well as actions (for example manual measures to fulfil CM instructions). Examples of such CM outputs are configuration lists, CM plans and/or behaviours during the product life-cycle.

132 **CM plan** — part of the CM documentation describing how the CM system is used for the TOE. The objective of issuing a CM plan is that staff members can see clearly what they have to do. From the point of view of the overall CM system this can be seen as an output document (because it may be produced as part of the application of the CM system). From the point of view of the concrete project it is a usage document because members of the project team use it in order to understand the steps that they have to perform during the project. The CM plan defines the usage of the system for the specific product; the same system may be used to a different extent for other products. That means the CM plan defines and describes the output of the CM system of a company which is used during the TOE development.

133 **CM system** — overall term for the set of procedures and tools (including their documentation) used by a developer to develop and maintain configurations of his products during their life-cycles. CM systems may have varying degrees of rigour and function. At higher levels, CM systems may be automated, with flaw remediation, change controls, and other tracking mechanisms.

134 **CM system records** — those CM output documents which are produced during the operation of the CM system documenting important activities. Examples of CM system records are CM item change control forms or CM item access approval forms.

135 **CM tools** — tools realising or supporting a CM system, for example tools for the version management of the parts of the TOE. They may require manual operation or may be automated.

136 **CM usage documentation** — that part of the CM system, which describes, how the CM system is defined and applied by using for example handbooks, regulations and/or documentation of tools and procedures.

137 **delivery** — the product life-cycle phase which is concerned with the transmission of the finished TOE from the production environment into the hands of the customer. This may include packaging and storage at the development site, but does not include transportations of the unfinished TOE or parts of the TOE between different developers or different development sites.

138 **developer** — the organisation responsible for the development of the TOE.

139 **development** — the product life-cycle phase which is concerned with generating the implementation representation of the TOE. Throughout the

ALC requirements, development and related terms (developer, develop) are meant in the more general sense to comprise development *and production*.

140 **development tools** — tools (including test software, if applicable) supporting the development and production of the TOE. E.g., for a software TOE, development tools are usually programming languages, compilers, linkers and generating tools.

141 **implementation representation** — the least abstract representation of the TSF, specifically the one that is used to create the TSF itself without further design refinement. Source code that is then compiled or a hardware drawing that is used to build the actual hardware are examples of parts of an implementation representation.

142 **life-cycle** — the sequence of stages of existence of an object (for example a product or a system) in time.

143 **life-cycle definition** — the definition of the life-cycle model.

144 **life-cycle model** — description of the stages and their relations to each other that are used in the management of the life-cycle of a certain object, how the sequence of stages looks like and which high level characteristics the stages have.

145 **measurable life-cycle model** — a life-cycle model using some quantitative valuation (arithmetic parameters and/or metrics) of the managed product in order to measure development properties of the product. Typical metrics are source code complexity metrics, defect density (errors per size of code) or mean time to failure.

146 **production** — the production life-cycle phase follows the development phase and consists of transforming the implementation representation into the implementation of the TOE, i.e. into a state acceptable for delivery to the customer. This phase may comprise manufacturing, integration, generation, internal transports, storage, and labelling of the TOE.

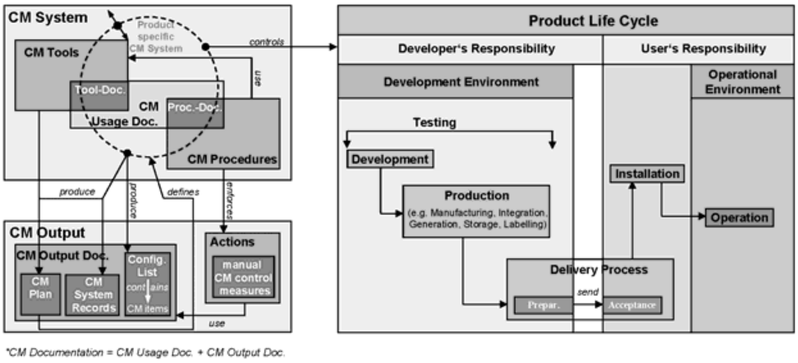

Figure 1 - Terminology in CM and in the product life-cycle

4.4 Terms and definitions related to the AVA class

147 **covert channel** — an enforced, illicit signalling channel that allows a user to surreptitiously contravene the multi-level separation policy and unobservability requirements of the TOE (this is a special case of monitoring attacks).

148 **encountered potential vulnerabilities** — potential weakness in the TOE identified by the evaluator while performing evaluation activities that could be used to violate the SFRs.

149 **exploitable vulnerability** — a weakness in the TOE that can be used to violate the SFRs in the operational environment for the TOE.

150 **monitoring attacks** — a generic category of attack methods that includes passive analysis techniques aiming at disclosure of sensitive internal data of the TOE by operating the TOE in the way that corresponds to the guidance documents.

151 **potential vulnerability** — a weakness the existence of which is suspected (by virtue of a postulated attack path), but not confirmed, to violate the SFRs.

152 **residual vulnerability** — a weakness that cannot be exploited in the operational environment for the TOE, but that could be used to violate the SFRs by an attacker with greater attack potential than is anticipated in the operational environment for the TOE.

153 **vulnerability** — a weakness in the TOE that can be used to violate the SFRs in some environment.

4.5 Terms and definitions related to the ACO class

154 **base component** — the entity in a composed TOE, which has itself been the subject of an evaluation, providing services and resources to a dependent component.

155 **compatible (components)** — one component providing the services required by the other component, through the corresponding interfaces of each component, in consistent operational environments.

156 **composed TOE** — comprised solely of two or more components that have been successfully evaluated.

157 **dependent component** — an entity in a composed TOE, which is itself the subject of an evaluation, relying on the provision on services by a base component.

158 **functional interface** — the (external) interfaces that provide a user with access to functionality of the TOE that is not directly involved in enforcing security functional requirements. In a composed TOE these are the interfaces provided by the base component that are required by the dependent component to support the operation of the composed TOE.

5 Symbols and abbreviated terms

159 The following abbreviations are used in one or more parts of the CC:

API	Application Programming Interface
CAP	Composed Assurance Package
CC	Common Criteria
CCRA	Arrangement on the Recognition of Common Criteria Certificates in the field of IT Security
DAC	Discretionary Access Control
EAL	Evaluation Assurance Level
GHz	Gigahertz
GUI	Graphical User Interface
IC	Integrated Circuit
IOCTL	Input Output Control
IP	Internet Protocol
IT	Information Technology
MB	Mega Byte
OS	Operating System
OSP	Organisational Security Policy
PC	Personal Computer
PCI	Peripheral Component Interconnect
PKI	Public Key Infrastructure
PP	Protection Profile
RAM	Random Access Memory
RPC	Remote Procedure Call
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFP	Security Function Policy
ST	Security Target
TCP	Transmission Control Protocol
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
VPN	Virtual Private Network

6 Overview

160 This Chapter introduces the main concepts of the CC. It identifies the concept “TOE”, the target audience of the CC, and the approach taken to present the material in the remainder of the CC.

6.1 The TOE

161 The previous Sections used the term “IT product”. The CC is flexible in what to evaluate and is therefore not tied to the boundaries of IT products. Instead of the term IT product, the CC uses the term “TOE” (Target of Evaluation).

162 A TOE is defined as a set of software, firmware and/or hardware possibly accompanied by guidance.

163 While there are cases where a TOE consists of an IT product, this need not be the case. The TOE may be an IT product, a part of an IT product, a set of IT products, a unique technology that may never be made into a product, or a combination of these.

164 As far as the CC is concerned, the precise relation between the TOE and any IT products is only important in one aspect: the evaluation of a TOE containing only part of an IT product should not be misrepresented as the evaluation of the entire IT product.

165 Examples of TOEs include:

- A software application;
- An operating system;
- A software application in combination with an operating system;
- A software application in combination with an operating system and a workstation;
- An operating system in combination with a workstation;
- A smart card integrated circuit;
- The cryptographic co-processor of a smart card integrated circuit;
- A Local Area Network including all terminals, servers, network equipment and software;
- A database application excluding the remote client software normally associated with that database application;

6.1.1 Different representations of the TOE

166 In the CC, a TOE can occur in several representations, such as (for a software TOE):

- a list of files in a configuration management system;
- a single master copy, that has just been compiled;
- a box containing a CD-ROM and a manual, ready to be shipped to a customer;
- an installed and operational version.

All of these are considered to be a TOE: and wherever the term “TOE” is used in the remainder of the CC, the context determines the representation that is meant.

6.1.2 Different configurations of the TOE

167 In general, IT products can be configured in many ways: installed in different ways, with different options enabled or disabled. As, during a CC evaluation, it will be determined whether a TOE meets certain requirements, this flexibility in configuration may lead to problems, as all possible configurations of the TOE must meet the requirements. For these reasons, it is often the case that the guidance part of the TOE strongly constrains the possible configurations of the TOE. That is: the guidance of the TOE may be different from the general guidance of the IT product.

168 An example is an operating system IT product. This product can be configured in many ways (e.g. types of users, number of users, types of external connections allowed/disallowed, options enabled/disabled etc.).

169 If the same IT product is to be a TOE, and is evaluated against a reasonable set of requirements, the configuration should be much more tightly controlled, as many options (e.g. allow all types of external connections or the system administrator does not need to be authenticated) will lead to a TOE not meeting the requirements.

170 For this reason, there would normally be a difference between the guidance of the IT product (allowing many configurations) and the guidance of the TOE (allowing only one or only configurations that do not differ in security-relevant ways).

171 Note that if the guidance of the TOE still allows more than one configuration, these configurations are collectively called “the TOE” and each such configuration must meet the requirements levied on the TOE.

6.2 Target audience of the CC

172 There are three groups with a general interest in evaluation of the security properties of TOEs: consumers, developers and evaluators. The criteria presented in this document have been structured to support the needs of all three groups. They are all considered to be the principal users of the CC. The three groups can benefit from the criteria as explained in the following paragraphs.

6.2.1 Consumers

173 The CC is written to ensure that evaluation fulfils the needs of the consumers as this is the fundamental purpose and justification for the evaluation process.

174 Consumers can use the results of evaluations to help decide whether a TOE fulfils their security needs. These security needs are typically identified as a result of both risk analysis and policy direction. Consumers can also use the evaluation results to compare different TOEs.

175 The CC gives consumers, especially in consumer groups and communities of interest, an implementation-independent structure, termed the Protection Profile (PP), in which to express their security requirements in an unambiguous manner.

6.2.2 Developers

176 The CC is intended to support developers in preparing for and assisting in the evaluation of their TOEs and in identifying security requirements to be satisfied by those TOEs. These requirements are contained in an implementation-dependent construct termed the Security Target (ST). This ST may be based on one or more PPs to show that the ST conforms to the security requirements from consumers as laid down in those PPs.

177 The CC can then be used to determine the responsibilities and actions to provide evidence that is necessary to support the evaluation of the TOE against these requirements. It also defines the content and presentation of that evidence.

6.2.3 Evaluators

178 The CC contains criteria to be used by evaluators when forming judgements about the conformance of TOEs to their security requirements. The CC describes the set of general actions the evaluator is to carry out. Note that the CC does not specify procedures to be followed in carrying out those actions. More information on these procedures may be found in Section 6.3.

6.2.4 Others

179 While the CC is oriented towards specification and evaluation of the IT security properties of TOEs, it may also be useful as reference material to all

parties with an interest in or responsibility for IT security. Some of the additional interest groups that can benefit from information contained in the CC are:

- system custodians and system security officers responsible for determining and meeting organisational IT security policies and requirements;

- auditors, both internal and external, responsible for assessing the adequacy of the security of an IT solution (which may consist of or contain a TOE);

- security architects and designers responsible for the specification of security properties of IT products;

- accreditors responsible for accepting an IT solution for use within a particular environment;

- sponsors of evaluation responsible for requesting and supporting an evaluation; and

- evaluation authorities responsible for the management and oversight of IT security evaluation programmes.

6.2.5 The different parts of the CC

180 The CC is presented as a set of distinct but related parts as identified below. Terms used in the description of the parts are explained in Chapter 7.

- **Part 1, Introduction and general model** is the introduction to the CC. It defines the general concepts and principles of IT security evaluation and presents a general model of evaluation.

- **Part 2, Security functional components** establishes a set of functional components that serve as standard templates upon which to base functional requirements for TOEs. CC Part 2 catalogues the set of functional components and organises them in families and classes.

- **Part 3, Security assurance components** establishes a set of assurance components that serve as standard templates upon which to base assurance requirements for TOEs. CC Part 3 catalogues the set of assurance components and organises them into families and classes. CC Part 3 also defines evaluation criteria for PPs and STs and presents seven pre-defined assurance packages which are called the Evaluation Assurance Levels (EALs).

181 In support of the three parts of the CC listed above, other documents have been published, most notably the CEM [CEM]. It is anticipated that other documents will be published, including technical rationale material and guidance documents.

182 The following table presents, for the three key target audience groupings, how the parts of the CC will be of interest.

	Consumers	Developers	Evaluators
Part 1	Use for background information and reference purposes. Guidance structure for PPs.	Use for background information and reference purposes. Development of security specifications for TOEs.	Use for background information and reference purposes. Guidance structure for PPs and STs.
Part 2	Use for guidance and reference when formulating statements of requirements for a TOE.	Use for reference when interpreting statements of functional requirements and formulating functional specifications for TOEs.	Use for reference when interpreting statements of functional requirements.
Part 3	Use for guidance when determining required levels of assurance.	Use for reference when interpreting statements of assurance requirements and determining assurance approaches of TOEs.	Use for reference when interpreting statements of assurance requirements.

Table 1 - Road map to the Common Criteria

6.3 Evaluation context

183 In order to achieve greater comparability between evaluation results, evaluations should be performed within the framework of an authoritative evaluation scheme that sets the standards, monitors the quality of the evaluations and administers the regulations to which the evaluation facilities and evaluators must conform.

184 The CC does not state requirements for the regulatory framework. However, consistency between the regulatory frameworks of different evaluation authorities will be necessary to achieve the goal of mutual recognition of the results of such evaluations.

185 An example of a regulatory framework is the CCRA (Arrangement on the Recognition of the CC Certificates in the field of IT Security). This arrangement has been executed among a number of evaluation authorities in different countries and provides the conditions for mutual recognition of CC certificates between these evaluation authorities.

186 A second way of achieving greater comparability between evaluation results is using a common methodology to achieve these results. For the CC, this methodology has been described in the Common Methodology for IT Security Evaluation [CEM].

187 Use of a common evaluation methodology contributes to the repeatability and objectivity of the results but is not by itself sufficient. Many of the evaluation criteria require the application of expert judgement and

background knowledge for which consistency is more difficult to achieve. In order to enhance the consistency of the evaluation findings, the final evaluation results may be submitted to a certification process.

188 The certification process is the independent inspection of the results of the evaluation leading to the production of the final certificate or approval, which is normally publicly available. The certification process is a means of gaining greater consistency in the application of IT security criteria.

189 The evaluation schemes and certification processes are the responsibility of the evaluation authorities that run such schemes and processes and are outside the scope of the CC.

7 General model

190 This Chapter presents the general concepts used throughout the CC, including the context in which the concepts are to be used and the CC approach for applying the concepts. CC Part 2 and CC Part 3 expand on the use of these concepts and assume that the approach described is used. This Chapter assumes some knowledge of IT security and does not propose to act as a tutorial in this area.

191 The CC discusses security using a set of security concepts and terminology. An understanding of these concepts and the terminology is a prerequisite to the effective use of the CC. However, the concepts themselves are quite general and are not intended to restrict the class of IT security problems to which the CC is applicable.

7.1 Assets and countermeasures

192 Security is concerned with the protection of assets. Assets are entities that someone places value upon. Examples of assets include:

- contents of a file or a server;
- the authenticity of votes cast in an election;
- the availability of an electronic commerce process;
- the ability to use an expensive printer;
- access to a classified facility.

but given that value is highly subjective, almost anything can be an asset.

193 The environment(s) in which these assets are located is called the operational environment. Examples of (aspects of) operational environments are:

- the computer room of a bank;
- connected to the Internet;
- a LAN;
- a general office environment.

194 Many assets are in the form of information that is stored, processed and transmitted by IT products to meet requirements laid down by owners of the information. Information owners may require that availability, dissemination and modification of any such information is strictly controlled and that the assets are protected from threats by countermeasures. Figure 2 illustrates these high level concepts and relationships.

General model

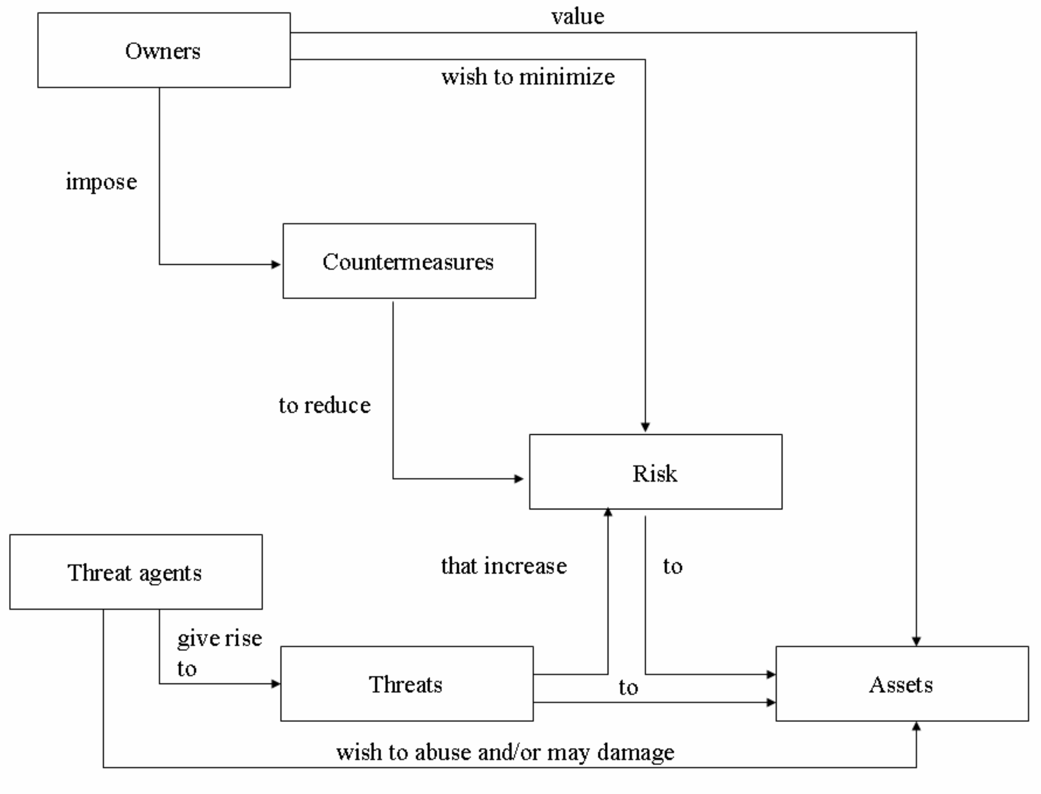

Figure 2 - Security concepts and relationships

195 Safeguarding assets of interest is the responsibility of owners who place value on those assets. Actual or presumed threat agents may also place value on the assets and seek to abuse assets in a manner contrary to the interests of the owner. Examples of threat agents include hackers, malicious users, non-malicious users (who sometimes make errors), computer processes and accidents.

196 The owners of the assets will perceive such threats as potential for impairment of the assets such that the value of the assets to the owners would be reduced. Security-specific impairment commonly includes, but is not limited to: loss of asset confidentiality, loss of asset integrity and loss of asset availability.

197 These threats therefore give rise to risks to the assets, based on the likelihood of a threat being realised and the impact on the assets when that threat is realised. Subsequently countermeasures are imposed to reduce the risks to assets. These countermeasures may consist of IT countermeasures (such as firewalls and smart cards) and non-IT countermeasures (such as guards and procedures).

198 Owners of assets may be (held) responsible for those assets and therefore should be able to defend the decision to accept the risks of exposing the assets to the threats.

199 Two important elements in defending this decision are being able to demonstrate that:

- the countermeasures are *sufficient*: if the countermeasures do what they claim to do, the threats to the assets are countered;

- the countermeasures are *correct*: the countermeasures do what they claim to do.

200 Many owners of assets lack the knowledge, expertise or resources necessary to judge sufficiency and correctness of the countermeasures, and they may not wish to rely solely on the assertions of the developers of the countermeasures. These consumers may therefore choose to increase their confidence in the sufficiency and correctness of some or all of their countermeasures by ordering an evaluation of these countermeasures.

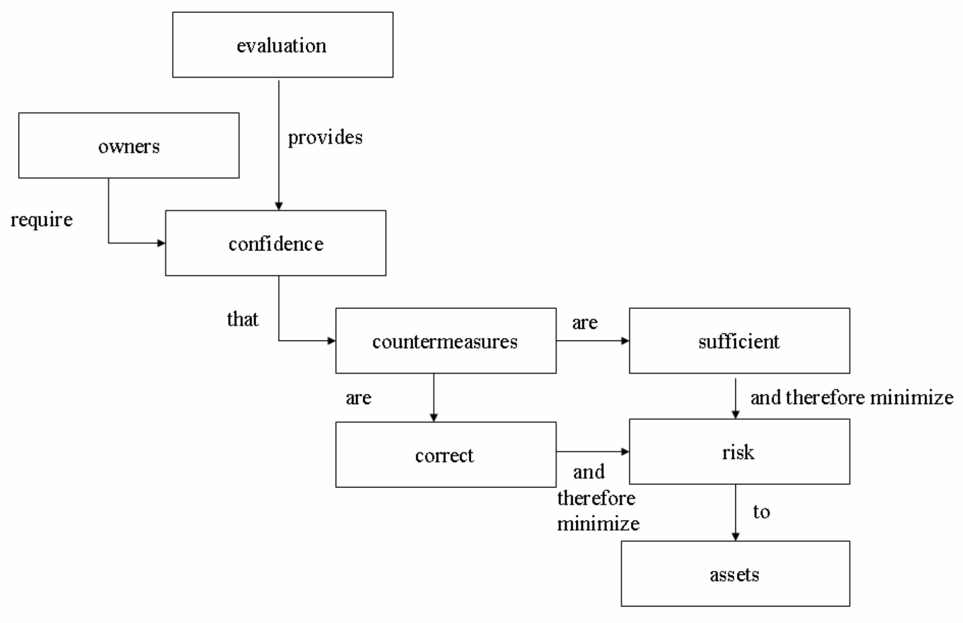

Figure 3 - Evaluation concepts and relationships

7.1.1 Sufficiency of the countermeasures

201 In an evaluation, sufficiency of the countermeasures is analysed through a construct called the Security Target. In this Section a simplified view on this construct is provided: a more detailed and complete description may be found in Annex A.

202 The Security Target begins with describing the assets and the threats to those assets. The Security Target then describes the countermeasures (in the form of Security Objectives) and demonstrates that these countermeasures are sufficient to counter these threats: if the countermeasures do what they claim to do, the threats are countered.

203 The Security Target then divides these countermeasures in two groups:

- the security objectives for the TOE: these describe the countermeasure(s) for which correctness will be determined in the evaluation;

- the security objectives for the Operational Environment: these describe the countermeasures for which correctness will not be determined in the evaluation;

204 The reasons for this division are:

– The CC is only suitable for assessing the correctness of IT-countermeasures. Therefore the non-IT countermeasures (e.g. human security guards, procedures) are always in the Operational Environment.

– Assessing correctness of countermeasures costs time and money, possibly making it infeasible to assess the correctness of all IT-countermeasures.

– The correctness of some IT-countermeasures may already have been assessed in another evaluation. It is therefore not cost-effective to assess this correctness again.

205 For the TOE (the IT-countermeasures whose correctness will be assessed during the evaluation), the Security Target requires a further detailing of the security objectives for the TOE in Security Functional Requirements (SFRs). These SFRs are formulated in a standardised language (described in CC Part 2) to ensure exactness and facilitate comparability.

206 In summary, the Security Target demonstrates that:

– The SFRs meet the security objectives for the TOE;

– The security objectives for the TOE and the security objectives for the operational environment counter the threats;

– And therefore, the SFRs and the security objectives for the operational environment counter the threats.

207 From this it follows that a correct TOE (meeting the SFRs) in combination with a correct operational environment (meeting the security objectives for the operational environment) will counter the threats. In the next two sections correctness of the TOE and correctness of the operational environment are discussed separately.

7.1.2 Correctness of the TOE

208 A TOE may be incorrectly designed and implemented, and may therefore contain errors that lead to vulnerabilities. By exploiting these vulnerabilities, attackers may still damage and/or abuse the assets.

209 These vulnerabilities may arise from accidental errors made during development, poor design, intentional addition of malicious code, poor testing etc.

210 To determine correctness of the TOE, various activities can be performed such as:

- testing the TOE;
- examining various design representations of the TOE;
- examining the physical security of the development environment of the TOE

211 The Security Target provides a structured description of these activities to determine correctness in the form of Security Assurance Requirements (SARs). These SARs are formulated in a standardised language (described in CC Part 3) to ensure exactness and facilitate comparability.

212 If the SARs are met, there exists assurance in the correctness of the TOE and the TOE is therefore less likely to contain vulnerabilities that can be exploited by attackers. The amount of assurance that exists in the correctness of the TOE is determined by the SARs themselves: a few “weak” SARs will lead to a little assurance, a lot of “strong” SARs will lead to a lot of assurance.

7.1.3 Correctness of the Operational Environment

213 The operational environment may also be incorrectly designed and implemented, and may therefore contain errors that lead to vulnerabilities. By exploiting these vulnerabilities, attackers may still damage and/or abuse the assets.

214 However, in the CC, no assurance is obtained regarding the correctness of the operational environment. Or, in other words, the operational environment is not evaluated (see the next Section).

215 As far as the evaluation is concerned, the operational environment is assumed to be a 100% correct instantiation of the security objectives for the operational environment.

216 This does not preclude a consumer of the TOE from using other methods to determine the correctness of his operational environment, such as:

- If, for an OS TOE, the security objectives for the operational environment state “The operational environment shall ensure that entities from an untrusted network (e.g. the Internet) can only access the TOE by ftp”, the consumer could select an evaluated firewall, and configure it to only allow ftp access to the TOE.
- If the security objectives for the operational environment state “The operational environment shall ensure that all administrative personnel will not behave maliciously”, the consumer could adapt his contracts with administrative personnel to include punitive sanctions for malicious behaviour.

but this determination is not part of a CC-evaluation.

7.2 Evaluation

217 The CC recognises two types of evaluation: an ST/TOE evaluation, which is described below, and a PP evaluation, which is described in more detail in Annex B. In many places, the CC uses the term evaluation (without qualifiers) to refer to an ST/TOE evaluation.

218 In the CC an ST/TOE evaluation proceeds in two steps:

- An ST evaluation: where the sufficiency of the TOE and the operational environment are determined;
- A TOE evaluation: where the correctness of the TOE is determined. As said earlier, the TOE evaluation does not assess correctness of the operational environment.

219 The ST evaluation is carried out by applying the Security Target evaluation criteria (which are defined in CC Part 3 Chapter ASE) to the Security Target. The precise method to apply the ASE criteria is determined by the evaluation methodology that is used.

220 The TOE evaluation is more complex. The principal inputs to a TOE evaluation are: the evaluation evidence, which includes the TOE and ST, but will usually also include input from the development environment, such as design documents or developer test results.

221 The TOE evaluation consists of applying the SARs (from the Security Target) to the evaluation evidence. The precise method to apply a specific SAR is determined by the evaluation methodology that is used.

222 How the results of applying the SARs are documented, and what reports need to be generated and in what detail, is determined by both the evaluation methodology that is used and the evaluation scheme under which the evaluation is carried out.

223 The result of the TOE evaluation process is either:

– A statement that not all SARs have been met and that therefore there is not enough assurance that the TOE meets the SFRs as stated in the ST;

– A statement that all SARs have been met, and that therefore there is enough assurance that the TOE meets the SFRs as stated in the ST.

224 The TOE evaluation may be carried out after TOE development has finished, or in parallel with TOE development.

225 The method of stating ST/TOE evaluation results is described in Chapter 9. These results also identify the PP(s) and package(s) to which the TOE claims conformance, and these constructs are described in the next Chapter.

8 Protection Profiles and Packages

8.1 Introduction

226 To allow consumer groups and communities of interest to express their security needs, and to facilitate writing STs, the CC provides two special constructs: packages and Protection Profiles (PPs). In the following two Sections these constructs are described in more detail, followed by a Section on how these constructs can be used.

8.2 Packages

227 A package is a named set of security requirements. A package is either

- a functional package, containing only SFRs, or
- an assurance package, containing only SARs.

Mixed packages containing both SFRs and SARs are not allowed.

228 A package can be defined by any party and is intended to be re-usable. To this goal it should contain requirements that are useful and effective in combination. Packages can be used in the construction of larger packages, PPs and STs. At present there are no criteria for the evaluation of packages, therefore any set of SFRs or SARs can be a package.

229 Examples of assurance packages are the evaluation assurance levels (EALs) that are defined in CC Part 3. At the time of writing there are no functional packages for this version of the CC.

8.3 Protection Profiles

230 Whereas an ST always describes a specific TOE (e.g. the MinuteGap v18.5 Firewall), a PP is intended to describe a TOE type (e.g. firewalls). The same PP may therefore be used as a template for many different STs to be used in different evaluations. A detailed description of PPs is given in Annex B.

231 In general an ST describes requirements for a TOE and is written by the developer of that TOE, while a PP describes the general requirements for a TOE type, and is therefore typically written by:

- A user community seeking to come to a consensus on the requirements for a given TOE type;
- A developer of a TOE, or a group of developers of similar TOEs wishing to establish a minimum baseline for that type of TOE;
- A government or large corporation specifying its requirements as part of its acquisition process.

232 PPs can be evaluated (by applying the APE criteria to them as listed in CC Part 3). The goal of such an evaluation is to demonstrate that the PP is complete, consistent, and technically sound and suitable for use as a template on which to build another PP or an ST.

233 Basing a PP/ST on an evaluated PP has two advantages:

- There is much less risk that there are errors, ambiguities or gaps in the PP. If any problems with a PP (that would have been caught by evaluating that PP) are found during the writing or evaluation of the new ST, significant time may elapse before the PP is corrected.
- Evaluation of the new PP/ST may often re-use evaluation results of the evaluated PP, resulting in less effort for evaluating the new PP/ST.

8.4 Using PPs and packages

234 If an ST claims to be conformant to one or more packages and/or Protection Profiles, the evaluation of that ST will (among other properties of that ST) demonstrate that the ST actually conforms to these packages and/or PPs that they claim conformance to. Details of this determination of conformance can be found in Annex A.

235 This allows the following process:

- An organisation seeking to acquire a particular type of IT security product develops their security needs into a PP, then has this evaluated and publishes it;
- A developer takes this PP, writes an ST that claims conformance to the PP and has this ST evaluated;
- The developer then builds a TOE (or uses an existing one) and has this evaluated against the ST.

236 The result is that the developer can prove that his TOE is conformant to the security needs of the organisation: the organisation can therefore acquire that TOE. A similar line of reasoning applies to packages.

8.5 Using Multiple Protection Profiles

237 The CC also allows PPs to conform to other PPs, allowing chains of PPs to be constructed, each based on the previous one(s).

238 For instance, one could take a PP for an Integrated Circuit and a PP for a Smart Card OS, and use these to construct a Smart Card PP (IC and OS) that claims conformance to the other two. One could then write a PP on Smart Cards for Public Transport based on the Smart Card PP and a PP on Applet Loading. Finally, a developer could then construct an ST based on this Smart Cards for Public Transport PP.

9 Evaluation results

9.1 Introduction

239 This Chapter presents the expected results from PP and ST/TOE evaluations.

- PP evaluations lead to catalogues of evaluated PPs.
- An ST evaluation leads to intermediate results that are used in the frame of a TOE evaluation.
- ST/TOE evaluations lead to catalogues of evaluated TOEs. In many cases these catalogues will refer to the IT products that the TOEs are derived from rather than the specific TOE. Therefore, the existence of an IT product in a catalogue should not be construed as meaning that the whole IT product has been evaluated; instead the actual extent of the ST/TOE evaluation is defined by the ST.

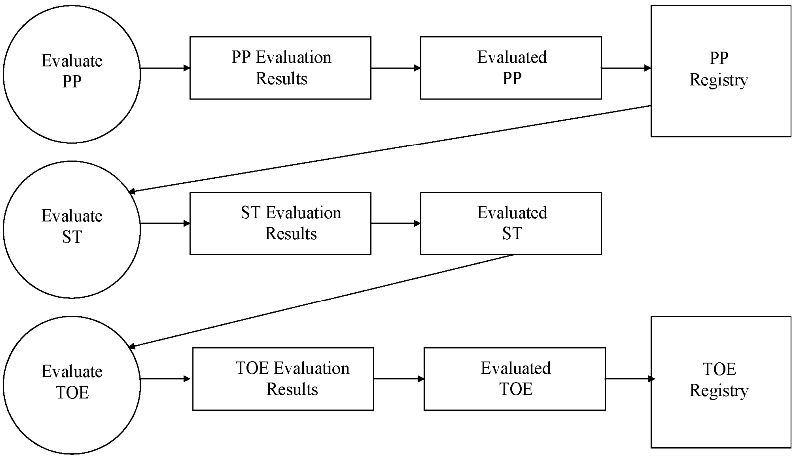

Figure 4 - Evaluation results

240 STs may be based on packages, evaluated PPs or non-evaluated PPs - however this is not mandatory, as STs do not have to be based on anything at all.

241 Evaluation should lead to objective and repeatable results that can be cited as evidence, even if there is no absolute objective scale for representing the results of a security evaluation. The existence of a set of evaluation criteria is a necessary pre-condition for evaluation to lead to a meaningful result and provides a technical basis for mutual recognition of evaluation results between evaluation authorities.

242 An evaluation result represents the findings of a specific type of investigation of the security properties of a TOE. Such a result does not automatically guarantee fitness for use in any particular application environment. The decision to accept a TOE for use in a specific application environment is

based on consideration of many security issues including the evaluation findings.

9.2 Results of a PP evaluation

243 The CC contains the evaluation criteria that permit an evaluator to state whether a PP is complete, consistent, and technically sound and hence suitable for use in developing an ST.

244 Evaluation of the PP shall result in a pass/fail statement. If the PP evaluation has resulted in a pass statement, the PP shall be eligible for inclusion within a registry. The results of the evaluation shall also include a “Conformance Claim” (see Section 9.4).

9.3 Results of an ST/TOE evaluation

245 The CC contains the evaluation criteria that enable an evaluator to determine whether sufficient assurance exists that the TOE satisfies the SFRs in the ST. Evaluation of the TOE shall therefore result in a pass/fail statement for the ST. If both the ST and the TOE evaluation have resulted in a pass statement, the underlying product is eligible for inclusion in a registry. The results of evaluation shall also include a “Conformance Claim” as defined in the next Section.

246 It may be the case that the evaluation results are subsequently used in a certification process, but this certification process is outside the scope of the CC.

9.4 Conformance claim

247 The conformance claim indicates the source of the collection of requirements that is met by a PP or ST that passes its evaluation. This conformance claim contains a CC conformance claim that:

- describes the version of the CC to which the PP or ST claims conformance.
- describes the conformance to CC Part 2 (security functional requirements) as either:
  - **CC Part 2 conformant -** A PP or ST is CC Part 2 conformant if all SFRs in that PP or ST are based only upon functional components in CC Part 2, or
  - **CC Part 2 extended** - A PP or ST is CC Part 2 extended if at least one SFR in that PP or ST is not based upon functional components in CC Part 2.
- describes the conformance to CC Part 3 (security assurance requirements) as either:

  - **CC Part 3 conformant** - A PP or ST is CC Part 3 conformant if all SARs in that PP or ST are based only upon assurance components in CC Part 3, or

  - **CC Part 3 extended** - A PP or ST is CC Part 3 extended if at least one SAR in that PP or ST is not based upon assurance components in CC Part 3.

248 Additionally, the conformance claim may include a statement made with respect to packages, in which case it consists of one of the following:

- *Package name Conformant* - A PP or ST is conformant to a pre-defined package (e.g. EAL) if:

  - the SFRs of that PP or ST are identical to the SFRs in the package, or

  - the SARs of that PP or ST are identical to the SARs in the package.

- *Package name Augmented* - A PP or ST is an augmentation of a predefined package if:

  - the SFRs of that PP or ST contain all SFRs in the package, but have at least one additional SFR or one SFR that is hierarchically higher than an SFR in the package.

  - the SARs of that PP or ST contain all SARs in the package, but have at least one additional SAR or one SAR that is hierarchically higher than an SAR in the package.

249 Note that when a TOE is successfully evaluated to a given ST, any conformance claims of the ST also hold for the TOE. A TOE can therefore also be e.g. CC Part 2 conformant.

250 Finally, the conformance claim may also include two statements with respect to Protection Profiles:

- *PP Conformant* - A PP or TOE meets specific PP(s), which are listed as part of the conformance result.

- *Conformance Statement* (Only for PPs) - This statement describes the manner in which PPs or STs must conform to this PP: strict or demonstrable. For more information on this Conformance Statement, see Annex A.

9.5 Use of ST/TOE evaluation results

251 Once an ST and a TOE have been evaluated, asset owners can have the assurance (as defined in the ST) that the TOE, together with the operational environment, counters the threats. The evaluation results may be used by the

asset owner in deciding whether to accept the risk of exposing the assets to the threats.

252 However, the asset owner should carefully check whether:

- the Security Problem Definition in the ST matches the security problem of the asset owner;
- the Operational Environment of the asset owner conforms (or can be made to conform) to the security objectives for the Operational Environment described in the ST.

253 If either of these is not the case, the TOE may not be suitable for the purposes of the asset owner.

254 Additionally, once an evaluated TOE is in operation, it is possible that previously unknown errors or vulnerabilities in the TOE may surface. In that case, the developer may correct the TOE (to repair the vulnerabilities) or the ST. However, the evaluation results of the old ST and TOE do not apply to the new ST and TOE.

255 If it is deemed necessary that confidence is regained, re-evaluation is needed. The CC may be used for this re-evaluation, but detailed procedures for re-evaluation are outside the scope of this document.

A Specification of Security Targets (normative)

A.1 Goal and structure of this Annex

256 The goal of this annex is to explain the Security Target (ST) concept. This annex does not define the ASE criteria; this definition can be found in CC Part 3.

257 This annex consists of four major parts:

- *What an ST must contain*. This is summarised in Section A.2, and described in more detail in Sections A.4 - A.10. These sections describe the mandatory contents of the ST, the interrelationships between these contents, and provide examples.
- *How an ST should be used*. This is summarised in Section A.3, and described in more detail in Section A.11. These sections describe how an ST should be used, and some of the questions that can be answered with an ST.
- *Low Assurance STs*. Low Assurance STs are STs with reduced content. They are described in detail in Section A.12.
- *Claiming compliance with standards*. Section A.13 describes how an ST writer can claim that the TOE meets a particular standard.

A.2 Mandatory contents of an ST

258 Figure 5 portrays the mandatory contents of an ST. Figure 5 may also be used as a structural outline of the ST, though alternative structures are possible. For instance, if the security requirements rationale is particularly bulky, it could be included in an appendix of the ST instead of in the security requirements section. The separate sections of an ST and the contents of those sections are briefly summarised below and described in much more detail in Sections A.4 to A.10. An ST normally contains:

- an *ST introduction* containing three narrative descriptions of the TOE on different levels of abstraction;
- a *conformance claim*, showing whether the ST claims conformance to any PPs and/or packages, and if so, to which PPs and/or packages;
- a *security problem definition*, showing the threats, OSPs and assumptions that must be countered, enforced and upheld by the TOE and its operational environment;

- *security objectives*, showing how the solution to the security problem is divided between security objectives for the TOE and security objectives for the operational environment of the TOE;

- *extended components definition*, where new components (i.e. not included in CC Part 2 or CC Part 3) may be defined. These new components are needed to define extended functional and extended assurance requirements;

- *security requirements*, where a translation of the security objectives for the TOE into a standardised language is provided. This standardised language is in the form of SFRs. Additionally this section defines the SARs;

- a *TOE summary specification*, showing how the SFRs are implemented in the TOE.

259 There also exists low assurance STs which have reduced contents; these are described in detail in Section A.12. The remainder of this Annex assumes that an ST with full contents is used.

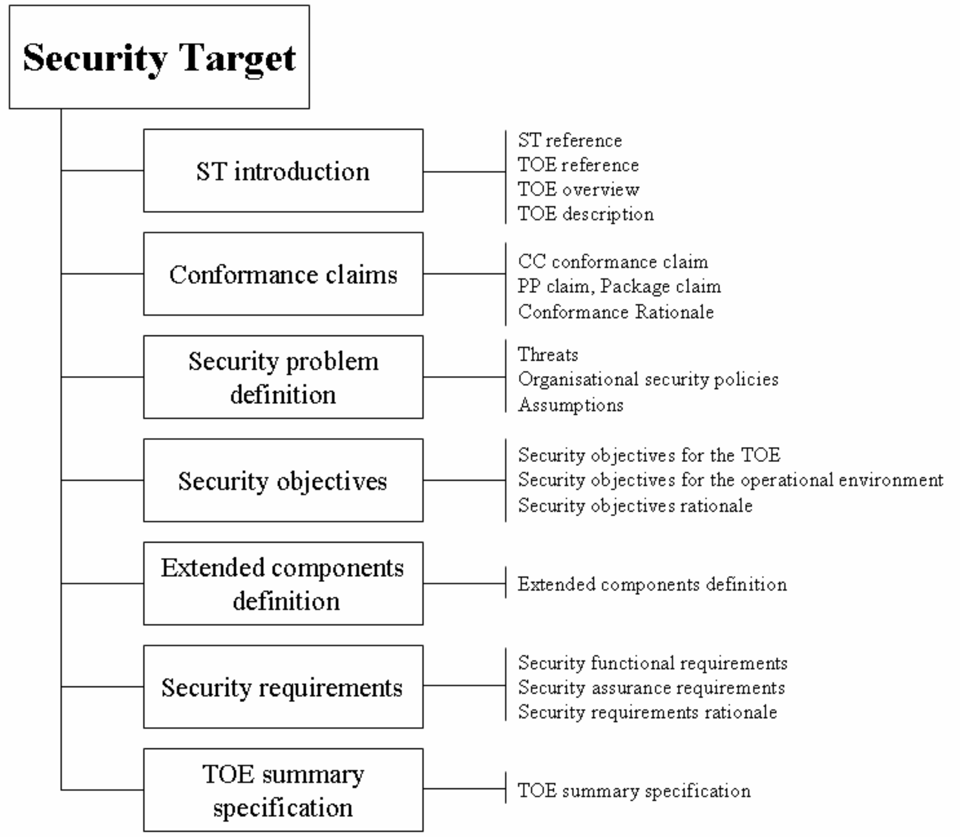
Security Target
ST introduction
ST reference
TOE reference
TOE overview
TOE description
Conformance claims
CC conformance claim
PP claim, Package claim
Conformance Rationale
Security problem definition
Threats
Organisational security policies
Assumptions
Security objectives
Security objectives for the TOE
Security objectives for the operational environment
Security objectives rationale
Extended components definition
Extended components definition
Security requirements
Security functional requirements
Security assurance requirements
Security requirements rationale
TOE summary specification
TOE summary specification

Figure 5 - Security Target contents

A.3 Using an ST

A.3.1 How an ST should be used

260 A typical ST fulfils two roles:

- Before and during the evaluation, the ST specifies "what is to be evaluated". In this role, the ST serves as a basis for agreement between the developer and the evaluator on the exact security properties of the TOE and the exact scope of the evaluation. Technical correctness and completeness are major issues for this role. Section A.7 describes how the ST should be used in this role.

- After the evaluation, the ST specifies "what was evaluated". In this role, the ST serves as a basis for agreement between the developer or re-seller of the TOE and the potential consumer of the TOE. The ST describes the exact security properties of the TOE in an abstract manner, and the potential consumer can rely on this description because the TOE has been evaluated to meet the ST. Ease of use and understandability are major issues for this role. Section A.11 describes how the ST should be used in this role.

A.3.2 How an ST should not be used

261 Two roles (among many) that an ST should not fulfil are:

- *a detailed specification*: An ST is designed to be a security specification on a relatively high level of abstraction. An ST should, in general, not contain detailed protocol specifications, detailed descriptions of algorithms and/or mechanisms, long description of detailed operations etc.

- *a complete specification*: An ST is designed to be a security specification and not a general specification. Unless security-relevant, properties such as interoperability, physical size and weight, required voltage etc. should not be part of an ST. This means that in general an ST may be a part of a complete specification, but is not a complete specification in itself.

A.4 ST Introduction (ASE_INT)

262 The ST introduction describes the TOE in a narrative way on three levels of abstraction:

- the ST reference and the TOE reference, which provide identification material for the ST and the TOE that the ST refers to;

- the TOE overview, which briefly describes the TOE;

- the TOE description, which describes the TOE in more detail.

A.4.1 ST reference and TOE reference

263 An ST contains a clear ST reference that identifies that particular ST. A typical ST reference consists of title, version, authors and publication date. An example of an ST reference is “MauveRAM Database ST, version 1.3, MauveCorp Specification Team, 11 October 2002”. The reference must be unique so that it is possible to distinguish between different STs and different versions of the same ST.

264 An ST also contains a TOE reference that identifies the TOE that claims conformance to the ST. A typical TOE reference consists of developer name, TOE name and TOE version number. An example of a TOE reference is “MauveCorp MauveRAM Database v2.11”. As a single TOE may be evaluated multiple times, for instance by different consumers of that TOE, and therefore have multiple STs, this reference is not necessarily unique.

265 If the TOE is constructed from one or more well-known Products, it is allowed to reflect this in the TOE reference, by referring to the Product name(s). However, this should not be used to mislead consumers: situations where major parts or security functionalities were not considered in the evaluation, yet the TOE reference does not reflect this are not allowed.

266 The ST reference and the TOE reference facilitate indexing and referencing the ST and TOE and their inclusion in summaries of lists of evaluated TOEs/Products.

A.4.2 TOE overview

267 The TOE overview is aimed at potential consumers of a TOE who are looking through lists of evaluated TOEs/Products to find TOEs that may meet their security needs, and are supported by their hardware, software and firmware. The typical length of a TOE overview is several paragraphs.

268 To this end, the TOE overview briefly describes the usage of the TOE and its major security features, identifies the TOE type and identifies any major non-TOE hardware/software/firmware required by the TOE.

A.4.2.1 Usage and major security features of a TOE

269 The description of the usage and major security features of the TOE is intended to give a very general idea of what the TOE is capable of in terms of security, and what it can be used for in a security context. This section should be written for (potential) TOE consumers, describing TOE usage and major security features in terms of business operations, using language that TOE consumers understand.

270 An example of this is “The MauveCorp MauveRAM Database v2.11 is a multi-user database intended to be used in a networked environment. It allows 1024 users to be active simultaneously. It allows password/token and biometric authentication, protects against accidental data corruption, and can roll-back ten thousand transactions. Its audit features are highly configurable,

so as to allow detailed audit to be performed for some users and transactions, while protecting the privacy of other users and transactions.”

A.4.2.2 TOE type

271 The TOE overview identifies the general type of TOE, such as: firewall, VPN-firewall, smart card, crypto-modem, intranet, web server, database, web server and database, LAN, LAN with web server and database, etc.

272 It may be the case that the TOE is not of a readily available type, in which case “none” would be acceptable.

273 In some cases, a TOE type can mislead consumers. Examples include:

- certain functionality can be expected of the TOE because of its TOE type, but the TOE does not have this functionality. Examples include:
  - an ATM-card type TOE, which does not support any identification/authentication functionality;
  - a firewall type TOE, which does not support protocols that are almost universally used;
  - a PKI-type TOE, which has no certificate revocation functionality.
- the TOE can be expected to operate in certain operational environments because of its TOE type, but it cannot do so. Examples include:
  - a PC-operating system type TOE, which is unable to function securely unless the PC has no network connection, floppy drive, and CD/DVD-player;
  - a firewall, which is unable to function securely unless all users that can connect through that firewall are benign.

274 In these cases, the TOE overview must contain additional information to ensure that potential consumers are not misled.

A.4.2.3 Required non-TOE hardware/software/firmware

275 While some TOEs do not rely upon other IT, many TOEs (notably software TOEs) rely on additional, non-TOE, hardware, software and/or firmware. In the latter case, the TOE overview is required to identify this non-TOE hardware/software/firmware.

276 It is not required to provide a complete and fully detailed identification of all this hardware/software/firmware, but the identification should be complete and detailed enough for potential consumers to determine the major hardware/software/firmware components needed to use the TOE.

277 Example hardware/software/firmware identifications are:

- a standard PC with a 1GHz or higher processor and 512MB or more RAM, running version 3.0 Update 6b, c, or 7, or version 4.0 of the Yaiza operating system;

- a standard PC with a 1GHz or higher processor and 512MB or more RAM, running version 3.0 Update 6d of the Yaiza operating system and the WonderMagic 1.0 Graphics card with the 1.0 WM Driver Set;

- a standard PC with version 3.0 of the Yaiza OS (or higher);

- a CleverCard SB2067 integrated circuit;

- a CleverCard SB2067 integrated circuit running v2.0 of the QuickOS smart card operating system;

- the December 2002 installation of the LAN of the Director-General's Office of the Department of Traffic.

A.4.3 TOE description

278 A TOE description is a narrative description of the TOE, likely to run to several pages. The TOE description should provide evaluators and potential consumers with a general understanding of the security capabilities of the TOE, in more detail than was provided in the TOE overview. The TOE description may also be used to describe the wider application context into which the TOE will fit.

279 The TOE description discusses the physical scope of the TOE: a list of all hardware, firmware, software and guidance parts that constitute the TOE. This list should be described at a level of detail that is sufficient to give the reader a general understanding of those parts.

280 The TOE description should also discuss the logical scope of the TOE: the logical security features offered by the TOE at a level of detail that is sufficient to give the reader a general understanding of those features. This description is expected to be in more detail than the major security features described in the TOE overview.

281 An important property of the physical and logical scopes is that they describe the TOE in such a way that there remains no doubt on whether a certain part or feature is in the TOE or whether this part or feature is outside the TOE. This is especially important when the TOE is intertwined with and cannot be easily separated from non-TOE entities.

282 Examples where the TOE is intertwined with non-TOE entities are:

- the TOE is a cryptographic co-processor of a smart card IC, instead of the entire IC;

- the TOE is a smart card IC, except for the cryptographic processor;
- the TOE is the Network Address Translation part of the MinuteGap Firewall v18.5.

A.5 Conformance claims (ASE_CCL)

283 This section of an ST describes how the ST conforms with:

- the Common Criteria itself
- Protection Profiles (if any)
- Packages (if any)

This description shall be structured in accordance with Section 9.4.

284 The description of how the ST conforms to the CC consists of two items: the version of the CC that is used and whether the ST contains extended security requirements or not (see Section A.8).

285 The description of conformance of the ST to Protection Profiles means that the ST lists the Protection Profiles that conformance is being claimed to. For a general explanation of this, see Section 9.4. For a detailed description, see Annex D.

286 The description of conformance of the ST to packages means that the ST lists the packages that conformance is being claimed to. For an explanation of this, see Section 9.4.

A.6 Security problem definition (ASE_SPD)

A.6.1 Introduction

287 The security problem definition defines the security problem that is to be addressed. The security problem definition is, as far as the CC is concerned, axiomatic. That is, the process of deriving the security problem definition falls outside the scope of the CC.

288 However, it should be noted that the usefulness of the results of an evaluation strongly depends on the ST, and the usefulness of the ST strongly depends on the quality of the security problem definition. It is therefore often worthwhile to spend significant resources and use well-defined processes and analyses to derive a good security problem definition.

289 Note that it is not mandatory to have statements in all sections, an ST can have no threats, or no OSPs, or no assumptions. However, if an ST has no threats, it must have OSPs, and if an ST has no OSPs it must have threats.

290 Also note that where the TOE is physically distributed, it may be better to discuss the relevant threats, OSPs and assumptions separately for distinct domains of the TOE operational environment.

A.6.2 Threats

291 This section of the security problem definition shows the threats that are to be countered by the TOE, its operational environment, or a combination of the two.

292 A threat consists of a threat agent, an asset and an adverse action of that threat agent on that asset.

293 *Threat agents* are entities that can adversely act on assets. Examples of threat agents are hackers, users, computer processes, TOE development personnel, and accidents. Threat agents may be further described by aspects such as expertise, resources, opportunity and motivation.

294 Threat agents may be described as individual entities, but in some cases it may be better to describe them as types of entities, groups of entities etc.

295 Examples of *assets* can be found in Section 7.1.

296 *Adverse actions* are actions performed by a threat agent on an asset. These actions influence one or more properties of an asset from which that asset derives its value.

297 Examples of threats are:

- a hacker (with substantial expertise, standard equipment, and being paid to do so) remotely copying confidential files from a company network;
- a worm seriously degrading the performance of a wide-area network;
- a system administrator violating user privacy;
- Someone on the Internet listening in on confidential electronic communication.

A.6.3 Organisational security policies (OSPs)

298 This section of the security problem definition shows the OSPs that are to be enforced by the TOE, its operational environment, or a combination of the two.

299 OSPs are security rules, procedures, or guidelines imposed (or presumed to be imposed) now and/or in the future by an actual or hypothetical organisation in the operational environment. OSPs may be laid down by an organisation controlling the operational environment of the TOE, or they may be laid down by legislative or regulatory bodies. OSPs can apply to the TOE and/or the operational environment of the TOE.

300 Examples of OSPs are:

- All products that are used by the Government must conform to the National Standard for password generation and encryption;
- Only users with System Administrator privilege and clearance of Department Secret shall be allowed to manage the Department Fileserver.

A.6.4 Assumptions

301 This section of the security problem definition shows the assumptions that are made on the operational environment in order to be able to provide security functionality. If the TOE is placed in an operational environment that does not meet these assumptions, the TOE may not be able to provide all of its security functionality anymore. Assumptions can be on physical, personnel and connectivity of the operational environment.

302 Examples of assumptions are:

- Assumptions on physical aspects of the operational environment:
  - It is assumed that the TOE will be placed in a room that is designed to minimise electromagnetic emanations;
  - It is assumed that the administrator consoles of the TOE will be placed in a restricted access area.
- Assumptions on personnel aspects of the operational environment:
  - It is assumed that users of the TOE will be trained sufficiently in order to operate the TOE;
  - It is assumed that users of the TOE are approved for information that is classified as National Secret;
  - It is assumed that users of the TOE will not write down their passwords.
- Assumptions on connectivity aspects of the operational environment:
  - It is assumed that a PC workstation with at least 10GB of disk space is available to run the TOE on;
  - It is assumed that the TOE is the only non-OS application running on this workstation;
  - It is assumed that the TOE will not be connected to an untrusted network.

303 Note that during the evaluation these assumptions are considered to be true: they are not tested in any way. For these reasons, assumptions can only be made on the operational environment. Assumptions can never be made on

the behaviour of the TOE because an evaluation consists of evaluating assertions made about the TOE and not by assuming that assertions on the TOE are true.

A.7 Security objectives (ASE_OBJ)

304 The security objectives are a concise and abstract statement of the intended solution to the problem defined by the security problem definition. The role of the security objectives is threefold:

- provide a high-level, natural language solution of the problem;
- divide this solution into two part wise solutions, that reflect that different entities each have to address a part of the problem;
- demonstrate that these part wise solutions form a complete solution to the problem.

A.7.1 High-level solution

305 The security objectives consist of a set of short and clear statements without overly much detail that together form a high-level solution to the security problem. The level of abstraction of the security objectives aims at being clear and understandable to knowledgeable potential consumers of the TOE. The security objectives are in natural language. A more exact, standardised description of some of the security objectives will be provided as part of the security requirements, which are described later on in this Annex.

A.7.2 Part wise solutions

306 In an ST the high-level security solution, as described by the security objectives, is divided into two part wise solutions. These part wise solutions are called the security objectives for the TOE and the security objectives for the operational environment. This reflects that these part wise solutions are to be provided by two different entities: the TOE, and the operational environment.

A.7.2.1 Security objectives for the TOE

307 The TOE provides security functionality to solve a certain part of the problem defined by the security problem definition. This part wise solution is called the security objectives for the TOE and consists of a set of objectives that the TOE should achieve in order to solve its part of the problem.

308 Examples of security objectives for the TOE are:

- The TOE shall keep confidential the content of all files transmitted between it and a Server;
- The TOE shall identify and authenticate all users before allowing them access to the Transmission Service provided by the TOE;

- The TOE shall restrict user access to data according to the Data Access policy described in Annex 3 of the ST.

309 If the TOE is physically distributed, it may be better to subdivide the ST section containing the security objectives for the TOE into several subsections to reflect this.

A.7.2.2 Security objectives for the operational environment

310 The operational environment of the TOE implements technical and procedural measures to assist the TOE in correctly providing its security functionality (which is defined by the security objectives for the TOE). This part wise solution is called the security objectives for the operational environment and consists of a set of statements describing the goals that the operational environment should achieve.

311 Examples of security objectives for the operational environment are:

- The operational environment shall provide a workstation with the OS Inux version 3.01b to execute the TOE on;
- The operational environment shall ensure that all human TOE users receive appropriate training before allowing them to work with the TOE;
- The operational environment of the TOE shall restrict physical access to the TOE to administrative personnel and maintenance personnel accompanied by administrative personnel;
- The operational environment shall ensure the confidentiality of the audit logs generated by the TOE before sending them to the central Audit Server.

312 If the operational environment of the TOE consists of multiple sites, each with different properties, it may be better to subdivide the ST section containing the security objectives for the operational environment into several subsections to reflect this.

A.7.3 Relation between security objectives and the security problem definition

313 The ST also contains a security objectives rationale containing two sections:

- a tracing that shows which security objectives address which threats, OSPs and assumptions;
- a set of justifications that shows that all threats, OSPs, and assumptions are effectively addressed by the security objectives.

A.7.3.1 Tracing between security objectives and the security problem definition

314 The tracing shows how the security objectives trace back to the threats, OSPs and assumptions as described in the security problem definition. This tracing must obey three rules:

- *No spurious objectives*: Each security objective traces to at least one threat, OSP or assumption.
- *Complete with respect to the security problem definition*: Each threat, OSP and assumption has at least one security objective tracing to it.
- *Correct tracing*: Since assumptions are always made by the TOE on the operational environment, security objectives for the TOE do not trace back to assumptions. The allowed tracings are depicted in Figure 6.

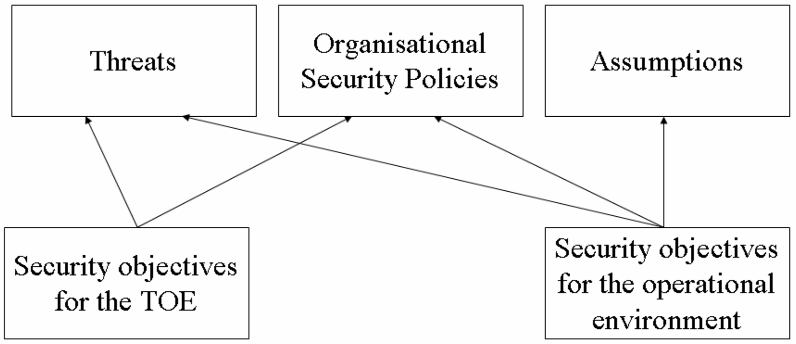
Threats
Organisational Security Policies
Assumptions
Security objectives for the TOE
Security objectives for the operational environment

Figure 6 - Allowed tracings between security objectives and security problem definition

315 Multiple security objectives may trace to the same threat, indicating that the combination of those security objectives counters that threat. A similar argument holds for OSPs and assumptions.

A.7.3.2 Providing a justification for the tracing

316 The security objectives rationale also demonstrates that the tracing is effective: if all security objectives tracing to a particular threat/OSP/assumption are achieved, that threat/OSP/assumption is countered/enforced/upheld.

317 This demonstration analyses the effect of achieving the relevant security objectives on countering the threats, enforcing the OSPs and upholding the assumptions and leads to the conclusion that this is indeed the case.

318 In some cases, where parts of the security problem definition very closely resemble some security objectives, the demonstration can be very simple. An example is: a threat “T17: Threat agent X reads the Confidential Information in transit between A and B”, a security objective for the TOE: “OT12: The

TOE shall ensure that all information transmitted between A and B is kept confidential", and a demonstration "T17 is directly countered by OT12".

A.7.3.3 On countering threats

319 Countering a threat does not necessarily mean removing that threat, it can also mean sufficiently diminishing that threat or sufficiently mitigating that threat.

320 Examples of removing a threat are:

- removing the ability to execute the adverse action from the threat agent;
- moving, changing or protecting the asset in such a way that the adverse action is no longer applicable to it;
- removing the threat agent (e.g. removing machines from a network that frequently crash that network).

321 Examples of diminishing a threat are:

- restricting the ability of a threat agent to perform adverse actions;
- restricting the opportunity to execute an adverse action of a threat agent;
- reducing the likelihood of an executed adverse action being successful;
- reducing the motivation to execute an adverse action of a threat agent by deterrence;
- requiring greater expertise or greater resources from the threat agent.

322 Examples of mitigating the effects of a threat are:

- making frequent back-ups of the asset;
- obtaining spare copies of an asset;
- insuring an asset;
- ensuring that successful adverse actions are always timely detected, so that appropriate action can be taken.

A.7.4 Security objectives: conclusion

323 Based on the security objectives and the security objectives rationale, the following conclusion can be drawn: if all security objectives are achieved then the security problem as defined in ASE_SPD is solved: all threats are countered, all OSPs are enforced, and all assumptions are upheld.

A.8 Extended Components Definition (ASE_ECD)

324 In many cases the security requirements (see the next Section) in an ST are based on components in CC Part 2 or CC Part 3. However, in some cases, there may be requirements in an ST that are not based on components in CC Part 2 or CC Part 3. In this case, new components (extended components) must be defined, and this definition should be done in the Extended Components Definition. For more information on this, see Annex C.5

325 Note that this section is intended to contain only the extended components and not the extended requirements (requirements based on extended components). The extended requirements should be included in the security requirements (see the next Section) and are for all purposes the same as requirements based on components in CC Part 2 or CC Part 3.

A.9 Security requirements (ASE_REQ)

326 The security requirements consists of two groups of requirements:

- *the security functional requirements (SFRs)*: a translation of the security objectives for the TOE into a standardised language;
- *the security assurance requirements (SARs)*: a description of how assurance is to be gained that the TOE meets the SFRs.

These two groups are discussed in the following two sections:

A.9.1 Security functional requirements (SFRs)

327 The SFRs are a translation of the security objectives for the TOE. They are usually at a more detailed level of abstraction, but they have to be a complete translation (the security objectives must be completely addressed). The CC requires this translation into a standardised language for several reasons:

- to provide an exact description of what is to be evaluated. As security objectives for the TOE are usually formulated in natural language, translation into a standardised language enforces a more exact description of the functionality of the TOE.
- to allow comparison between two STs. As different ST authors may use different terminology in describing their security objectives, the standardised language enforces using the same terminology and concepts. This allows easy comparison.

328 There is no translation required in the CC for the security objectives for the operational environment, because the operational environment is not evaluated and does therefore not require a description aimed at its evaluation.

329 It may be the case that parts of the operational environment are evaluated in another evaluation, but this is out of scope for the current evaluation. For example: an OS TOE may require a firewall to be present in its operational

environment. Another evaluation may subsequently evaluate the firewall, but this evaluation has nothing to do with the evaluation of the OS TOE.

A.9.1.1 How the CC supports this translation

330 The CC supports this translation in three ways: ways:

- by providing a predefined precise “language” designed to describe exactly what is to be evaluated. This language is defined as a set of components defined in CC Part 2. The use of this language as a well-defined translation of the security objectives for the TOE to SFRs is mandatory, though some exceptions exist (see C.5).

- by providing operations: mechanisms that allow the ST writer to modify the SFRs to provide a more accurate translation of the security objectives for the TOE. The CC has four operations: assignment, selection, iteration, and refinement. These are described further in Section C.4.4.

- by providing dependencies: a mechanism that supports a more complete translation to SARs. In the CC Part 2 language, an SFR can have a dependency on other SFRs. This signifies that if an ST uses that SFR, it generally needs to use those other SFRs as well. This makes it much harder for the ST writer to overlook including necessary SFRs and thereby improves the completeness of STs. Dependencies are described further in Annex C.3.

A.9.1.2 Relation between SFRs and security objectives

331 The ST also contains a security requirements rationale, consisting of two sections:

- a tracing that shows which SFRs address which security objectives for the TOE;

- a set of justifications that shows that all security objectives for the TOE are effectively addressed by the SFRs.

A.9.1.2.1 Tracing between SFRs and the security objectives for the TOE

332 The tracing shows how the SFRs trace back to the security objectives for the TOE. This tracing must obey two rules:

- *No spurious SFRs*: Each SFR traces back to at least one security objective.

- *Complete with respect to the security objectives for the TOE*: Each security objective for the TOE has at least one SFR tracing to it.

333 Multiple SFRs may trace to the same security objective for the TOE, indicating that the combination of those security requirements meets that security objective for the TOE.

A.9.1.2.2 Providing a justification for the tracing

334 The security requirements rationale must also demonstrate that the tracing is effective: if all SFRs tracing to a particular security objective for the TOE are satisfied, that security objective for the TOE is achieved.

335 This demonstration should analyse the effects of satisfying the relevant SFRs on achieving the security objective for the TOE and lead to the conclusion that this is indeed the case.

336 In cases where SFRs very closely resemble security objectives for the TOE, the demonstration can be very simple.

A.9.1.3 Security assurance requirements (SARs)

337 The SARs are a description of how the TOE is to be evaluated. This description uses a standardised language for two reasons:

- to provide an exact description of how the TOE is to be evaluated. Using a standardised language assists in creating an exact description and avoids ambiguity.
- to allow comparison between two STs. As different ST authors may use different terminology in describing the evaluation, the standardised language enforces using the same terminology and concepts. This allows easy comparison.

338 This standardised language is defined as a set of components defined in CC Part 3. The use of this language is mandatory, though some exceptions exist (see Annex C.5).The CC enhances this languages in two ways:

- by providing operations: mechanisms that allow the ST writer to modify the SARs to provide a more accurate translation of the security objectives for the TOE and the development environment. The CC has four operations: assignment, selection, iteration, and refinement. These are described further in Section C.4.4.
- by providing dependencies: a mechanism that supports a more complete translation to SARs. In the CC Part 3 language, an SAR can have a dependency on other SARs. This signifies that if an ST uses that SAR, it generally needs to use those other SARs as well. This makes it much harder for the ST writer to overlook including necessary SARs and thereby improves the completeness of STs. Dependencies are described further in Annex C.3.

A.9.1.4 SARs and the security requirement rationale

339 The ST also contains a security requirements rationale that explains why this particular set of SARs was deemed appropriate. There are no specific requirements for this explanation: the explanation could range from “None” to “because the PP or a national law requires it” to a detailed risk analysis of the TOE and the development environment of the TOE. The goal for this explanation is to allow the readers of the ST to understand the reasons why this particular set was chosen.

340 Note that the SARs shall still be consistent with the remainder of the ST. An example of an inconsistency is if the security problem description mentions threats where the threat agent is very capable, and a low (or no) AVA_VAN is included in the SARs.

A.9.2 Security requirements: conclusion

341 In the security problem definition of the ST, the security problem is defined as consisting of threats, OSPs and assumptions. In the security objectives section of the ST, the solution is provided in the form of two sub-solutions:

– security objectives for the TOE;

– security objectives for the operational environment.

342 Additionally, a security objectives rationale is provided showing that if all security objectives are achieved, the security problem is solved: all threats are countered, all OSPs are enforced, and all assumptions are upheld.

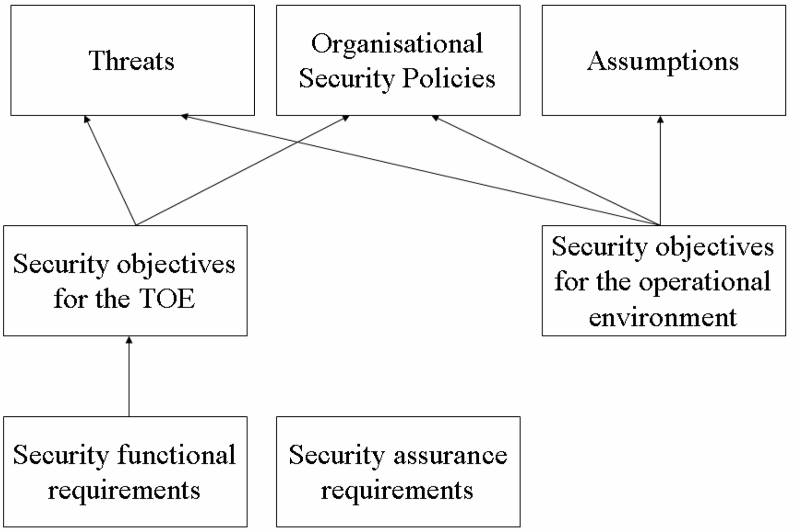
Threats
Organisational Security Policies
Assumptions
Security objectives for the TOE
Security objectives for the operational environment
Security functional requirements
Security assurance requirements

Figure 7 - Relations between the security problem definition, the security objectives and the security requirements

343 In the security requirements section of the ST, the security objectives for the TOE are translated to SFRs and a security requirements rationale is provided

showing that if all SFRs are satisfied, all security objectives for the TOE are achieved.

344 Additionally, a set of SARs is provided to show how the TOE is evaluated, together with an explanation for selecting these SARs.

345 All of the above can be combined into the statement: If all SFRs and SARs are satisfied and all security objectives for the operational environment are achieved, then there exists assurance that the security problem as defined in ASE_SPD is solved: all threats are countered, all OSPs are enforced, and all assumptions are upheld. This is illustrated in Figure 7.

346 The amount of assurance obtained is defined by the SARs, and whether this amount of assurance is sufficient is defined by the explanation for choosing these SARs.

A.10 TOE summary specification (ASE_TSS)

347 The objective for the TOE summary specification is to provide potential consumers of the TOE with a description of how the TOE satisfies all the SFRs. The TOE summary specification should provide the general technical mechanisms that the TOE uses for this purpose. The level of detail of this description should be enough to enable potential consumers to understand the general form and implementation of the TOE.

348 For instance if the TOE is an Internet PC and the SFRs contain FIA_UAU.1 to specify authentication, the TOE summary specification should indicate how this authentication is done: password, token, iris scanning etc. More information, like applicable standards that the TOE uses to meet SFRs, or more detailed descriptions may also be provided.

A.11 Questions that may be answered with an ST

349 After the evaluation, the ST specifies "what was evaluated". In this role, the ST serves as a basis for agreement between the developer or re-seller of the TOE and the potential consumer of the TOE. The ST can therefore answer the following questions (and more):

- *How can I find the ST/TOE that I need given the multitude of existing STs/TOEs?* This question is addressed by the TOE overview, which gives a brief (several paragraphs) summary of the TOE;

- *Does this TOE fit in with my existing IT-infrastructure?* This question is addressed by the TOE overview, which identifies the major hardware/firmware/software elements needed to run the TOE;

- *Does this TOE fit in with my existing operational environment?* This question is addressed by the security objectives for the operational environment, which identifies all constraints the TOE places on the operational environment in order to function;

- *What does the TOE do (interested reader)?* This question is addressed by the TOE overview, which gives a brief (several paragraphs) summary of the TOE;
- *What does the TOE do (potential consumer)?* This question is addressed by the TOE description, which gives a less brief (several pages) summary of the TOE;
- *What does the TOE do (technical)?* This question is addressed by the TOE summary specification which provides a high-level description of the mechanisms the TOE uses;
- *What does the TOE do (expert)?* This question is addressed by the SFRs which provide an abstract highly technical description, and the TOE summary specification which provide additional detail;
- *Does the TOE address the problem as defined by my government/organisation?* If your government/organisation has defined packages and/or PPs to define this solution, then the answer can be found in the Conformance Claims section of the ST, which lists all packages and PPs that the ST conforms to.
- *Does the TOE address my security problem (expert)?* What are the threats countered by the TOE? What organisational security policies does it enforce? What assumptions does it make about the operational environment? These questions are addressed by the security problem definition;
- *How much trust can I place in the TOE?* This can be found in the SARs in the security requirements section, which provide the assurance level that was used to evaluate the TOE, and hence the trust that the evaluation provides in the correctness of the TOE.

A.12 Low assurance Security Targets

350 Writing an ST is not a trivial task, and may, especially in low assurance evaluations, be a major part of the total effort expended by the developer and the evaluator in the whole of the evaluation. For this reason, it is also possible to write a low assurance ST.

351 The CC allows the use of a low assurance ST for an EAL 1 evaluation, but not for EAL 2 and up. A low-assurance ST may only claim conformance to a low-assurance PP (see Annex B). A non-low assurance ST may claim conformance with a low assurance PP.

352 A low assurance ST has a significantly reduced content compared to a non-low assurance ST:

- – there is no need to describe the security problem definition (threats, OSPs and assumptions that the TOE must counter, enforce and uphold);

- there is no need to describe the security objectives for the TOE. The security objectives for the operational environment shall still be described;
- there is no need to describe the security objectives rationale as there is no security problem definition in the ST;
- the security requirements rationale only needs to justify (any) dependencies not being satisfied as there are no security objectives for the TOE in the ST.

353 All that remains are:

- the references to TOE and ST
- the conformance claim
- the various narrative descriptions
  1. the TOE overview
  2. the TOE description
  3. the TOE summary specification
- the security objectives for the operational environment
- the SFRs and the SARs (including the extended components definition) and the security requirements rationale (only if the dependencies are not satisfied).

354 The reduced content of a low assurance ST is shown in Figure 8.

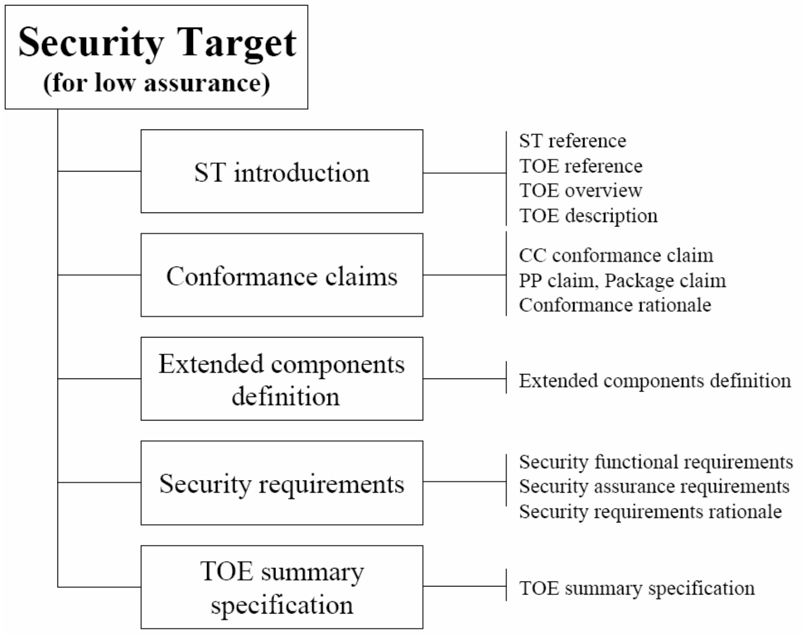
Security Target
(for low assurance)
ST introduction
ST reference
TOE reference
TOE overview
TOE description
Conformance claims
CC conformance claim
PP claim, Package claim
Conformance rationale
Extended components definition
Extended components definition
Security requirements
Security functional requirements
Security assurance requirements
Security requirements rationale
TOE summary specification
TOE summary specification

Figure 8 - Contents of a Low Assurance Security Target

A.13 Referring to other standards in an ST

355 In some cases, an ST writer may wish to refer to an external standard, such as a particular cryptographic standard or protocol. The CC allows three ways of doing this:

- As an organisational security policy (or part of it).

  If, for example, there exists a government standard defining how passwords have to be chosen, this may be stated as an organisational security policy in an ST. This may lead to an objective for the environment (e. g. if users of the TOE need to choose passwords accordingly), or it may lead to security objectives for the TOE and then to appropriate SFRs (likely of the FIA class), if the TOE generates passwords. In both cases the rationale of the developer needs to make plausible that the security objectives for the TOE and the SFRs are suitable to fulfil the OSP. The evaluator will examine if this is in fact plausible (and may decide to look into the standard for this), if the OSP is implemented by SFRs, as explained below.

- As a technical standard (for example a cryptographic standard) used in a refinement of an SFR.

  In this case conformance to the standard is part of the fulfilment of the SFR by the TOE and is treated as if the full text of the standard is part of the SFR. Conformance is subsequently determined like any other conformance to SFRs: during ADV and ATE it is analysed, by design analysis and tests, that the SFR is completely and fully implemented in the TOE. If reference to only a certain part of a standard is desired, that part should be unambiguously stated in the SFR refinement.

- As a technical standard (for example a cryptographic standard) mentioned in the TOE summary specification.

  The TOE summary specification is only considered as an explanation of how the SFRs are realised, and is not strictly used as a strict implementation requirement like the SFRs or the documents delivered for ADV. So the evaluator may detect an inconsistency if the TSS references a technical standard and this is not reflected in ADV documentation, but there is no routine activity to test fulfilment of the standard.

B Specification of Protection Profiles (normative)

B.1 Goal and structure of this Annex

356 The goal of this Annex is to explain the Protection Profile (PP) concept. This Annex does not define the APE criteria; this definition can be found in CC Part 3.

357 As PPs and STs have a significant overlap, this Annex focuses on the differences between PPs and STs. The material that is identical between STs and PPs is described in Annex A.

358 This annex consists of four major parts:

- *What a PP must contain*. This is summarised in Section B.2, and described in more detail in Sections B.4-B.9. These sections describe the mandatory contents of the PP, the interrelationships between these contents, and provide examples.
- *How a PP should be used*. This is summarised in Section B.3.
- *Low Assurance PPs*. Low Assurance PPs are PPs with reduced content. They are described in detail in Section B.11.
- *Claiming compliance with standards*. Section B.12 describes how a PP writer can claim that the TOE is to meet a particular standard.

B.2 Mandatory contents of a PP

359 Figure 9 portrays the mandatory content for a PP. Figure 9 may also be used as a structural outline of the PP, though alternative structures are allowed. For instance, if the security requirements rationale is particularly bulky, it could be included in an appendix of the PP instead of in the security requirements section. The separate sections of a PP and the contents of those sections are briefly summarised below and described in much more detail in Sections B.4 - B.9. A PP must contain:

- a PP *introduction* containing a narrative description of the TOE type;
- a *conformance claim*, showing whether the PP claims conformance to any PPs and/or packages, and if so, to which PPs and/or packages;
- a *security problem definition*, showing the threats, OSPs and assumptions that must be countered, enforced and upheld by the TOE and its operational environment;

- *security objectives*, showing how the solution to the security problem is divided between security objectives for the TOE and security objectives for the operational environment of the TOE;

- *extended components definition*, where new components (i.e. not included in CC Part 2 or CC Part 3) may be defined. These new components can then be used are needed to define extended functional and extended assurance requirements;

- *security requirements*, where a translation of the security objectives for the TOE into a standardised language is provided. This standardised language is in the form of SFRs. Additionally this section defines the SARs;

360 There also exist low assurance PPs, which have reduced contents, which are described in detail in Section B.11. The remainder of this Annex assumes that a PP with full contents is used.

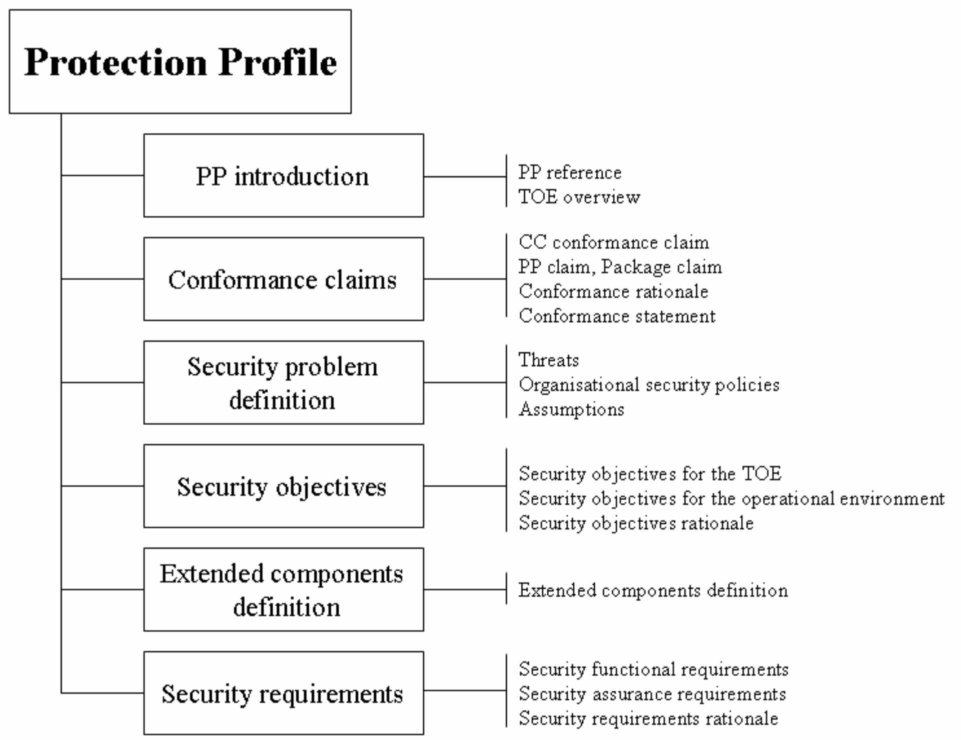
Protection Profile
PP introduction
PP reference
TOE overview
Conformance claims
CC conformance claim
PP claim, Package claim
Conformance rationale
Conformance statement
Security problem definition
Threats
Organisational security policies
Assumptions
Security objectives
Security objectives for the TOE
Security objectives for the operational environment
Security objectives rationale
Extended components definition
Extended components definition
Security requirements
Security functional requirements
Security assurance requirements
Security requirements rationale

Figure 9 - Protection Profile contents

B.3 Using the PP

B.3.1 How a PP should be used

361 A PP is typically a statement of need where a user community, a regulatory entity, or a group of developers define a common set of security needs. A PP gives consumers a means of referring to this set, and facilitates future evaluation against these needs.

362 A PP is therefore typically used as:

- part of a requirement specification for a specific consumer or group of consumers, who will only consider buying a specific type of IT if it meets the PP;
- part of a regulation from a specific regulatory entity, who will only allow a specific type of IT to be used if it meets the PP;
- a baseline defined by a group of IT developers, who then agree that all IT that they produce of this type will meet this baseline.

though this does not preclude other uses.

B.3.2 How a PP should not be used

363 Three roles (among many) that a PP should not fulfil are:

- *a detailed specification*: A PP is designed to be a security specification on a relatively high level of abstraction. A PP should, in general, not contain detailed protocol specifications, detailed descriptions of algorithms and/or mechanisms, long description of detailed operations etc.
- *a complete specification*: A PP is designed to be a security specification and not a general specification. Unless security-relevant, properties such as interoperability, physical size and weight, required voltage etc. should not be part of a PP. This means that in general a PP is a part of a complete specification, but not a complete specification itself.
- *a specification of a single product*: Unlike an ST, a PP is designed to describe a certain type of IT, and not a single product. When only a single product is described, it is better to use an ST for this purpose.

B.4 PP introduction (APE_INT)

364 A PP introduction describes the TOE on two levels of abstraction:

- the PP reference;
- the TOE overview.

B.4.1 PP reference

365 A PP contains a clear PP reference that identifies that particular PP. A typical PP reference consists of title, version, authors and publication date. An example of a PP reference is “Atlantean Navy CablePhone Encryptor PP, version 2b, Atlantean Navy Procurement Office, April 7, 2003”. The reference must be unique so that it is possible to tell different PPs and different versions of the same PP apart.

366 The PP reference facilitates indexing and referencing the PP and its inclusion in lists of PPs.

B.4.2 TOE overview

367 The TOE overview is aimed at potential consumers of a TOE who are looking through lists of evaluated products to find TOEs that may meet their security needs, and are supported by their hardware, software and firmware.

368 The TOE overview is also aimed at developers who may use the PP in designing TOEs or in adapting existing products.

369 The typical length of a TOE overview is several paragraphs.

370 To this end, the TOE overview briefly describes the usage of the TOE and its major security features, identifies the TOE type and identifies any major non-TOE hardware/software/firmware available to the TOE.

B.4.2.1 Usage and major security features of a TOE

371 The description of the usage and major security features of the TOE is intended to give a very general idea of what the TOE should be capable of, and what it can be used for. This section should be written for (potential) TOE consumers, describing TOE usage and major security features in terms of business operations, using language that TOE consumers understand.

372 An example of this is "The Atlantean Navy CablePhone Encryptor is an encryption device that should allow confidential communication between ships across the Atlantean Navy CablePhone system. To this end it should allow at least 32 different users and support at least 100Mb encryption speed. It should allow both bilateral communication between ships and broadcast across the entire network."

B.4.2.2 TOE Type

373 The TOE overview identifies the general type of TOE, such as: firewall, VPN-firewall, smart card, crypto-modem, intranet, web server, database, web server and database, LAN, LAN with web server and database, etc.

B.4.2.3 Available non-TOE hardware/software/firmware

374 While some TOEs do not rely upon other IT, many TOEs (notably software TOEs) rely on additional, non-TOE, hardware, software and/or firmware. In the latter case, the TOE overview is required to identify the non-TOE hardware/software/firmware.

375 As a Protection Profile is not written for a specific product, in many cases only a general idea can be given of the available hardware/software/firmware. In some other cases, e.g. a requirements specification for a specific consumer where the platform is already known, (much) more specific information may be provided.

376 Examples of hardware/software/firmware identifications are:

- None. (for a completely stand-alone TOE)
- The Yaiza 3.0 Operating System running on a general PC.
- a CleverCard SB2067 integrated circuit;
- a CleverCard SB2067 IC running v2.0 of the QuickOS smart card operating system;
- the December 2002 installation of the LAN of the Director-General's Office of the Department of Traffic.

B.5 Conformance claims (APE_CCL)

377 This section of a PP describes how the PP conforms with other PPs and with packages. It is identical to the conformance claims section for an ST (see Section A.5), with one exception: the conformance statement.

378 The conformance statement in the PP states how STs and/or other PPs must conform to that PP. The PP author selects whether “strict” or “demonstrable” conformance is required. See Annex D for more details on this.

379 The authors of other PP/STs that subsequently claim conformance to the PP must then conform to the PP according to the conformance statement in the PP.

B.6 Security problem definition (APE_SPD)

380 This section is identical to the security problem definition section of an ST as described in Section A.6.

B.7 Security objectives (APE_OBJ)

381 This section is identical to the security objectives section of an ST as described in Section A.7.

B.8 Extended components definition (APE_ECD)

382 This section is identical to the extended components section of an ST as described in Section A.8.

B.9 Security requirements (APE_REQ)

383 This section is identical to the security requirements section of an ST as described in Section A.9. Note however that the rules for completing operations in a PP are slightly different from the rules for completing operations in an ST. This is described in more detail in Section C.4.

B.10 TOE summary specification

384 A PP has no TOE summary specification.

B.11 Low assurance Protection Profiles

385 A low assurance PP has the same relationship to a regular PP, as a low assurance ST has to a regular ST. This means that a low-assurance PP consists of

- a PP introduction, consisting of a PP reference and a TOE overview;
- a conformance claim;
- security objectives for the operational environment;
- the SFRs and the SARs (including the extended components definition) and the security requirements rationale (only if the dependencies are not satisfied).

386 A low-assurance PP may only claim conformance to a low-assurance PP (see Annex B). A non-low assurance PP may claim conformance with a low assurance PP.

387 The reduced content of a low assurance PP is shown in Figure 10.

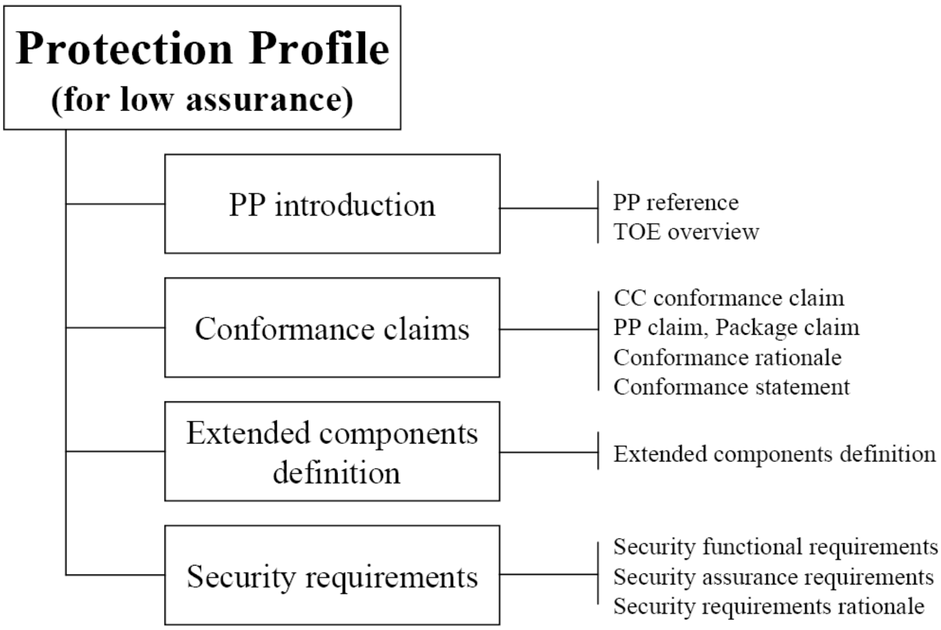

Figure 10 - Contents of a Low Assurance Protection Profile

B.12 Referring to other standards in a PP

388 This section is identical to the section on standards for STs as described in Section A.13, with one exception: as a PP has no TOE summary specification, the third option is not valid for PPs.

389 The PP author is reminded that referring to a standard in SFRs may impose a significant burden on a developer developing a TOE to meet that PP (depending on the size and complexity of the standard and the assurance level required), and that it may be more suitable to require alternative (non-CC related) ways to assess conformance to that standard.

C Security Requirements (normative)

C.1 Introduction

390 In the CC, packages, Protection Profiles and Security Targets contain security requirements. The CC has been developed around the central notion that these requirements are derived from:

- pre-defined security functional components that are listed in CC Part 2, and
- pre-defined security assurance components that are listed in CC Part 3.

391 These predefined components represent the preferred expression of security requirements as they are based on experience and represent a well-known and understood domain.

392 The components in CC Part 2 and CC Part 3 should be considered as pre-defined templates for SFRs and SARs, to be filled in and modified by operations in a PP, ST or package.

C.2 Organisation of components

393 The CC has organised the components in CC Part 2 and CC Part 3 into hierarchical structures:

- Classes, consisting of
- Families, consisting of
- Components, consisting of
- Elements.

394 This organisation into a hierarchy of class - family - component - element is provided to assist consumers, developers and evaluators in locating specific components.

395 The CC presents functional and assurance components in the same general hierarchical style and uses the same organisation and terminology for each.

C.2.1 Class

396 The term class is used for the most general grouping of security components. All the members of a class share a common general focus. An example of a class is the FIA class that is focused at identification of users, authentication

of users and binding of users and subjects. The members of a class are termed families.

C.2.2 Family

397 A family is a grouping of components that share a more specific focus but may differ in emphasis or rigour. An example of a family is the User authentication (FIA_UAU) family which is part of the FIA class. This family concentrates on the authentication of users. The members of a family are termed components.

C.2.3 Component

398 A component is the smallest selectable unit in the CC. The set of components within a family may be ordered to represent increasing strength or capability. They may also be partially ordered to represent related non-hierarchical sets. In some instances, there is only one component in a family so ordering is not applicable. An example of a component is FIA_UAU.3 Unforgeable authentication which concentrates on unforgable authentication.

C.2.4 Element

399 The components are constructed from individual elements. The element is the lowest level expression of a security need that is verified by the evaluation. An example of an element is FIA_UAU.3.2 which concentrates on the prevention of use of copied authentication data.

C.3 Dependencies between components

400 Dependencies may exist between components. Dependencies arise when a component is not self sufficient and relies upon the presence of another component to provide security functionality or assurance.

401 The functional components in CC Part 2 have only dependencies on other functional components and the assurance components in CC Part 3 have only dependencies on other assurance components. However, this does not preclude extended functional components having dependencies on assurance components or vice versa.

402 Component dependency descriptions are part of the CC component definitions. In order to ensure completeness of the TOE security requirements, dependencies should be satisfied when requirements based on components with dependencies are incorporated into PPs and STs. Dependencies should also be considered when constructing packages.

403 In other words: if component A has a dependency on component B, this means that whenever a PP/ST contains a security requirement based on component A, the PP/ST shall also contain one of :

- a security requirement based on component B, or

- a security requirement based on a component that is hierarchically higher than B, or

- a justification why the PP/ST does not contain a security requirement based on component B.

404 In cases a) and b), when a security requirement is included because of a dependency, it may be necessary to complete operations (assignment, iteration, refinement, selection) on that security requirement in a particular manner to make sure that it actually satisfies the dependency.

405 In case c), the justification that a security requirement is not included should address either:

- why the dependency is not necessary or useful, or

- that the dependency has been addressed by the operational environment of the TOE, in which case the justification should describe how the security objectives for the operational environment address this dependency, or

- that the dependency has been addressed by the other SFRs in some other manner (extended SFRs, combinations of SFRs etc.)

C.4 Operations

406 CC functional and assurance components may be used exactly as defined in the CC, or they may be tailored through the use of permitted operations. When using operations, the PP/ST author should be careful that the dependency needs of other requirements that depend on this requirement are satisfied. The permitted operations are selected from the following set:

- Iteration: allows a component to be used more than once with varying operations;

- Assignment: allows the specification of parameters;

- Selection: allows the specification of one or more items from a list; and

- Refinement: allows the addition of details.

407 The assignment and selection operations are permitted only where specifically indicated in a component. Iteration and refinement are permitted for all components. The operations are described in more detail below.

C.4.1 The iteration operation

408 The iteration operation may be performed on every component. The PP/ST author performs an iteration operation by including multiple requirements based on the same component. Each iteration of a component shall be

different from all other iterations of that component, which is realised by completing assignments and selections in a different way, or by applying refinements to it in a different way. An example of an iteration is FCS_COP.1 being iterated twice in order to require the implementation of two different cryptographic algorithms.

409 Different iterations should be uniquely identified to allow clear rationales and tracings to and from these requirements.

C.4.2 The assignment operation

410 An assignment operation occurs where a given component contains an element with a parameter that may be set by the PP/ST author. The parameter may be an unrestricted variable, or a rule that narrows the variable to a specific range of values. An example of an element with an assignment is: FIA_AFL.1.2 “When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **[assignment: list of actions]**.”

411 Whenever an element in a PP contains an assignment, a PP author shall do one of four things:

- leave the assignment uncompleted. The PP author could include FIA_AFL.1.2 “When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **[assignment: list of actions]**.” in the PP.

- complete the assignment. As an example, the PP author could include FIA_AFL.1.2 “When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **prevent that external entity from binding to any subject in the future**.” in the PP.

- narrow the assignment, to further limit the range of values that is allowed. As an example, the PP author could include FIA_AFL.1.1 “The TSF shall detect when [assignment: positive integer between 4 and 9] unsuccessful authentication attempts occur ...” in the PP.

- transform the assignment to a selection, thereby narrowing the assignment. As an example, the PP author could include FIA_AFL.1.2 “When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall **[selection: prevent that user from binding to any subject in the future, notify the administrator]**.” in the PP.

412 Whenever an element in an ST contains an assignment, an ST author shall complete that assignment, as indicated in b) above. Options a), c) and d) are not allowed for STs.

413 The values chosen in options b), c) and d) shall conform to the indicated type required by the assignment.

414 When an assignment is to be completed with a set (e.g. subjects), one may list a set of subjects, but also some description of the set from which the elements of the set can be derived such as:

- all subjects
- all subjects of type X
- all subjects except subject a

as long as it is clear which subjects are meant.

C.4.3 The selection operation

415 The selection operation occurs where a given component contains an element where a choice from several items has to be made by the PP/ST author. An example of an element with a selection is: FPT_TST.1.1 “The TSF shall run a suite of self tests [selection: during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self test should occur]] to demonstrate the correct operation of ...”

416 Whenever an element in a PP contains a selection, the PP author may do one of three things:

- leave the selection uncompleted. As an example, the PP author could include FPT_TST.1 “The TSF shall run a suite of self tests **[selection: during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self test should occur]]** to...” in the PP.
- complete the selection by choosing one or more items. As an example, the PP author could include FPT_TST.1 “The TSF shall run a suite of self tests **during initial start-up and periodically during normal operation** to...” in the PP.
- restrict the selection by removing some of the choices, but leaving two or more. As an example, the PP author could include FPT_TST.1 “The TSF shall run a suite of self tests **[selection: during initial start-up, periodically during normal operation]** to...” in the PP.

417 Whenever an element in an ST contains a selection, an ST author shall complete that selection, as indicated in b) above. Options a) and c) are not allowed for STs.

418 The item or items chosen in b) and c) shall be taken from the items provided in the selection.

C.4.4 The refinement operation

419 The refinement operation can be performed on every requirement. The PP/ST author performs a refinement by altering that requirement. The first rule for a refinement is that a TOE meeting the refined requirement also meets the unrefined requirement in the context of the PP/ST (i.e. a refined requirement must be "stricter" than the original requirement). If a refinement does not meet this rule, the resulting refined requirement is considered to be an extended requirement and shall be treated as such.

420 An example of a valid refinement is FIA_UAU.2.1 "The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user." being refined to "The TSF shall require each user to be successfully authenticated **by username/password** before allowing any other TSF-mediated actions on behalf of that user."

421 The only exception to this rule is that a PP/ST author is allowed to refine a SFR to apply to some but not all subjects, objects, operations, security attributes and/or external entities.

422 An example of a such an exception is FIA_UAU.2.1 "The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user." being refined to "The TSF shall require each user **originating from the internet** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user."

423 However, this exception does not apply to refining SFRs that are taken from PPs that compliance is being claimed to; these SFRs may not be refined to apply to fewer subjects, objects, operations, security attributes and/or external entities than the SFR in the PP.

424 The second rule for a refinement is that the refinement shall be related to the original component. For example, refining an audit component with an extra element on prevention of electromagnetic radiation is not allowed.

425 A special case of refinement is an editorial refinement, where a small change is made in a requirement, i.e. rephrasing a sentence due to adherence to proper English grammar, or to make it more understandable to the reader. This change is not allowed to modify the meaning of the requirement in any way. Examples of editorial refinements include:

- the SFR FPT_FLS.1 "The TSF shall continue to preserve a secure state when the following failures occur: **breakdown of one CPU**" could be refined to FPT_FLS.1 "The TSF shall continue to preserve a secure state when the following failure occurs: **breakdown of one CPU**" or even FPT_FLS.1 "The TSF shall continue to preserve a secure state when **one CPU breaks down**"

C.5 Extended components

426 In the CC it is mandatory to base requirements on components from CC Part 2 or CC Part 3 with two exceptions:

- there are security objectives for the TOE that can not be translated to Part 2 SFRs, or there are security objectives for the development environment that can not be translated to Part 3 SARs (e.g. strength of cryptographic algorithms);
- a security objective can be translated, but only with great difficulty and/or complexity based on components in CC Part 2 and/or CC Part 3.

427 In both cases the PP/ST author is required to define his own components. These newly defined components are called extended components. A precisely defined extended component is needed to provide context and meaning to the extended SFRs and SARs based on that component.

428 After the new components have been defined correctly, the PP/ST author can then base one or more SFRs or SARs on these newly defined extended components and use them in the same way as the other SFRs and SARs. From this point on, there is no further distinction between SARs and SFRs based on the CC and SARs and SFRs based on extended components.

C.5.1 How to define extended components

429 Whenever a PP/ST author defines an extended component, this has to be done in a similar manner to the existing CC components: clear, unambiguous and evaluatable (it is possible to systematically demonstrate whether a requirement based on that component holds for a TOE). Extended components must use similar labelling, manner of expression, and level of detail as the existing CC components.

430 The PP/ST author also has to make to sure that all applicable dependencies of an extended component are included in the definition of that extended component. Examples of possible dependencies are:

- if an extended component refers to auditing, dependencies to components of the FAU class may have to be included;
- if an extended component modifies or accesses data, dependencies to components of the FDP_ACC family may have to be included;
- if an extended component uses a particular design description a dependency to the appropriate ADV family (e.g. Functional Specification) may have to be included.

431 In the case of an extended functional component, the PP/ST author also has to include any applicable audit and associated operations information in the definition of that component, similar to existing CC Part 2 components. In

the case of an extended assurance component, the PP/ST author also has to provide suitable methodology for the component, similar to the methodology provided in the CEM.

432 Extended components may be placed in existing families, in which case the PP/ST writer has to show how these families change. If they do not fit into an existing family, they shall be placed in a new family. New families have to be defined similarly to the CC.

433 New families may be placed in existing classes in which case the PP/ST writer has to show how these classes change. If they do not fit into an existing class, they shall be placed in a new class. New classes have to be defined similarly to the CC.

D PP conformance (normative)

D.1 Introduction

434 A PP is intended to be used as a “template" for an ST. That is: the PP describes a set of user needs, while an ST that conforms to that PP describes a TOE that satisfies those needs.

435 Note that it is also possible for a PP to be used as a template for another PP. This case is completely similar to that of an ST vs. a PP. For clarity this Annex describes only the ST/PP case, but it holds also for the PP case.

436 This Annex describes what it means for an ST to conform to a PP. The CC recognises two types of conformance:

- *strict conformance*: there exists a very strict relation between the PP and the ST. This relation can be roughly defined as “the ST shall contain all statements that are in the PP, but may contain more”. Strict conformance is expected to be used for stringent requirements that are to be adhered to in a single manner;

- *demonstrable conformance*: there is no subset-superset type relation between the PP and the ST. The PP and the ST may contain entirely different statements that discuss different entities, use different concepts etc. However, the ST shall contain a rationale on why the ST is considered to be “equivalent or more restrictive” than the PP (see Section D.3). Demonstrable conformance allows a PP author to describe a common security problem to be solved and provide generic guidelines to the requirements necessary for its resolution, in the knowledge that there is likely to be more than one way of specifying a resolution. Demonstrable conformance is also suitable for a TOE type where several similar PPs already exist (or likely to exist in the future), thus allowing the ST author to claim conformance to all these PPs simultaneously, thereby saving work.

437 The allowed type of conformance is determined by the PP. That is, the PP states (in the PP conformance statement, see Section B.5) what the allowed types of conformance for the ST are:

- if the PP states that strict conformance is required, the ST shall conform to the PP in a strict manner;

- if the PP states that demonstrable conformance is required, the ST shall conform to the PP in a strict or demonstrable manner.

438 Restating this in other words, an ST is only allowed to conform in a PP in a demonstrable manner, if the PP explicitly allows this.

439 If an ST claims conformance to multiple PPs, it shall conform (as described above) to each PP in the manner ordained by that PP. This may mean that the ST conforms strictly to some PPs and demonstrably to other PPs.

440 Note that either the ST conforms to the PP in question or it does not. The CC does not recognise “partial" conformance. It is therefore the responsibility of the PP author to ensure the PP is not overly onerous, prohibiting PP/ST authors in claiming conformance to the PP.

D.2 Strict conformance

441 Strict conformance is oriented to the PP-author who requires evidence that the requirements in the PP are met, that the ST is an instantiation of the PP, though the ST could be broader than the PP. In essence, the ST specifies that the TOE does at least the same as in the PP, while the operational environment does at most the same as in the PP. In more detail:

- **Security problem definition**: The ST shall contain the security problem definition of the PP, may specify additional threats and OSPs, but may not specify additional assumptions.
- **Security objectives**: The ST:
    - shall contain all security objectives for the TOE of the PP but may specify additional security objectives for the TOE;
    - shall contain all security objectives for the operational environment (with one exception in the next bullet) but may not specify additional security objectives for the operational environment;
    - may specify that certain objectives for the operational environment in the PP are security objectives for the TOE in the ST. This is called *re-assigning* a security objective.
- **Security requirements**: The ST shall contain all SFRs and SARs in the PP, but may claim additional or hierarchically stronger SFRs and SARs. The completion of operations in the ST must be consistent with that in the PP; either the same completion will be used in the ST as that in the PP or one that makes the requirement more restrictive (the rules of refinement apply).

442 Note that in some cases a PP author may not wish that some or all objectives for the operational environment are re-assigned as objectives of the TOE. If this is the case, this should be stated in the PP.

443 Also note that it is allowed to restate threats, OSPs, assumptions and security objectives using a terminology that may be more familiar to ST consumers for that particular ST (e.g. an ST for a medical system may use terms like “doctors”, “medical assistants”, “hospital administrator”) even though it claims conformance to a more general PP, that uses terminology like “senior

staff”, “junior staff” and “administrative staff”). In this case, the conformance rationale in the PP shall demonstrate the equivalence of the different terminologies.

D.3 Demonstrable conformance

444 Demonstrable conformance is orientated to the PP-author who requires evidence that the ST is a suitable solution to the generic security problem described in the PP. Where there is a clear subset-superset type relation between PP and ST in the case of strict conformance, the relation is less clear-cut in the case of demonstrable conformance. The general statement is that the ST must be equivalent or more restrictive than the PP. An ST is equivalent or more restrictive than a PP if:

- all TOEs that meet the PP also meet ST, and
- all operational environments that meet the ST also meet the PP.

or, informally, the ST shall levy the same or more, restrictions on the TOE and the same or less restrictions on the operational environment of the TOE.

445 This general statement can be made more specific for various sections of the ST:

- **Security problem definition**: The conformance rationale in the ST shall demonstrate that the security problem definition in the ST is equivalent (or more restrictive) than the security problem definition in the PP. This means that:
    - all TOEs that would meet the security problem definition in the ST also meet the security problem definition in the PP;
    - all operational environments that would meet the security problem definition in the PP would also meet the security problem definition in the ST.
- **Security objectives**: The conformance rationale in the ST shall demonstrate that the security objectives in the ST is equivalent (or more restrictive) than the security objectives in the PP. This means that:
    - all TOEs that would meet the security objectives for the TOE in the ST also meet the security objectives for the TOE in the PP;
    - all operational environments that would meet the security objectives for the operational environment in the PP would also meet the security objectives for the operational environment in the ST.

- **SFRs**: The conformance rationale in the ST shall demonstrate that the SFRs in the ST are equivalent (or more restrictive) than the SFRs in the PP. This means that all TOEs that would meet the SFRs in the ST would also meet the SFRs in the PP;

- **SARs**: The ST shall contain all SARs in the PP, but may claim additional or hierarchically stronger SARs. The completion of operations in the ST must be consistent with that in the PP; either the same completion will be used in the ST as that in the PP or a completion that makes the SAR more restrictive (the rules of refinement apply).