

NIST Special Publication 800-xx

# Recommendation for Common Criteria Assurance Packages

**NIST**

**National Institute of  
Standards and Technology**

Technology Administration  
U.S. Department of Commerce

Gary Stoneburner

INFORMATION SECURITY

DRAFT

Computer Security Division  
Information Technology Laboratory  
National Institute of Standards and Technology  
Gaithersburg, MD 20899-8930

VERSION 0.2  
Initial Public Draft

30 January 2004



**U.S. Department of Commerce**

*Donald L. Evans, Secretary*

**Technology Administration**

*Phillip J. Bond, Under Secretary of Commerce for Technology*

**National Institute of Standards and Technology**

*Arden L. Bement, Jr., Director*

## Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analyses to advance the development and productive use of information technology. ITL's responsibilities include the development of management, administrative, technical, and physical standards and guidelines for the cost-effective security and privacy of non-national security-related information in federal information systems. This special publication 800-series reports on ITL's research, guidelines, and outreach efforts in information system security, and its collaborative activities with industry, government, and academic organizations.

DRAFT

**U.S. GOVERNMENT PRINTING OFFICE  
WASHINGTON: 2003**

---

For sale by the Superintendent of Documents, U.S. Government Printing Office  
Internet: [bookstore.gpo.gov](http://bookstore.gpo.gov) — Phone: (202) 512-1800 — Fax: (202) 512-2250  
Mail: Stop SSOP, Washington, DC 20402-0001

## Authority

This document has been developed by the National Institute of Standards and Technology (NIST) in furtherance of its statutory responsibilities under the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

NIST is responsible for developing standards and guidelines, including minimum requirements, for providing adequate information security for all agency operations and assets, but such standards and guidelines shall not apply to national security systems. This guideline is consistent with the requirements of the Office of Management and Budget (OMB) Circular A-130, Section 8b(3), Securing Agency Information Systems, as analyzed in A-130, Appendix IV: Analysis of Key Sections. Supplemental information is provided A-130, Appendix III.

This guideline has been prepared for use by federal agencies. It may be used by nongovernmental organizations on a voluntary basis and is not subject to copyright. (Attribution would be appreciated by NIST.)

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding on federal agencies by the Secretary of Commerce under statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director of the OMB, or any other federal official.

**National Institute of Standards and Technology Special Publication 800-xx, yy pages**  
**(date) CODEN: NSPUE2**

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

THE PUBLIC COMMENT PERIOD FOR THIS DOCUMENT BEGINS ON *DATE*  
AND ENDS ON *DATE*. COMMENTS MAY BE SUBMITTED TO THE COMPUTER  
SECURITY DIVISION, NIST, VIA ELECTRONIC MAIL AT *EMAIL ADDRESS*

OR VIA REGULAR MAIL AT

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY  
ATTN: SP 800-XX COMMENTS  
100 BUREAU DRIVE (MAIL STOP 8930)  
GAITHERSBURG, MD 20899-8930

## Acknowledgements

TDB

**DRAFT**

## Note to Reviewers

This first public draft is presented as a means of engaging interested parties in a dialog leading to enhanced cost-effectiveness for IT assurance measures. It is expected that substantial changes may occur in this document prior to eventual release.

Although currently formatted as a NIST Special Publication, this document may be released in another the form instead; for example, a NIST Interagency Report (NISTIR). Comments concerning the form of publication (SP, NISTIR, other) are solicited in addition to comments about the document contents.

**DRAFT**

## Table of Contents

1.	INTRODUCTION.....	1
1.1	PURPOSE AND APPLICABILITY .....	1
1.2	ORGANIZATION OF THIS SPECIAL PUBLICATION .....	2
2.	ASSURANCE FUNDAMENTALS .....	3
2.1	WHY ASSURANCE IS IMPORTANT .....	3
2.2	WHAT IS ASSURANCE? .....	3
2.3	MEETS ITS SECURITY OBJECTIVES .....	5
2.4	EFFECTIVE ASSURANCES.....	6
3.	ASSURANCE PACKAGES - DESCRIPTIONS.....	8
3.1	AL1 - VULNERABILITY TESTING .....	8
3.2	AL2 - FUNCTIONAL TESTING.....	9
3.3	AL3 - RIGOROUS COTS DEVELOPMENT.....	10
3.4	AL4 – SIGNIFICANT SECURITY ENGINEERING.....	11
3.5	AL5 – VERIFIED SIGNIFICANT SECURITY ENGINEERING.....	13
3.6	AL6 – RIGOROUS SECURITY ENGINEERING.....	15
3.7	AL7 – VERIFIED, RIGOROUS SECURITY ENGINEERING.....	17
3.8	AL8 – FORMAL METHODS.....	19
3.9	AL9 – VERIFIED FORMAL METHODS.....	20
4.	ASSURANCE PACKAGES - CONTENTS .....	22
4.1	AL1 - VULNERABILITY TESTING .....	24
4.2	AL2 - FUNCTIONAL TESTING.....	28
4.3	AL3 - RIGOROUS COTS DEVELOPMENT.....	34
4.4	AL4 – SIGNIFICANT SECURITY ENGINEERING.....	51
4.5	AL5 – VERIFIED SECURITY ENGINEERING.....	74
4.6	AL6 – RIGOROUS SECURITY ENGINEERING.....	96
4.7	AL7 – VERIFIED, RIGOROUS SECURITY ENGINEERING.....	122
4.8	AL8 – FORMAL METHODS.....	149
4.9	AL9 – VERIFIED FORMAL METHODS .....	175
A.	CATALOG OF ASSURANCE COMPONENTS.....	202
A.1	CONFIGURATION MANAGEMENT .....	202
A.2	DELIVERY AND OPERATION .....	222
A.3	DEVELOPMENT .....	228
A.4	GUIDANCE DOCUMENTATION.....	259
A.5	LIFE CYCLE SUPPORT.....	262
A.6	TESTS.....	278
A.7	VULNERABILITY ASSESSMENT .....	297
B.	REFERENCES.....	318
C.	GLOSSARY.....	319
D.	ACRONYMS .....	325

**List of Figures**

Figure 2-1 Assurance Model..... 7

**List of Tables**

Table 4-1 Assurance Level (AL) Contents in terms of similar CC components ..... 23

DRAFT

## CHAPTER ONE

## 1

**1. INTRODUCTION**

## THE NEED FOR ADDITIONAL ASSURANCE PACKAGES

The EALs (Evaluation Assurance Levels) in the Common Criteria, are expressions of the paradigm that assurance is evaluation and more assurance is achieved by more evaluation. This paradigm is at odds with the commercial off the shelf (COTS) marketplace, reflecting neither how confidence is typically obtained nor providing a cost-effective means for supplying grounds for confidence in the security capabilities of the information technology being evaluated.

The CC EALs, by their very nature, place requirements upon the developer for documentation to support evaluation and place requirements for evaluator effort. These requirements increase the cost to the developer and without justification that the end consumer is obtaining an increase in security quality commensurate with the cost of evaluation.

An effective COTS strategy includes providing a baseline level of participation and a migration paths that are less costly than that codified in the current CC EALs.

The significant achievement of the Common Criteria is its wide, international acceptance. In like manner, the first major step forward in COTS security will be a large number of COTS vendors seeking AL (Assurance Level) compliance, whatever the AL.

**1.1 PURPOSE AND APPLICABILITY**

This purpose of this publication is to provide a more cost-effective set of assurance packages than the current CC evaluation assurance levels (EALs). Additionally, the intent is to better express the explicit developer actions that are necessary to achieve significant increases in the security quality of information technology.

The recommendations provided in Special Publication 800-xx are applicable to all Federal information systems<sup>1</sup> other than those systems designated as national security systems as defined in 44 U.S.C., Section 3542.<sup>2</sup> These recommendations have been broadly developed

---

<sup>1</sup> A Federal information system is an information system used or operated by an executive agency, by a contractor of an executive agency, or by another organization on behalf of an executive agency.

<sup>2</sup> A national security system is any information system (including any telecommunications system) used or operated by an agency or by a contractor of an agency, or other organization on behalf of an agency— (i) the function, operation, or use of which: involves intelligence activities; involves cryptologic activities related to national security; involves command and control of military forces; involves equipment that is an integral part of a weapon or weapons system; or is critical to the direct fulfillment of military or intelligence missions (excluding a system that is to be used for routine administrative and business applications, for example, payroll, finance, logistics, and personnel management applications); or, (ii) is protected at all times by procedures established for information that have been specifically authorized under criteria established by an Executive Order or an Act of Congress to be kept classified in the interest of national defense or foreign policy. Agencies should consult NIST Special Publication 800-59, *Guide for Identifying an Information System as a National Security System*, for guidance on determining the status of their information systems.



from a technical perspective to complement similar guidelines issued by agencies and offices operating or exercising control over national security systems. State, local, and tribal governments as well as private sector organizations are encouraged to consider the use of these recommendations as appropriate.

## **1.2 ORGANIZATION OF THIS SPECIAL PUBLICATION**

This document consists of the following sections:

- Chapter 1, this introduction
- Chapter 2 provides a background on assurance fundamentals
- Chapter 3 describes the 9 recommended assurance packages
- Chapter 4 gives the specific contents of each package
- Appendix A - Catalog of assurance components used (presented as additions and deletions to existing common criteria components)
- Appendix B - References
- Appendix C - Glossary
- Appendix D - Acronyms

## CHAPTER TWO

## 2

## 2. ASSURANCE FUNDAMENTALS

### HOW ASSURANCE IS OBTAINED IN THE REAL WORLD

*Assurance - “Grounds for confidence that an entity meets its security objectives” (CC, ISO 15408)*

Despite an internationally recognized definition for “assurance”, that term has in practice multiple, often conflicting meanings. Assurance is used to describe the degree to which confidence is held, to describe the amount of information available upon which to base confidence, and to describe characteristics of the information technology that exist where or not confidence in their existence is present.

### 2.1 WHY ASSURANCE IS IMPORTANT

Over the last 20 years the nature of IT and its relationship to our business or mission processes has changed dramatically. We have wholeheartedly embraced this change, seeing the great potential and rewards. However, like most innovation there is a downside as well. Because this downside is not as obvious and has not been captured in a way that most IT users can assess, it has been frequently, in practice, ignored.

IT Twenty Years Ago. Twenty years ago IT was largely stand-alone systems that stood outside our business/mission processes in a supporting role. The number of IT competent individuals was small and the systems were generally housed in physically protected environments.

IT Today. Today our IT is interconnected and integral to our business/mission processes. The number of IT proficient individuals is very large and the systems, by and large, no longer have the degree of physical protection that was common place 20 years ago.

IT that is interconnected and integral is radically different from IT that is stand-alone and supporting. Add to this the explosion in IT proficiency, and the potential for negative impacts to our organizations becomes tremendous! IT-based processes have become the norm even though the IT itself is still being produced largely with a level of quality appropriate for its use as stand-alone and supporting rather than interconnected and integral. The functionality is there, but the general level of quality has not risen to match the change in our use of IT.

### 2.2 WHAT IS ASSURANCE?

IT security has developed its own terminology and much of the underlying science held within a fairly closed group of evaluators and trusted product vendors. Only recently has IT security developed as an academic discipline. The word “assurance”, while a common English word, has unique meanings in the context of computing security. The on-line Merriam-Webster dictionary says:

Assurance - 1 : the act or action of assuring ... 2 : the state of being assured ... 3 : something that inspires or tends to inspire confidence ... [Merriam-Webster]

According to these definitions, assurance can be both something done to inspire confidence and the state of being confident. The IT security community uses “assurance” in both of these ways and even more:

- a. The confidence that the IT system is effective in meeting its security objectives. In this use assurance is a measure of how sure one is that the IT system will do what it is supposed to do and not do what it is not supposed to do. Note that “confidence” is a highly subjective quality that has a lot to do with “feelings” and emotions.
- b. The above plus confidence that the objectives themselves are correct. While this is the same type of use as “a.” above, the validity of the security objectives is added. The point is that even when using assurance as “confidence” the extent of what one is confident about varies, but the same word, “assurance”, is used.
- c. A specific type of measure that provides a basis for having confidence. This is distinctly different from the subjective nature of “a.” and “b.” above. Here the use is as an objective measurement related to the IT system. This is not a measure of confidence, nor is it a guarantee of a certain level of confidence, since that is a matter of human judgment, varying from individual to individual.

When used in this manner, assurance relates to a specific type of measurement. While the individual probably understands that other measures are possible, in practical terms “assurance” frequently becomes narrowly defined.

- d. Collection of measures of or facts about the system that provide a basis for having confidence. This is very similar to “c.” above, but includes multiple types of measurement and fact in the practical, working definition.
- e. The inherent security “quality” of the IT. The term “assurance” is used, not only as confidence or a metric, but as a statement of an IT characteristic. For example the phrase “high-assurance system” commonly means that the IT is of high security quality. This is distinctly different from how one feels about the system (confidence) and from measurements of the system. Here assurance is used to describe what is being measured.

These definitions show that assurance is being used to mean:

- a. A measure of human subjectivity; i.e., confidence.
- b. An objective measurement of or fact about the IT system.
- c. An IT characteristic that exists independent of confidence in the system or any measurement of or fact about the system.

The definition used in this Special Publication is that given in the Common Criteria, which is paraphrased as:

Assurance - The grounds for confidence that the IT meets explicitly identified security expectations.

This assurance can be used in making the determination that the negative impacts on an organization's operations (including mission, functions, image, or reputation), organizational assets, or individuals resulting from the operation of this IT are less than the positive benefits the IT provides.

Assurance is an objective measurement of or fact about your IT system upon which you can base your confidence. The definition above also captures the basic intent of IT risk management, focusing not on the IT but on how IT impacts you and your organization.

Assurance is therefore primarily concerning what is known about the system. Yet there is another important aspect that, as indicated above, is frequently tied up within the usage of the term "assurance".

Security Quality - The inherent system characteristic that is the object of assurance and in which users place their confidence.

Security quality is determined by how the system was specified, designed, and implemented. Moreover, security quality, like quality in general, is a system "ility" that is not changed by measurements taken, although actions taken because of the measurement may impact the security quality.

Confidence is subjective while security quality is a system characteristic that exists independent of the level of confidence users hold in the system. Users may have high confidence despite low quality or low confidence even though the quality is high.

Within this publication the goal for assurance packages is to produce given levels of security quality at a range of levels of verification that this amount of quality has been achieved.

## 2.3 MEETS ITS SECURITY OBJECTIVES

For information technology to "meet its security objectives" can be stated as the IT-related [business, mission] risks are either:

- a. small enough or
- b. less than the IT-related benefits

In either case (a or b above), the fundamental point is mitigating risks to the organization resulting from the use of the IT. This is an important distinction from the commonly applied definition, which addresses assurance with respect to protecting the IT itself.

Effective security requirement sets will contain, or point to, the following:

- a. A non-technical, clear, and concise description of the nature of the operational environment (in business or mission terms) and the degree of protection this requirement set addresses.
- b. A set of security requirements to meet the above.
- c. Compelling rationale for the claim that the given requirements will meet the needs.

## 2.4 EFFECTIVE ASSURANCES

Useful criteria is effective criteria that has been embraced by the purchase decision-makers as their statement of need. Criteria is NOT useful if it is (1) lacking in effectiveness, (2) has not been embraced by the users, or (3) perhaps has been embraced, but not by those who decide what to buy. This is true whether the criteria concerns functions to be performed or “assurances” that the security objectives will be satisfied.

To be effective, assurance is evidence that:

- Is useful in human decisions related to confidence (must inspire confidence)
- Can be tied back to security quality via a metric (must objectively inspire confidence)
- Is cost-effective for the amount of confidence obtained

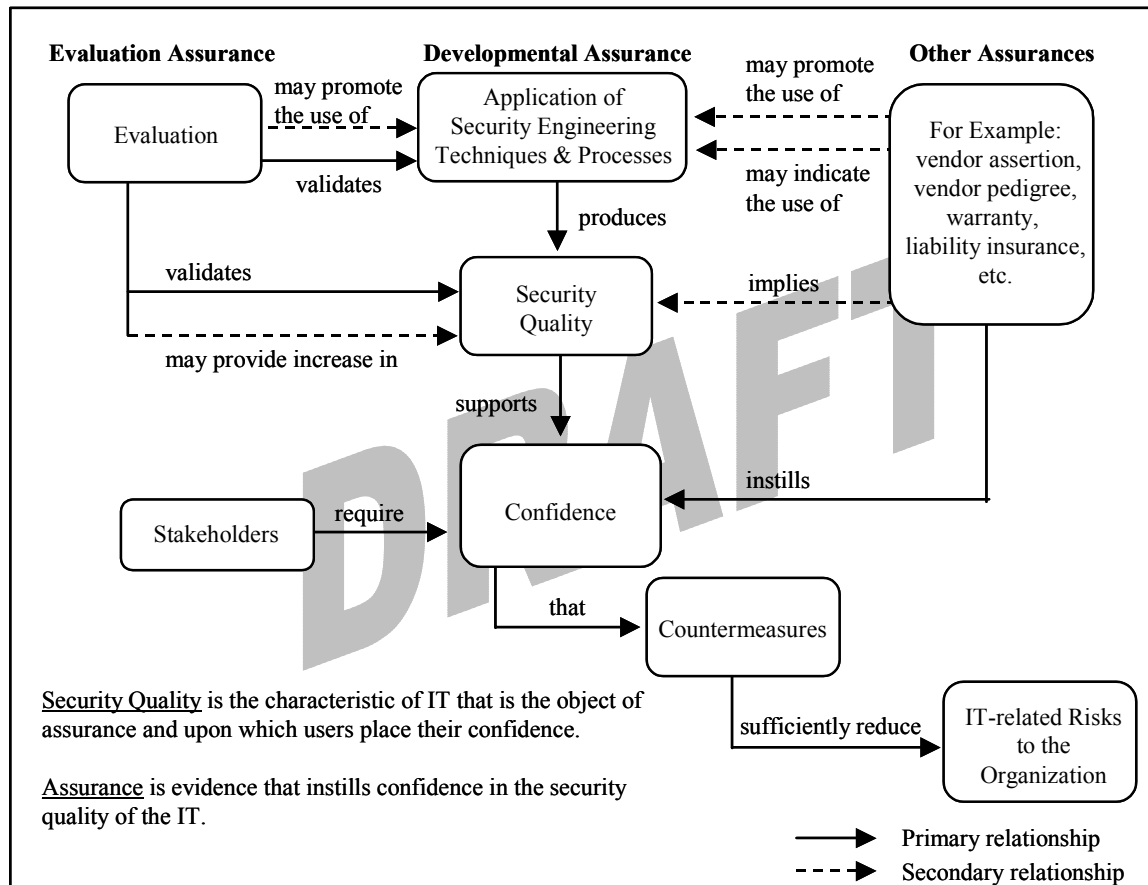
Inspire confidence. Assurance is evidence that convinces. If it does not serve to inspire confidence in the IT system, then the evidence is not useful. A metric is needed to weigh the relative usefulness of evidence, while keeping in clear focus the subjective nature of confidence. This usefulness metric is therefore inherently “soft” and is a measure of whether the evidence is suitable for inspiring confidence, not a measure of how much confidence it will inspire. Development of a metric remains a research issue. This document provides an ordering of assurance levels into a hierarchy and does not address the issue of how the results of assurance activities are captured to be most beneficial for the users.

Objective. The intent is to provide objective, scientifically based evidence as a foundation for what is inherently a subjective decision (i.e., confidence). It is therefore important to draw a distinction between the amount of confidence the assurance provides and the amount of “grounds for confidence” provided. The former is a subjective, human decision that can vary widely from individual to individual. The latter is objective information that can be measured and does not change from individual to individual. A metric is needed to measure the relative amount of objective information the evidence supplies about the underlying security quality of the IT. Again, the development of such a metric remains a research issue. For this document, the assurance levels are placed in a hierarchy of amounts of information as well as amount of security quality to be expected. A metric and precise measurement is not yet available.

Cost-effective. If the relative usefulness and the degree of objective information are known, then cost to obtain can be compared between different amounts of a given type of evidence, between different types of evidence, and between different sets of evidence to obtain the most cost-effective assurance. This remains a concept rather than a precise science with respect to well-defined metrics for which measures can be obtained and cost/benefit deter-

mined. For this document, cost-effectiveness is only generally achieved by providing for limited evaluation as an option and by generally adding evaluation effort only with added development effort.

The concepts from this chapter are captured in Figure 2-1 which depicts the relationship between different types of assurance, security quality, and managing risk to the organization.



**Figure 2-1 Assurance Model**

## CHAPTER THREE

## 3

### 3. ASSURANCE PACKAGES - DESCRIPTIONS

RECOMMENDED ASSURANCE PACKAGES EXPRESSED AS NINE (9) ASSURANCE LEVELS (ALs)

**N**ine assurance levels (ALs) are recommended:

- AL1 – Vulnerability testing
- AL2 – Functional testing
- AL3 – Rigorous COTS development
- AL4 – Significant security engineering
- AL5 – Verified, significant security engineering
- AL6 – Rigorous security engineering
- AL7 – Verified, rigorous security engineering
- AL8 – Formal methods
- AL9 – Verified formal methods

#### 3.1 AL1 - VULNERABILITY TESTING

Overview. AL1 provides a basic level of assurance primarily through third-party testing for obvious flaws and common vulnerabilities. Analysis of the TOE is conducted only to the extent necessary to support this testing. AL1 compliant components are not required to be of high security quality and hence, cost-effectiveness dictates minimal added cost through evaluator actions.

Description:

**AL1 provides assurance through these developer actions: configuration management.**

**AL1 provides evaluation assurance through the evaluator action of verifying, via penetration testing, that known attacks (e.g., those in the public domain) and obvious penetration means are not effective.**

## 3.2 AL2 - FUNCTIONAL TESTING

Overview. The purpose of AL2 is to increase assurance over AL1 without requiring a significant increase in cost and time. To accomplish this AL2 adds evaluator verification of functional compliance, allowing review of developer testing to be the means of accomplishing this action if developer testing is adequate. AL2 compliant components still are not required to be of high security quality and hence, cost-effectiveness continues to dictate minimal added cost through evaluator actions.

Description:

**AL2** provides assurance through these developer actions: configuration management, **and the development of a functional specification.**

**AL2** provides evaluation assurance through the evaluator actions of verifying, via penetration testing, that known attacks (e.g., those in the public domain) and obvious penetration means are not effective, **and ensuring that all functional requirements have been met (via review of developer testing if performed and adequate, else performing the testing necessary to come to this conclusion).**

**AL2** represents a meaningful increase in assurance from AL1 by requiring independent confirmation that functional requirements have been met.



### 3.3 AL3 - RIGOROUS COTS DEVELOPMENT

Overview. AL3 provides for greater assurance primarily through developer actions – some positive security engineering and the use of sound development practices. The required depth and rigor of these actions are intended to be achievable without changes to existing, good commercial practices. Because significant security engineering (and hence higher security quality) is not yet required, the evaluation activities remain minimal.

AL3 provides assurance through the requirement for developer analysis of the security functions using a security policy model, a functional and interface specification, and a high-level design of the subsystems of the TOE, helping to ensure a better understanding of the security behavior.

NIST Interagency Report 6462, *CSPP - Guidance for COTS Security Protection Profiles* [4], was used as the basis for the selection of assurance components for AL3. The resulting set of assurances is similar to Common Criteria EAL3 with the following differences:

<u>AL3 – Similar CC component</u>	<u>EAL3</u>
ACM_SCP.2	ACM_SCP.1
ADV_HLD.1	ADV_HLD.2
ADV_SPM.1	none
ALC_FLC.2	none
ALC_LCD.1	none
ALC_TAT.1	none
AVA_MSU.2	AVA_MSU.1
AVA_VLA.2	AVA_VLA.1

#### Description:

**AL3** provides assurance through these developer actions: **“gray” box testing, more configuration management, the development of a functional specification, development of a security policy model, development of a high-level design, verifying correspondence between design representations, life-cycle support, secure delivery procedures, and misuse analysis looking for potential errors in documentation and operating procedures.**

**AL3** provides evaluation assurance through the evaluator actions of verifying, via penetration testing, that known attacks (e.g., those in the public domain) and obvious penetration means are not effective, ensuring that all functional requirements have been met (via review of developer testing if performed and adequate, else performing the testing necessary to come to this conclusion), **and reviewing developer documentation.**

**AL3** represents a meaningful increase in assurance from **AL2** by requiring **additional developer actions that produce a better understanding of the security functionality and more comprehensive testing.**

### 3.4 AL4 – SIGNIFICANT SECURITY ENGINEERING

Overview. AL4 provides a significant gain in component security quality by requiring rigorous commercial development practices coupled with significant application of specialist security engineering knowledge and skills, producing significantly higher quality components. AL4 remains a lower cost evaluation by providing only minimal third-party confirmation of the developer actions. For the most part, developer documentation is taken at face value. It is intended that when a significant third-party validation is required, AL5 or higher will be selected. AL4 provides for greater assurance primarily through developer actions involving positive security engineering using expert knowledge and skills. When these actions are executed to the depth and rigor required at AL4, there is a reasonable expectation of a significant improvement in TOE security quality over AL3 compliant components. This is the evaluation level for moderate quality components in those situations where either the:

- Vendor has been shown to be trustworthy in both the ability to and likelihood of meeting requirements at this assurance level (sufficient assurance without need for extensive evaluation) or
- Marketplace will not pay the cost (perhaps primarily time) of an extensive independent evaluation (accepting added risk due to relying on vendor assertion)

AL4 provides assurance through the requirement for developer analysis of the security functions using a functional and interface specification, a high-level design of the subsystems of the TOE, and a low-level design. This ensures both a better design and a better understanding of the TOE's security behavior. In addition, AL4 requires moderate application of security engineering principles and techniques, to include vulnerability analysis and penetration testing with regard to moderate attack capability.

The National Security Agency's medium robustness guideline, as captured in [3], was used as a basis for the selection of assurance components for AL4. In terms of components included (not degree of evaluation performed), AL4 augments the medium robustness guideline with ACM\_SCP.3 and ALC\_TAT.2 and uses ALC\_CCA.1 in lieu of the associated, extended component in the medium robustness guidelines. Again, in terms of components included and not degree of evaluation performed, the resulting set of assurances is similar to Common Criteria EAL5 with the following differences (all of these differences result from using the medium robustness guideline as a basis for component selection):

<u>AL4 – Similar CC Component</u>	<u>EAL5</u>
ADV_FSP.2	ADV_FSP.3
ADV_HLD.2	ADV_HLD.3
ADV_RCR.1	ADV_RCR.2
ADV_SPM.1	ADV_SPM.3
ALC_FLC.2	none
ALC_LCD.1	ALC_LCD.2.

Description:

**AL4** provides assurance through these developer actions: “gray” box testing, **extensive** configuration management, the development of a functional specification, development of a security policy model, development of a **security-enforcing** high-level design, **development of a low-level design, producing a modular design**, verifying correspondence between design representations, life-cycle support, secure delivery procedures, misuse analysis looking for potential errors in documentation and operating procedures, **and vulnerability analysis and penetration testing with regard to moderate attack capability**.

**AL4** provides evaluation assurance through the evaluator actions of verifying, via penetration testing, that known attacks (e.g., those in the public domain) and obvious penetration means are not effective, ensuring that all functional requirements have been met (via review of developer testing if performed and adequate, else performing the testing necessary to come to this conclusion), and reviewing developer documentation.

**AL4** represents a meaningful increase in assurance from **AL3** by requiring **significant security engineering by the developer**.

### 3.5 AL5 – VERIFIED SIGNIFICANT SECURITY ENGINEERING

Overview. AL5 adds **extensive third-party evaluation** to the rigorous commercial development practices, significant security engineering, limited third-party evaluation of AL4, producing verified, high quality components. Additional developer actions are introduced at AL5 to facilitate the evaluation effort rather than to enhance security quality. AL5 is applicable in those circumstances where a significant increase in component security quality is required and a high level of evaluation assurance is also necessary. This is the evaluation level for moderate quality components in those situations when the:

- Vendor has not been shown to be trustworthy in both the ability to and likelihood of meeting requirements at this assurance level (insufficient assurance necessitating additional assurance via evaluation) and
- Marketplace is willing to pay the costs associated with an extensive independent evaluation (not willing to accept the added risk due to relying on vendor assertion).

AL5 provides assurance through the requirement for developer analysis of the security functions using a functional and interface specification, a high-level design of the subsystems of the TOE, and a low-level design. This ensures both a better design and a better understanding of the TOE's security behavior. AL5 requires moderate application of securing engineering principles and techniques, to include vulnerability analysis and penetration testing with regard to moderate attack capability, and adds independent evaluation to AL4.

The National Security Agency's medium robustness guideline, as captured in [3], was used as a basis for the selection of assurance components for AL5. AL5 augments the medium robustness guideline with ACM\_SCP.3 and ALC\_TAT.2 and uses ALC\_CCA.1 in lieu of the associated, extended component in the medium robustness guidelines. The resulting set of assurances is similar to Common Criteria EAL5 with the following differences (all of these differences result from using the medium robustness guideline as a basis for component selection):

<u>AL5 – Similar CC Component</u>	<u>EAL5</u>
ADV_FSP.2	ADV_FSP.3
ADV_HLD.2	ADV_HLD.3
ADV_RCR.1	ADV_RCR.2
ADV_SPM.1	ADV_SPM.3
ALC_FLC.2	none
ALC_LCD.1	ALC_LCD.2.

Description:

**AL5** provides assurance through these developer actions: “gray” box testing, extensive configuration management, the development of a functional specification, development of a security policy model, development of a security-enforcing high-level design, development of a

low-level design, producing a modular design, verifying correspondence between design representations, life-cycle support, secure delivery procedures, misuse analysis looking for potential errors in documentation and operating procedures, and vulnerability analysis and penetration testing with regard to moderate attack capability.

**AL5** provides **significant** evaluation assurance through the evaluator actions of **searching for vulnerabilities and performing penetration testing to the extent necessary to verify resistance to all penetration attacks at moderate attack potential, confirming the developer test results, testing of the security functions, and extensive analysis and confirmation** of the developer documentation.

**AL5** represents a meaningful increase in assurance from **AL4** by requiring a significant **increase in the scope, depth, and rigor of the third-party evaluation**.

DRAFT

### 3.6 AL6 – RIGOROUS SECURITY ENGINEERING

Overview. AL6 adds **rigorous development processes and rigorous security engineering** to the rigorous commercial development practices, significant security engineering, limited third-party evaluation of AL4, producing high assurance components. This is the evaluation level for high quality components in those situations where either the:

- Vendor has been shown to be trustworthy in both the ability to and likelihood of meeting requirements at this assurance level (sufficient assurance without need for extensive evaluation) or
- Marketplace will not pay the cost (perhaps primarily time) of an extensive independent evaluation (accepting added risk due to relying on vendor assertion)

AL6 provides assurance through the requirement for developer analysis of the security functions using a functional and interface specification, a high-level design of the subsystems of the TOE, and a low-level design. This ensures both a better design and a better understanding of the TOE's security behavior. AL6 requires rigorous application of securing engineering principles and techniques, to include vulnerability analysis and penetration testing with regard to high attack capability.

In terms of components included (not degree of evaluation performed), AL6 is similar to Common Criteria EAL6. The major differences reflect some added assurances, the decision not to use semi-formal as a design documentation requirement (informal is sufficient), and the decision to not require a formal policy model without other formal design representations (limited utility). This results the following differences between AL6 and EAL6:

<u>AL6 – Similar CC Component</u>	<u>EAL6</u>
ADV_FSP.2	ADV_FSP.3
ADV_HLD.4	ADV_HLD.4
ADV_INT.3	ADV_INT.2
ADV_LLD.1	ADV_LLD.2
ADV_RCR.1	ADV_RCR.2
ADV_SPM.1	ADV_SPM.3
ALC_FLC.2	none
ATE_DPT.3	ATE_DPT.2

Description:

**AL6** provides assurance through these developer actions: “gray” box testing, extensive configuration management, the development of a functional specification, development of a security policy model, development of a security-enforcing high-level design, development of a low-level design, producing a modular, **layered, and minimized complexity** design, verifying correspondence between design representations, life-cycle support, secure delivery pro-

cedures, misuse analysis looking for potential errors in documentation and operating procedures, and vulnerability analysis and penetration testing with regard to **high** attack capability.

**AL6** provides **limited** evaluation assurance through the evaluator actions of searching for obvious vulnerabilities, selectively confirming the developer test results, testing of the security functions, and review of the developer documentation.

**AL6** represents a meaningful increase in assurance from AL4 (and potentially AL5) by requiring a significant **increase in the scope, depth, and rigor of the security engineering by the developer**.

DRAFT

### 3.7 AL7 – VERIFIED, RIGOROUS SECURITY ENGINEERING

Overview. AL7 adds **extensive third-party validation** to the rigorous development processes, rigorous security engineering, and limited third-party validation of AL6, producing verified, high assurance components. Additional developer actions are introduced at AL7 to facilitate the evaluation effort rather than to enhance security quality. This is the evaluation level for high quality components in those situations when the:

- Vendor has not been shown to be trustworthy in both the ability to and likelihood of meeting requirements at this assurance level (insufficient assurance necessitating additional assurance via evaluation) and
- Marketplace is willing to pay the costs associated with an extensive independent evaluation (not willing to accept the added risk due to relying on vendor assertion).

AL7 provides assurance through the requirement for developer analysis of the security functions using a functional and interface specification, a high-level design of the subsystems of the TOE, and a low-level design. This ensures both a better design and a better understanding of the TOE's security behavior. AL7 requires rigorous application of securing engineering principles and techniques, to include vulnerability analysis and penetration testing with regard to high attack capability, and adds independent evaluation to AL6.

In terms of components included, AL7 is similar to Common Criteria EAL6. The major differences reflect some added assurances, the decision not to use semi-formal as a design documentation requirement (informal is sufficient), and the decision to not require a formal policy model without other formal design representations (limited utility). This results the following differences between AL7 and EAL6:

<u>AL7 – Similar CC Component</u>	<u>EAL6</u>
ADV_FSP.2	ADV_FSP.3
ADV_HLD.4	ADV_HLD.4
ADV_INT.3	ADV_INT.2
ADV_LLD.1	ADV_LLD.2
ADV_RCR.1	ADV_RCR.2
ADV_SPM.1	ADV_SPM.3
ALC_FLC.2	none
ATE_DPT.3	ATE_DPT.2

#### Description:

**AL7** provides assurance through these developer actions: “gray” box testing, extensive configuration management, the development of a functional specification, development of a security policy model, development of a security-enforcing high-level design, development of a low-level design, producing a modular, layered, and minimized complexity design, verifying



correspondence between design representations, life-cycle support, secure delivery procedures, misuse analysis looking for potential errors in documentation and operating procedures, and vulnerability analysis and penetration testing with regard to high attack capability.

**AL7** provides significant evaluation assurance through the evaluator actions of searching for vulnerabilities and performing penetration testing to the extent necessary to verify resistance to all penetration attacks at **high** attack potential, confirming the developer test results, testing of the security functions, and extensive analysis and confirmation of the developer documentation.

**AL7** represents a potentially meaningful increase in assurance from **AL6** by **adding significant evaluation assurance**.

DRAFT

### 3.8 AL8 – FORMAL METHODS

Overview. AL8 **adds formal methods** to the rigorous development processes, rigorous security engineering, and limited third-party validation of AL6, producing very high assurance components. This is the evaluation level for very high quality components in those situations where either the:

- Vendor has been shown to be trustworthy in both the ability to and likelihood of meeting requirements at this assurance level (sufficient assurance without need for extensive evaluation) or
- Marketplace will not pay the cost (perhaps primarily time) of an extensive independent evaluation (accepting added risk due to relying on vendor assertion)

AL8 provides assurance through the requirement for developer analysis of the security functions using a functional and interface specification, a high-level design of the subsystems of the TOE, and a low-level design. This ensures both a better design and a better understanding of the TOE's security behavior. AL8 requires rigorous application of securing engineering principles, to include, to include vulnerability analysis and penetration testing with regard to high attack capability and use of formal methods.

In terms of components included (not degree of evaluation performed), AL8 is similar to Common Criteria EAL7. The major differences reflect the decision not to use semi-formal as a design documentation requirement (informal is sufficient), and the decision to not require formal design representations for high-level and low-level design (not considered feasible). This results the following differences between AL8 and EAL7:

<u>AL8 – Similar CC Component</u>	<u>EAL7</u>
ADV_HLD.4	ADV_HLD.5
ADV_LLD.1	ADV_LLD.2
ALC_FLC.2	none

#### Description:

**AL8** provides assurance through these developer actions: “gray” box testing, extensive configuration management, the development of a functional specification, development of a security policy model, development of a security-enforcing high-level design, development of a low-level design, producing a modular, layered, and minimized complexity design, verifying correspondence between design representations, life-cycle support, secure delivery procedures, and misuse analysis looking for potential errors in documentation and operating procedures, vulnerability analysis and penetration testing with regard to high attack capability, **and use of formal methods.**

**AL8** provides **limited** evaluation assurance through the evaluator actions of searching for obvious vulnerabilities, selectively confirming the developer test results, testing of the security functions, and review of the developer documentation.

**AL8** represents a meaningful increase in assurance from AL6 (and potentially AL7) by **adding formal methods**.

### 3.9 AL9 – VERIFIED FORMAL METHODS

Overview. AL9 adds **extensive third-party validation** to the formal methods, rigorous development processes, rigorous security engineering, and limited third-party validation of AL8, producing verified, very high assurance components. The additional developer actions at AL9 are introduced to facilitate the evaluation effort rather than to enhance security quality. This is the evaluation level for very high quality components in those situations when the:

- Vendor has not been shown to be trustworthy in both the ability to and likelihood of meeting requirements at this assurance level (insufficient assurance necessitating additional assurance via evaluation) and
- Marketplace is willing to pay the costs associated with an extensive independent evaluation (not willing to accept the added risk due to relying on vendor assertion).

AL9 provides assurance through the requirement for developer analysis of the security functions using a functional and interface specification, a high-level design of the subsystems of the TOE, and a low-level design. This ensures both a better design and a better understanding of the TOE's security behavior. AL8 requires rigorous application of securing engineering principles, to include vulnerability analysis and penetration testing with regard to moderate attack capability and use of formal methods, and adds independent evaluation to AL8.

In terms of components included, AL9 is similar to Common Criteria EAL7. The major differences reflect the decision not to use semi-formal as a design documentation requirement (informal is sufficient), and the decision to not require formal design representations for high-level and low-level design (not considered feasible). This results the following differences between AL9 and EAL7:

<u>AL9 – Similar CC Component</u>	<u>EAL7</u>
ADV_HLD.4	ADV_HLD.5
ADV_LLD.1	ADV_LLD.2
ALC_FLC.2	none

#### Description:

**AL9** provides assurance through these developer actions: “gray” box testing, extensive configuration management, the development of a functional specification, development of a security policy model, development of a security-enforcing high-level design, development of a low-level design, producing a modular, layered, and minimized complexity design, verifying correspondence between design representations, life-cycle support, secure delivery procedures, and misuse analysis looking for potential errors in documentation and operating pro-

cedures, vulnerability analysis and penetration testing with regard to high attack capability, and use of formal methods.

**AL9** provides **significant** evaluation assurance through the evaluator actions of searching for vulnerabilities and performing penetration testing to the extent necessary to verify resistance to all penetration attacks at **high** attack potential, confirming the developer test results, testing of the security functions, and extensive analysis and confirmation of the developer documentation.

**AL9** represents a potentially meaningful increase in assurance from **AL8** by **adding significant evaluation assurance**.

DRAFT

## CHAPTER FOUR

## 4

## 4. ASSURANCE PACKAGES - CONTENTS

### RECOMMENDED ASSURANCE PACKAGES EXPRESSED AS NINE (9) ASSURANCE LEVELS (ALs)

The contents of the nine assurance levels (ALs) are presented in this section and summarized in Table 4-1. This table identifies the assurance components by relating them to comparable, existing common criteria components.

The changes from existing CC components can be summarized as follows:

- For all components, the developer focus has been changed from “documentation for evaluation” to explicit statement of “developer actions that enhance security capability”.
- Developer actions are defined in the developer action elements in lieu of being captured by implication in other types of assurance elements.
- Developer supplied documentation is frequently not the identified means for confirming the developer actions. Instead, the sponsor of the evaluation can choose to provide documentary evidence; presumably when it is determined that providing such documentation is the cost-effective means to obtain the evaluation result. When such evidentiary documentation is not available for evaluator review, the evaluator will need to supplement with other means such as on-site inspection or reverse-engineering. Whether such other means are more cost-effective to the sponsor of the evaluation than providing documentation is not germane to the requirement itself and hence has been abstracted out when feasible.
- The level of evaluation performed is no longer the primary differentiator between assurance levels. The primary differences are now the developer actions and the resulting potential for security quality. Where less evaluation is to be performed than a similar CC component the notation “L” (for evaluation light) is used and when the amount of evaluation performed is not intended to be less the notation “R” (for revised) is used.

Note that while this section presents the assurance components without identifying specific changes from the corresponding CC component, Appendix A lists the assurance components with additions and deletions from the CC counterpart depicted with strike-through and underline.

**Table 4-1 Assurance Level (AL) Contents in terms of similar CC components**

	Similar EAL				EAL3			EAL5			EAL6			EAL7
Assurance Class	Family	AL1	AL2	AL3		AL4	AL5		AL6	AL7		AL8	AL9	
Configuration Management	ACM_AUT					L.1	R.1	1	L.2	R.2	2	L.2	R.2	2
	ACM_CAP	L.1	L.1	L.3	3	L.4	R.4	4	L.5	R.5	5	L.5	R.5	5
	ACM_SCP			L.2	1	L.3	R.3	3	L.3	R.3	3	L.3	R.3	3
Delivery and Operation	ADO_DEL			L.1	1	L.2	R.2	2	L.2	R.2	2	L.3	R.3	3
	ADO_IGS	L.1	L.1	L.1	1	L.1	R.1	1	L.1	R.1	1	L.1	R.1	1
Development	ADV_FSP		L.1	L.1	1	L.2	R.2	3	L.2	R.2	3	L.4	R.4	4
	ADV_HLD			L.1	2	L.2	R.2	3	L.4	R.4	4	L.4	R.4	5
	ADV_IMP					L.2	R.2	2	L.3	R.3	3	L.3	R.3	3
	ADV_INT					L.1	R.1	1	L.3	R.3	2	L.3	R.3	3
	ADV_LLD					L.1	R.1	1	L.1	R.1	2	L.1	R.1	2
	ADV_RCR			L.1	1	L.1	R.1	2	L.1	R.1	2	L.3	R.3	3
	ADV_SPM			L.1		L.1	R.1	3	L.1	R.1	3	L.3	R.3	3
Guidance Documentation	AGD_ADM	L.1	L.1	L.1	1	L.1	R.1	1	L.1	R.1	1	L.1	R.1	1
	AGD_USR	L.1	L.1	L.1	1	L.1	R.1	1	L.1	R.1	1	L.1	R.1	1
Life Cycle Support	ALC_DVS			L.1	1	L.1	R.1	1	L.2	R.2	2	L.2	R.2	2
	ALC_FLR			L.2		L.2	R.2		L.2	R.2		L.2	R.2	
	ALC_LCD			L.1		L.1	R.1	2	L.2	R.2	2	L.3	R.3	3
	ALC_TAT			L.1		L.2	R.2	2	L.3	R.3	3	L.3	R.3	3
Tests	ATE_COV			L.2	2	L.2	R.2	2	L.3	R.3	3	L.3	R.3	3
	ATE_DPT			L.1	1	L.2	R.2	2	L.3	R.3	2	L.3	R.3	3
	ATE_FUN			L.1	1	L.1	R.1	1	L.2	R.2	2	L.2	R.2	2
	ATE_IND		R.1	L.2	2	L.2	R.2	2	L.2	R.2	2	L.3	R.3	3
Vulnerability Assessment	AVA_CCA					L.1	R.1	1	L.2	R.2	2	L.2	R.2	2
	AVA_MSU			L.2	1	L.2	R.2	2	L.3	R.3	3	L.3	R.3	3
	AVA_SOF		L.1	L.1	1	L.1	R.1	1	L.1	R.1	1	L.1	R.1	1
	AVA_VLA	L.2	L.2	L.2	1	L.3	R.3	3	L.4	R.4	4	L.4	R.4	4

## 4.1 AL1 - VULNERABILITY TESTING

### 4.1.1 ACM\_CAP-L.1 Version numbers

Objectives. A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using.

Dependencies: No dependencies.

Developer action elements:

ACM\_CAP-L.1.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ACM\_CAP-L.1.1E The evaluator shall check that the TOE is labeled with a reference that can reasonably be expected to be unique to each version of the TOE.

### 4.1.2 ADO\_IGS-L.1 Installation generation and start-up procedures

Dependencies: AGD\_ADM-L.1 Administrator guidance

Developer action elements:

ADO\_IGS-L.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_IGS-L.1.1E The evaluator shall check that the documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

### 4.1.3 AGD\_ADM-L.1 Administrator guidance

Dependencies: None

Developer action elements:

AGD\_ADM-L.1.1D The developer shall provide administrator guidance addressed to system Administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_ADM-L.1.1C The administrator guidance shall describe the Administrative functions and interfaces available to the administrator of the TOE.

AGD\_ADM-L.1.2C The administrator guidance shall describe how to Administer the TOE in a secure manner.

AGD\_ADM-L.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-L.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-L.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-L.1.6C The administrator guidance shall describe each type of security-relevant event relative to the Administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-L.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-L.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD\_ADM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.1.4 AGD\_USR-L.1 User guidance**

Dependencies: None

Developer action elements:



AGD\_USR-L.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_USR-L.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-L.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-L.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-L.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-L.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-L.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD\_USR-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### 4.1.5AVA\_VLA-L.2 Independent vulnerability analysis

Objectives. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks performed using publicly known attacks and otherwise obvious attack methods.

Dependencies:

AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

Developer action elements:

AVA\_VLA-L.2.1D *CC element deleted*

AVA\_VLA-L.2.2D *CC element deleted.*

AVA\_VLA-L.2.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

Content and presentation of evidence elements:

None

Evaluator action elements:

AVA\_VLA-L.2.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing.

AVA\_VLA-L.2.2E *CC element deleted*

AVA\_VLA-L.2.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.2.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.2.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods.

## 4.2 AL2 - FUNCTIONAL TESTING

### 4.2.1 ACM\_CAP-L.1 Version numbers

Objectives. A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using.

Dependencies: No dependencies.

Developer action elements:

ACM\_CAP-L.1.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ACM\_CAP-L.1.1E The evaluator shall check that the TOE is labeled with a reference that can reasonably be expected to be unique to each version of the TOE.

### 4.2.2 ADO\_IGS-L.1 Installation generation and start-up procedures

Dependencies: AGD\_ADM-L.1 Administrator guidance

Developer action elements:

ADO\_IGS-L.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_IGS-L.1.1E The evaluator shall check that the documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

### 4.2.3 ADV\_FSP-L.1 Informal functional specification

Dependencies: None

Developer action elements:

ADV\_FSP-L.1.1D The developer shall provide an informal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

Content and presentation of evidence elements:

ADV\_FSP-L.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV\_FSP-L.1.2C The functional specification shall be internally consistent.

ADV\_FSP-L.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV\_FSP-L.1.4C The functional specification shall completely represent the TSF.

Evaluator action elements:

ADV\_FSP-L.1.1E The evaluator shall confirm, to the level of rigor of no obvious errors or omissions and at a level of completeness sufficient to support effective functional testing against the functional requirements in the associated security target, that the functional specification meets all requirements for content and presentation of evidence.

### 4.2.4 AGD\_ADM-L.1 Administrator guidance

Dependencies: None

Developer action elements:

AGD\_ADM-L.1.1D The developer shall provide administrator guidance addressed to system Administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_ADM-L.1.1C The administrator guidance shall describe the Administrative functions and interfaces available to the administrator of the TOE.

AGD\_ADM-L.1.2C The administrator guidance shall describe how to Administer the TOE in a secure manner.

AGD\_ADM-L.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-L.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-L.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-L.1.6C The administrator guidance shall describe each type of security-relevant event relative to the Administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-L.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-L.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD\_ADM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.2.5 AGD\_USR-L.1 User guidance**

Dependencies: None

Developer action elements:

AGD\_USR-L.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_USR-L.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-L.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-L.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-L.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-L.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-L.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD\_USR-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.2.6 ATE\_IND-R.1 Independent testing - conformance**

**Objectives:** In this component, the objective is to demonstrate that the security functions perform as specified.

**Application notes:** This component does not address the use of developer test results. It is applicable where such results are not available, and also in cases where the developer's testing is accepted without validation. The evaluator is required to devise and conduct tests with the objective of confirming that the TOE security functional requirements are met..

**Dependencies:**

ADV\_FSP-L.1 Informal functional specification  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

**Developer action elements:**

ATE\_IND-R.1.1D The developer shall provide the TOE suitable for testing.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_IND-R.1.1E The evaluator shall confirm that the developer has provides the TOE suitable for testing.

ATE\_IND-R.1.2E The evaluator shall test the TSF as appropriate to confirm that the TOE meets all functional requirements in the associated security target.

**4.2.7AVA\_SOF-L.1 Strength of TOE security function evaluation****Dependencies:**

ADV\_FSP-L.1 Informal functional specification

**Developer action elements:**

AVA\_SOF-L.1.1D The developer shall perform a strength of TOE security function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

AVA\_SOF-L.1.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer's strength of function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

**4.2.8AVA\_VLA-L.2 Independent vulnerability analysis**

**Objectives.** The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks performed using publicly known attacks and otherwise obvious attack methods.

**Dependencies:**

AGD\_ADM-L.1 Administrator guidance

AGD\_USR-L.1 User guidance

**Developer action elements:**

AVA\_VLA-L.2.1D *CC element deleted*

AVA\_VLA-L.2.2D *CC element deleted.*

AVA\_VLA-L.2.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

Content and presentation of evidence elements:

None

Evaluator action elements:

AVA\_VLA-L.2.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing.

AVA\_VLA-L.2.2E *CC element deleted*

AVA\_VLA-L.2.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.2.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.2.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods.



## 4.3 AL3 - RIGOROUS COTS DEVELOPMENT

### 4.3.1 ACM\_CAP-L.3 Authorization controls

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE.

**Dependencies:**

ACM\_SCP-L.1 TOE CM coverage  
ALC\_DVS-L.1 Identification of security measures

**Developer action elements:**

ACM\_CAP-L.3.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-L.3.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-L.3.3D The developer shall provide a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.3.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_CAP-L.3.1E The evaluator shall check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-L.3.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to uniquely identify all configuration items and provide measures such that only authorized changes are made to the configuration items.

### 4.3.2 ACM\_SCP-L.2 Problem tracking CM coverage

**Objectives:** A CM system can control changes only to those items that have been placed under CM. Placing the TOE implementation representation, design, tests, user and administrator documentation, and CM documentation under CM provides assurance that they have been modified in a controlled manner with proper authorizations. The ability to track security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution.

**Dependencies:** ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ACM\_SCP-L.2.1D The developer shall perform CM such that, as a minimum, the following are kept under CM: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_SCP-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the following are being configuration managed: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.

### 4.3.3 ADO\_DEL-L.1 Delivery procedures

**Dependencies:** No dependencies.

**Developer action elements:**

ADO\_DEL-L.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADO\_DEL-L.1.1E The evaluator shall check that the documented delivery procedures describe the procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.

#### **4.3.4 ADO\_IGS-L.1 Installation generation and start-up procedures**

Dependencies: AGD\_ADM-L.1 Administrator guidance

Developer action elements:

ADO\_IGS-L.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_IGS-L.1.1E The evaluator shall check that the documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

#### **4.3.5 ADV\_FSP-L.1 Informal functional specification**

Dependencies: None

Developer action elements:

ADV\_FSP-L.1.1D The developer shall provide an informal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

Content and presentation of evidence elements:

ADV\_FSP-L.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV\_FSP-L.1.2C The functional specification shall be internally consistent.

ADV\_FSP-L.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV\_FSP-L.1.4C The functional specification shall completely represent the TSF.

Evaluator action elements:

ADV\_FSP-L.1.1E The evaluator shall confirm, to the level of rigor of no obvious errors or omissions and at a level of completeness sufficient to support effective functional testing against the functional requirements in the associated security target, that the functional specification meets all requirements for content and presentation of evidence.

#### **4.3.6 ADV\_HLD-L.1 Descriptive high-level design**

Dependencies:

ADV\_FSP-L.1 Informal functional specification

ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:

ADV\_HLD-L.1.1D The developer shall produce a high-level design of the TSF that provides the information identified in the content and presentation section below.

ADV\_HLD-L.1.2D(+) The developer shall, as an essential part of the TOE development process, produce the high-level design as a precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.1.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_HLD-L.1.1C The presentation of the high-level design shall be informal.

ADV\_HLD-L.1.2C The high-level design shall be internally consistent.

ADV\_HLD-L.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-L.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-L.1.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-L.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-L.1.7C The high-level design shall identify which of the interfaces to the sub-systems of the TSF are externally visible.

Evaluator action elements:

ADV\_HLD-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-L.1.2E *placeholder for element appearing at higher component*

ADV\_HLD-L.1.3E(+) The evaluator shall confirm, the level of rigor of appears to be true, that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.1.4E(+) The evaluator shall confirm, the level of rigor of appears to be true, that the developer design process maintains the high-level design to reflect the actual implementation.

#### **4.3.7 ADV\_RCR-L.1 Informal correspondence demonstration**

Dependencies: No dependencies.

Developer action elements:

ADV\_RCR-L.1.1D The developer shall conduct an informal analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_RCR-L.1.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the results of the developer analysis of correspondence show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

#### **4.3.8 ADV\_SPM-L.1 Informal TOE security policy model**

Dependencies: ADV\_FSP-L.1 Informal functional specification

Developer action elements:

ADV\_SPM-L.1.1D The developer shall produce an informal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.

ADV\_SPM-L.1.2D The developer shall show correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_SPM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the SPM is an informal TSP model that describes the rules and characteristics of the policies of the TSP.

ADV\_SPM-L.1.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has shown correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

#### **4.3.9 AGD\_ADM-L.1 Administrator guidance**

Dependencies: None

Developer action elements:

AGD\_ADM-L.1.1D The developer shall provide administrator guidance addressed to system Administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_ADM-L.1.1C The administrator guidance shall describe the Administrative functions and interfaces available to the administrator of the TOE.

AGD\_ADM-L.1.2C The administrator guidance shall describe how to Administer the TOE in a secure manner.

AGD\_ADM-L.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-L.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-L.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-L.1.6C The administrator guidance shall describe each type of security-relevant event relative to the Administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-L.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-L.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

**Evaluator action elements:**

AGD\_ADM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

**4.3.10 AGD\_USR-L.1 User guidance**

Dependencies: None

**Developer action elements:**

AGD\_USR-L.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

**Content and presentation of evidence elements:**

AGD\_USR-L.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-L.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-L.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-L.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-L.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-L.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

**Evaluator action elements:**

AGD\_USR-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.3.11 ALC\_DVS-L.1 Identification of security measures**

Dependencies: No dependencies.

**Developer action elements:**

ALC\_DVS-L.1.1D The developer shall identify the physical, procedural, personnel, and other security measures that are deemed necessary by the developer to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-L.1.2D(+) The developer shall implement the measures identified.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ALC\_DVS-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the physical, procedural, personnel, and other security measures that are deemed necessary by the developer to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-L.1.2E The evaluator shall confirm, to the level of rigor of appears to be true, that the security measures are being applied.

#### **4.3.12 ALC\_FLR-L.2 Flaw reporting procedures**

Dependencies: No dependencies.

**Developer action elements:**

ALC\_FLR-L.2.1D The developer shall establish flaw remediation procedures that provide the capabilities included in the content and presentation section below.



ALC\_FLR-L.2.2D The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

Content and presentation of evidence elements:

ALC\_FLR-L.2.1C The flaw remediation procedures shall include tracking of all reported security flaws in each release of the TOE.

ALC\_FLR-L.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC\_FLR-L.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC\_FLR-L.2.4C The flaw remediation procedures shall include providing flaw information, corrections and guidance on corrective actions to TOE users.

ALC\_FLR-L.2.5C The procedures for processing reported security flaws shall include measures to ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC\_FLR-L.2.6C The procedures for processing reported security flaws shall include safeguards to reduce the potential for corrections to these security flaws to introduce new flaws.

Evaluator action elements:

ALC\_FLR-L.2.1E The evaluator shall confirm, to the level of rigor of appears true, that the flaw remediation procedures meet all the requirements identified in the content and presentation of evidence section above.

ALC\_FLR-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears true, that the flaw remediation procedures include provisions for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

### 4.3.13 ALC\_LCD-L.1 Developer defined life-cycle model

Dependencies: No dependencies.

Developer action elements:

ALC\_LCD-L.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_LCD-L.1.1E The evaluator shall confirm, to the level of rigor of appears true, that the developer has established a life-cycle model to be used in the development and maintenance of the TOE.

#### 4.3.14 ALC\_TAT-L.1 Well-defined development tools

Dependencies: None

Developer action elements:

ALC\_TAT-L.1.1D The developer shall use well-defined development tools for the TOE.

ALC\_TAT-L.1.2D The developer shall identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.1.3D(+) *placeholder for element appearing in higher component*

ALC\_TAT-L.1.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_TAT-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using well-defined development tools for the TOE.

ALC\_TAT-L.1.2E(+) *placeholder for element appearing in higher component*

ALC\_TAT-L.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.1.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### 4.3.15 ATE\_COV-L.2 Analysis of coverage

**Objectives:** In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved by the developer conducting testing on the basis of an analysis of correspondence.

**Application notes:** The developer is required to have conducted testing of all of the security functions as described in the functional specification on the basis of an analysis that shows the correspondence between tests and security functions.

**Dependencies:**

ADV\_FSP-L.1 Informal functional specification

ATE\_FUN-L.1 Functional testing

**Developer action elements:**

ATE\_COV-L.2.1D The developer shall conduct testing on the basis of an analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_COV-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer testing was performed on the basis of an analysis of test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

#### 4.3.16 ATE\_DPT-L.1 Testing: high-level design

**Objectives:** The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized.

**Application notes:** The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts.

**Dependencies:**

ADV\_HLD-L.1 Descriptive high-level design

ATE\_FUN-L.1 Functional testing

Developer action elements:

ATE\_DPT-L.1.1D The developer shall conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

Content and presentation of evidence elements:

None

Evaluator action elements:

ATE\_DPT-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that developer testing was conducted on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

#### 4.3.17 ATE\_FUN-L.1 Functional testing

Objectives: The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation.

Dependencies: No dependencies.

Developer action elements:

ATE\_FUN-L.1.1D The developer shall test the TSF.

ATE\_FUN-L.1.2D The developer shall produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

Content and presentation of evidence elements:

ATE\_FUN-L.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-L.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-L.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE\_FUN-L.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-L.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements:

ATE\_FUN-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the that the developer tested the TSF.

ATE\_FUN-L.1.2E(+) *placeholder for element appearing at higher component*

ATE\_FUN-L.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.

#### **4.3.18 ATE\_IND-L.2 Independent testing - sample**

Objectives: The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes selecting and repeating a sample of the developer tests.

Application notes: The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. This component contains a requirement that the evaluator has available test results from the developer to supplement the program of testing. The evaluator will repeat a sample of the developer's tests to gain confidence in the results obtained. Having established such confidence the evaluator will build upon the developer's testing, as necessary to confirm the developer testing, by conducting additional tests that exercise the TOE in a different manner. By using a platform of validated developer test results the evaluator is able to gain confidence that the TOE operates correctly in a wider range of conditions than would be possible purely using the developer's own efforts, given a fixed level of resource. Having gained confidence that the developer has tested the TOE, the evaluator will also have more freedom, where appropriate, to concentrate testing in areas where examination of documentation or specialist knowledge has raised particular concerns.

Dependencies:

ADV\_FSP-L.1 Informal functional specification  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance  
ATE\_FUN-L.1 Functional testing

Developer action elements:

ATE\_IND-L.2.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Content and presentation of evidence elements:

None

Evaluator action elements:

ATE\_IND-L.2.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE\_IND-L.2.2E The evaluator shall test the TSF only as necessary to confirm that the developer testing shows that the TOE meets all functional requirements in the associated security target.

ATE\_IND-L.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

#### **4.3.19 AVA\_MSU-L.2 Validation of analysis**

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met.

Dependencies:

ADO\_IGS-L.1 Installation, generation, and start-up procedures  
ADV\_FSP-L.1 Informal functional specification  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

Developer action elements:

AVA\_MSU-L.2.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-L.2.2D The developer shall analyze the guidance documentation to determine that the guidance documentation is complete.

#### Content and presentation of evidence elements:

AVA\_MSU-L.2.1C The guidance documentation shall identify possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-L.2.2C *CC element incorporated into developer elements*

AVA\_MSU-L.2.3C The guidance documentation shall list assumptions about the intended environment.

AVA\_MSU-L.2.4C The guidance documentation shall list requirements for external security measures (including external procedural, physical and personnel controls).

#### Evaluator action elements:

AVA\_MSU-L.2.1E The evaluator shall check that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-L.2.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-L.2.3E The evaluator shall determine, to the level of rigor of appears to be true, that the use of the guidance documentation allows insecure states to be detected.

AVA\_MSU-L.2.4E The evaluator shall confirm, to the level of rigor of appears reasonable, that the analysis documentation shows that guidance is provided for secure operation in the modes of operation of the TOE.

### 4.3.20 AVA\_SOF-L.1 Strength of TOE security function evaluation

#### Dependencies:

ADV\_FSP-L.1 Informal functional specification

#### Developer action elements:

AVA\_SOF-L.1.1D The developer shall perform a strength of TOE security function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

#### Content and presentation of evidence elements:

None

Evaluator action elements:

AVA\_SOF-L.1.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer's strength of function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

#### **4.3.21 AVA\_VLA-L.2 Independent vulnerability analysis**

Objectives: The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks performed using publicly known attacks and otherwise obvious attack methods.

Dependencies:

AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

Developer action elements:

AVA\_VLA-L.2.1D *CC element deleted*

AVA\_VLA-L.2.2D *CC element deleted*

AVA\_VLA-L.2.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

Content and presentation of evidence elements:

None

Evaluator action elements:

AVA\_VLA-L.2.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing.

AVA\_VLA-L.2.2E *CC element deleted*

AVA\_VLA-L.2.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods.



AVA\_VLA-L.2.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.2.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods.

**DRAFT**

## 4.4 AL4 – SIGNIFICANT SECURITY ENGINEERING

### 4.4.1 ACM\_AUT-L.1 Partial CM automation

**Objectives:** In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorized. It is the objective of this component to ensure that the implementation representation is controlled through automated means.

**Dependencies:** ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ACM\_AUT-L.1.1D The developer shall use a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation.

ACM\_AUT-L.1.1.2D *placeholder for element appearing in higher component*

ACM\_AUT-L.1.3D(+) The developer shall use a CM system that provides automated means to support the generation of the TOE.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_AUT-L.1.1E The evaluator shall check that the developer is using a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation

ACM\_AUT-L.1.2E(+) The evaluator shall check that the developer is using a CM system that includes automated means to support the generation of the TOE.

### 4.4.2 ACM\_CAP-L.4 Generation support and acceptance procedures

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE. The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized.

**Dependencies:**

ACM\_SCP-L.1 TOE CM coverage  
ALC\_DVS-L.1 Identification of security measures

**Developer action elements:**

ACM\_CAP-L.4.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-L.4.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-L.4.3D The developer shall provide a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.4.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

ACM\_CAP-L.4.5D The developer shall use a CM system that supports the generation of the TOE.

ACM\_CAP-L.4.6D The developer shall use a CM system that includes procedures used to accept modified or newly created configuration items as part of the TOE.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_CAP-L.4.1E The evaluator shall check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-L.4.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.4.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to uniquely identify all configuration items.

ACM\_CAP-L.4.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to provide measures such that only authorized changes are made to the configuration items.

ACM\_CAP-L4.5E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to support generation of the TOE.

ACM\_CAP-L4.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to accept modified or newly created configuration items as part of the TOE.

#### **4.4.3 ACM\_SCP-L.3 Development tools CM coverage**

**Objectives:** A CM system can control changes only to those items that have been placed under CM. Placing the TOE implementation representation, design, tests, user and administrator documentation, and CM documentation under CM provides assurance that they have been modified in a controlled manner with proper authorizations. The ability to track security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution. Development tools play an important role in ensuring the production of a quality version of the TOE. Therefore, it is important to control modifications to these tools.

**Dependencies:** ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ACM\_SCP-L.3.1D The developer shall perform CM such that, as a minimum, the following are kept under CM: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_SCP-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the following are being configuration managed: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

#### **4.4.4 ADO\_DEL-L.2 Detection of modification**

**Dependencies:** ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ADO\_DEL-L.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures necessary to maintain security when distributing versions of the TOE to a user's site.

*ADO\_DEL-L.2.2D CC element deleted*

ADO\_DEL-L.2.3D(+) The developer shall explain how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-L.2.4D(+) The developer shall explain how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADO\_DEL-L.2.1E The evaluator shall check that the documented delivery procedures describe the procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.

ADO\_DEL-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has explained how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-L.2.3E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has explained how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

**4.4.5 ADO\_IGS-L.1 Installation generation and start-up procedures**

Dependencies: AGD\_ADM-L.1 Administrator guidance

**Developer action elements:**

ADO\_IGS-L.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADO\_IGS-L.1.1E The evaluator shall check that the documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

#### **4.4.6 ADV\_FSP-L.2 Fully defined external interfaces**

Dependencies: ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:

ADV\_FSP-L.2.1D The developer shall provide an informal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-L.2.2D(+) The developer shall show that the TSF is a completely and correctly represented by this functional specification.

Content and presentation of evidence elements:

ADV\_FSP-L.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV\_FSP-L.2.2C The functional specification shall be internally consistent.

ADV\_FSP-L.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

Evaluator action elements:

ADV\_FSP-L.2.1E The evaluator shall confirm, to the level of rigor of no obvious errors or omissions and at a level of completeness sufficient to support effective functional testing against the functional requirements in the associated security target that, the functional specification meets all requirements for content and presentation of evidence.

ADV\_FSP-L.2.2E The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has shown that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

#### 4.4.7 ADV\_HLD-L.2 Security enforcing high-level design

##### Dependencies:

ADV\_FSP-L.1 Informal functional specification

ADV\_RCR-L.1 Informal correspondence demonstration

##### Developer action elements:

ADV\_HLD-L.2.1D The developer shall produce a high-level design of the TSF that provides the information identified in the content and presentation section below.

ADV\_HLD-L.2.2D(+) The developer shall, as an essential part of the TOE development process, produce the high-level design as a precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.2.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

##### Content and presentation of evidence elements:

ADV\_HLD-L.2.1C The presentation of the high-level design shall be informal.

ADV\_HLD-L.2.2C The high-level design shall be internally consistent.

ADV\_HLD-L.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-L.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-L.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-L.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-L.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV\_HLD-L.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_HLD-L.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements:

ADV\_HLD-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-L.1.2E *placeholder for element appearing at higher component*

ADV\_HLD-L.2.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.2.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process maintains the high-level design to reflect the actual implementation.

#### 4.4.8 ADV\_IMP-L.2 Implementation of the TSF

Application notes

Dependencies:

ADV\_LLD-L.1 Descriptive low-level design

ADV\_RCR-L.1 Informal correspondence demonstration

ALC\_TAT-L.1 Well-defined development tools

Developer action elements:

ADV\_IMP-L.2.1D The developer shall produce an implementation representation for the entire TSF that unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_IMP-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the implementation representation unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.



#### **4.4.9 ADV\_INT-L.1 Modularity**

**Dependencies:**

ADV\_IMP-L.1 Subset of the implementation of the TSF

ADV\_LLD-L.1 Descriptive low-level design

**Developer action elements:**

ADV\_INT-L.1.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADV\_INT-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed, structured, and implemented the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

#### **4.4.10 ADV\_LLD-L.1 Descriptive low-level design**

**Dependencies:**

ADV\_HLD-L.2 Security enforcing high-level design

ADV\_RCR-L.1 Informal correspondence demonstration

**Developer action elements:**

ADV\_LLD-L.1.1D The developer shall provide the low-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_LLD-L.1.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a low-level design.

ADV\_LLD-L.1.3D(+) The developer shall, as an essential part of the TOE development process, maintain the low-level design to reflect the actual implementation.

**Content and presentation of evidence elements:**

ADV\_LLD-L.1.1C The presentation of the low-level design shall be informal.

ADV\_LLD-L.1.2C The low-level design shall be internally consistent.

ADV\_LLD-L.1.3C The low-level design shall describe the TSF in terms of modules.

ADV\_LLD-L.1.4C The low-level design shall describe the purpose of each module.

ADV\_LLD-L.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV\_LLD-L.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV\_LLD-L.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV\_LLD-L.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV\_LLD-L.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_LLD-L.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

ADV\_LLD-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the documented low-level design meets all requirements for content and presentation of evidence.

ADV\_LLD-L.1.2E *CC element deleted* ADV\_LLD-L.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process produces the low-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_LLD-L.1.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process maintains the low-level design to reflect the actual implementation.

#### **4.4.11 ADV\_RCR-L.1 Informal correspondence demonstration**

Dependencies: No dependencies.

Developer action elements:

ADV\_RCR-L.1.1D The developer shall conduct an informal analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_RCR-L.1.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the results of the developer analysis of correspondence show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

#### **4.4.12 ADV\_SPM-L.1 Informal TOE security policy model**

Dependencies: ADV\_FSP-L.1 Informal functional specification

Developer action elements:

ADV\_SPM-L.1.1D The developer shall produce an informal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.

ADV\_SPM-L.1.2D The developer shall show correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_SPM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the SPM is an informal TSP model that describes the rules and characteristics of the policies of the TSP.

ADV\_SPM-L.1.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has shown correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

#### 4.4.13 AGD\_ADM-L.1 Administrator guidance

Dependencies: None

Developer action elements:

AGD\_ADM-L.1.1D The developer shall provide administrator guidance addressed to system Administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_ADM-L.1.1C The administrator guidance shall describe the Administrative functions and interfaces available to the administrator of the TOE.

AGD\_ADM-L.1.2C The administrator guidance shall describe how to Administer the TOE in a secure manner.

AGD\_ADM-L.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-L.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-L.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-L.1.6C The administrator guidance shall describe each type of security-relevant event relative to the Administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-L.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-L.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD\_ADM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### 4.4.14 AGD\_USR-L.1 User guidance

Dependencies: None

Developer action elements:

AGD\_USR-L.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_USR-L.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-L.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-L.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-L.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-L.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-L.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD\_USR-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.4.15 ALC\_DVS-L.1 Identification of security measures**

Dependencies: No dependencies.

Developer action elements:

ALC\_DVS-L.1.1D The developer shall identify the physical, procedural, personnel, and other security measures that are deemed necessary by the developer to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-L.1.2D(+) The developer shall implement the measures identified.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_DVS-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the physical, procedural, personnel, and other security measures that are deemed necessary by the developer to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-L.1.2E The evaluator shall confirm, to the level of rigor of appears to be true, that the security measures are being applied.

#### **4.4.16 ALC\_FLR-L.2 Flaw reporting procedures**

Dependencies: No dependencies.

Developer action elements:

ALC\_FLR-L.2.1D The developer shall establish flaw remediation procedures that provide the capabilities included in the content and presentation section below.

ALC\_FLR-L.2.2D The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

Content and presentation of evidence elements:

ALC\_FLR-L.2.1C The flaw remediation procedures shall include tracking of all reported security flaws in each release of the TOE.

ALC\_FLR-L.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC\_FLR-L.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC\_FLR-L.2.4C The flaw remediation procedures shall include providing flaw information, corrections and guidance on corrective actions to TOE users.

ALC\_FLR-L.2.5C The procedures for processing reported security flaws shall include measures to ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC\_FLR-L.2.6C The procedures for processing reported security flaws shall include safeguards to reduce the potential for corrections to these security flaws to introduce new flaws.

Evaluator action elements:

ALC\_FLR-L.2.1E The evaluator shall confirm, to the level of rigor of appears true, that the flaw remediation procedures meet all the requirements identified in the content and presentation of evidence section above.

ALC\_FLR-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears true, that the flaw remediation procedures include provisions for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

#### **4.4.17 ALC\_LCD-L.1 Developer defined life-cycle model**

Dependencies: No dependencies.

Developer action elements:

ALC\_LCD-L.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_LCD-L.1.1E The evaluator shall confirm, to the level of rigor of appears true, that the developer has established a life-cycle model to be used in the development and maintenance of the TOE.

#### 4.4.18 ALC\_TAT-L.2 Compliance with implementation standards

Dependencies: ADV\_IMP-L.1 Subset of the implementation of the TSF

Developer action elements:

ALC\_TAT-L.2.1D The developer shall use well-defined development tools for the TOE.

ALC\_TAT-L.2.2D The developer shall identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.2.3D The developer shall apply identified implementation standards.

ALC\_TAT-L.2.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_TAT-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using well-defined development tools for the TOE.

ALC\_TAT-L.2.2E The evaluator shall confirm, to the level of rigor of appears to be true, that the identified implementation standards have been applied.

ALC\_TAT-L.2.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.2.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### 4.4.19 ATE\_COV-L.2 Analysis of coverage

Objectives: In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved by the developer conducting testing on the basis of an analysis of correspondence.

Application notes: The developer is required to have conducted testing of all of the security functions as described in the functional specification on the basis of an analysis that shows the correspondence between tests and security functions.



**Dependencies:**

ADV\_FSP-L.1 Informal functional specification  
ATE\_FUN-L.1 Functional testing

**Developer action elements:**

ATE\_COV-L.2.1D The developer shall conduct testing on the basis of an analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_COV-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer testing was performed on the basis of an analysis of test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

**4.4.20 ATE\_DPT-L.2 Testing: low-level design**

**Objectives:** The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized. The modules of a TSF provide a description of the internal workings of the TSF. Testing at the level of the modules, in order to demonstrate the presence of any flaws, provides assurance that the TSF modules have been correctly realized.

**Application notes:** The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts. The developer is expected to describe the testing of the low-level design of the TSF in terms of "modules". The term "modules" is used to express the notion of decomposing each of the "subsystems" of the TSF into a relatively small number of parts.

**Dependencies:**

ADV\_HLD-L.1 Descriptive high-level design  
ADV\_LLD-L.1 Descriptive low-level design  
ATE\_FUN-L.1 Functional testing

**Developer action elements:**

ATE\_DPT-L.2.1D The developer shall conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and its low-level design.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_DPT-L.2.1E The evaluator shall confirm that developer testing was conducted conduct testing on the basis of an analysis of the depth of testing that demonstrates, to the level of rigor of appears reasonable, that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and its low-level design.

**4.4.21 ATE\_FUN-L.1 Functional testing**

**Objectives:** The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation.

**Dependencies:** No dependencies.

**Developer action elements:**

ATE\_FUN-L.1.1D The developer shall test the TSF.

ATE\_FUN-L.1.2D The developer shall produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

**Content and presentation of evidence elements:**

ATE\_FUN-L.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-L.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-L.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE\_FUN-L.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-L.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements:

ATE\_FUN-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the that the developer tested the TSF.

ATE\_FUN-L.1.2E(+) *placeholder for element appearing at higher component*

ATE\_FUN-L.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.

#### 4.4.22 ATE\_IND-L.2 Independent testing - sample

Objectives: The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes selecting and repeating a sample of the developer tests.

Application notes: The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. This component contains a requirement that the evaluator has available test results from the developer to supplement the program of testing. The evaluator will repeat a sample of the developer's tests to gain confidence in the results obtained. Having established such confidence the evaluator will build upon the developer's testing, as necessary to confirm the developer testing, by conducting additional tests that exercise the TOE in a different manner. By using a platform of validated developer test results the evaluator is able to gain confidence that the TOE operates correctly in a wider range of conditions than would be possible purely using the developer's own efforts, given a fixed level of resource. Having gained confidence that the developer has tested the TOE, the evaluator will also have more freedom, where appropriate, to concentrate testing in areas where examination of documentation or specialist knowledge has raised particular concerns.

Dependencies:

ADV\_FSP-L.1 Informal functional specification  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance  
ATE\_FUN-L.1 Functional testing

Developer action elements:

ATE\_IND-L.2.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Content and presentation of evidence elements:

None

Evaluator action elements:

ATE\_IND-L.2.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE\_IND-L.2.2E The evaluator shall test the TSF only as necessary to confirm that the developer testing shows that the TOE meets all functional requirements in the associated security target.

ATE\_IND-L.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

#### **4.4.23 AVA\_CCA-L.1 Covert channel analysis**

Objectives: The objective is to identify covert channels that are identifiable, through an informal search for covert channels.

Dependencies:

ADV\_FSP-L.2 Fully defined external interfaces

ADV\_IMP-L.2 Implementation of the TSF

AGD\_ADM-L.1 Administrator guidance

AGD\_USR-L.1 User guidance

Developer action elements:

AVA\_CCA-L.1.1D The developer shall conduct a search for covert channels for each information flow control policy; identifying each covert channel, estimating its capacity, and determining the worst-case exploitation scenario.

Content and presentation of evidence elements:

None

**Evaluator action elements:**

AVA\_CCA-L.1.1E The evaluator shall check that the developer has identified covert channels and for each channel estimated its capacity, and determined the worst-case exploitation scenario.

**4.4.24 AVA\_MSU-L.2 Validation of analysis**

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met.

**Dependencies:**

ADO\_IGS-L.1 Installation, generation, and start-up procedures  
ADV\_FSP-L.1 Informal functional specification  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

**Developer action elements:**

AVA\_MSU-L.2.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-L.2.2D The developer shall analyze the guidance documentation to determine that the guidance documentation is complete.

**Content and presentation of evidence elements:**

AVA\_MSU-L.2.1C The guidance documentation shall identify possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-L.2.2C *CC element incorporated into developer elements*

AVA\_MSU-L.2.3C The guidance documentation shall list assumptions about the intended environment.

AVA\_MSU-L.2.4C The guidance documentation shall list requirements for external security measures (including external procedural, physical and personnel controls).

**Evaluator action elements:**

AVA\_MSU-L.2.1E The evaluator shall check that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-L.2.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-L.2.3E The evaluator shall determine, to the level of rigor of appears to be true, that the use of the guidance documentation allows insecure states to be detected.

AVA\_MSU-L.2.4E The evaluator shall confirm, to the level of rigor of appears reasonable, that the analysis documentation shows that guidance is provided for secure operation in the modes of operation of the TOE.

#### **4.4.25 AVA\_SOF-L.1 Strength of TOE security function evaluation**

Dependencies:

ADV\_FSP-L.1 Informal functional specification

Developer action elements:

AVA\_SOF-L.1.1D The developer shall perform a strength of TOE security function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

Content and presentation of evidence elements:

None

Evaluator action elements:

AVA\_SOF-L.1.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer's strength of function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

#### 4.4.26 AVA\_VLA-L.3 Moderately resistant

**Objectives:** A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks using publicly known attacks and otherwise obvious attack methods. The developer performs penetration testing, supported by a vulnerability analysis, to determine that the TOE is resistant to penetration attacks by attackers possessing a moderate attack potential.

**Dependencies:**

ADV\_FSP-L.1 Informal functional specification  
ADV\_HLD-L.2 Security enforcing high-level design  
ADV\_IMP-L.1 Subset of the implementation of the TSF  
ADV\_INT-L.1 Modularity  
ADV\_LLD-L.1 Descriptive low-level design  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance  
FPT\_RVM.1 Non-Bypassability of the TSP  
FPT\_SEP.2 SFP domain separation

**Developer action elements:**

AVA\_VLA-L.3.1D The developer shall perform a systematic analysis of the TOE searching for ways in which a user possessing a moderate attack potential can violate the TSP.

AVA\_VLA-L.3.2D The developer shall conduct penetration testing based upon this analysis to determine that the TOE is resistant to penetration attacks by attackers possessing a moderate attack potential.

AVA\_VLA-L.3.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

AVA\_VLA-L.3.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing.

AVA\_VLA-L.3.2E *CC element deleted*

AVA\_VLA-L.3.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.3.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.3.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods.

AVA\_VLA-L.3.6E(+) The evaluator shall determine, to the level of rigor of appears reasonable, that the developer has performed penetration testing on the basis of a systematic vulnerability analysis to determine that the TOE is resistant to penetration attacks by attackers possessing moderate attack capability.

DRAFT



## 4.5 AL5 – VERIFIED SECURITY ENGINEERING

### 4.5.1 ACM\_AUT-R.1 Partial CM automation

**Objectives:** In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorized. It is the objective of this component to ensure that the implementation representation is controlled through automated means.

**Dependencies:** ACM\_CAP-R.3 Authorization controls

**Developer action elements:**

ACM\_AUT-R.1.1D The developer shall use a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation.

ACM\_AUT-R.1.2D The developer shall provide a CM plan.

ACM\_AUT-R.1.3D(+) The developer shall use a CM system that provides automated means to support the generation of the TOE.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_AUT-R.1.1E The evaluator shall confirm that the developer is using a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation.

ACM\_AUT-R.1.2E(+) The evaluator shall confirm that the developer is using a CM system that includes automated means to support the generation of the TOE.

### 4.5.2 ACM\_CAP-R.4 Generation support and acceptance procedures

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE. The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized.

**Dependencies:**

ACM\_SCP-R.1 TOE CM coverage  
ALC\_DVS-R.1 Identification of security measures

**Developer action elements:**

ACM\_CAP-R.4.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-R.4.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-R.4.3D The developer shall provide a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-R.4.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

ACM\_CAP-R.4.5D The developer shall use a CM system that supports the generation of the TOE.

ACM\_CAP-R.4.6D The developer shall use a CM system that includes procedures used to accept modified or newly created configuration items as part of the TOE.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_CAP-R.4.1E The evaluator shall check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-R.4.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-R.4.3E(+) The evaluator shall confirm that the CM system is being used to uniquely identify all configuration items.

ACM\_CAP-R.4.4E(+) The evaluator shall confirm that the CM system is being used to provide measures such that only authorized changes are made to the configuration items.

ACM\_CAP-R.4.5E(+) The evaluator shall confirm that the CM system is being used to support generation of the TOE.

ACM\_CAP-R.4.6E(+) The evaluator shall confirm that the CM system is being used to accept modified or newly created configuration items as part of the TOE.

### 4.5.3 ACM\_SCP-R.3 Development tools CM coverage

**Objectives:** A CM system can control changes only to those items that have been placed under CM. Placing the TOE implementation representation, design, tests, user and administrator documentation, and CM documentation under CM provides assurance that they have been modified in a controlled manner with proper authorizations. The ability to track security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution. Development tools play an important role in ensuring the production of a quality version of the TOE. Therefore, it is important to control modifications to these tools.

**Dependencies:** ACM\_CAP-R.3 Authorization controls

**Developer action elements:**

ACM\_SCP-R.3.1D The developer shall perform CM such that, as a minimum, the following are kept under CM: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_SCP-R.3.1E The evaluator shall confirm that the following are being configuration managed: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

### 4.5.4 ADO\_DEL-R.2 Detection of modification

**Dependencies:** ACM\_CAP-R.3 Authorization controls

**Developer action elements:**

ADO\_DEL-R.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures necessary to maintain security when distributing versions of the TOE to a user's site.

ADO\_DEL-R.2.2D *CC element deleted*

ADO\_DEL-R.2.3D(+) The developer shall explain how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-R.2.4D(+) The developer shall explain how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_DEL-R.2.1E The evaluator shall confirm that the documented delivery procedures describe the procedures necessary to maintain security when distributing versions of the TOE to a user's site.

ADO\_DEL-R.2.2E(+) The evaluator shall confirm that the developer has explained how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-R.2.3E(+) The evaluator shall confirm that the developer has explained how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

#### **4.5.5 ADO\_IGS-R.1 Installation generation and start-up procedures**

Dependencies: AGD\_ADM-R.1 Administrator guidance

Developer action elements:

ADO\_IGS-R.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_IGS-R.1.1E The evaluator shall confirm that the documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

ADO\_IGS-R.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

#### **4.5.6 ADV\_FSP-R.2 Fully defined external interfaces**

Dependencies: ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_FSP-R.2.1D The developer shall provide an informal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-R.2.2D(+) The developer shall show that the TSF is a completely and correctly represented by this functional specification.

Content and presentation of evidence elements:

ADV\_FSP-R.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV\_FSP-R.2.2C The functional specification shall be internally consistent.

ADV\_FSP-R.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

Evaluator action elements:

ADV\_FSP-R.2.1E The evaluator shall confirm, to the level of rigor for completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target, that the functional specification meets all requirements for content and presentation of evidence.

ADV\_FSP-R.2.2E The evaluator shall determine that the developer has shown that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

#### **4.5.7 ADV\_HLD-R.2 Security enforcing high-level design**

Dependencies:

ADV\_FSP-R.1 Informal functional specification

ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_HLD-R.2.1D The developer shall produce a high-level design of the TSF that provides the information identified in the content and presentation section below.

ADV\_HLD-R.2.2D(+) The developer shall, as an essential part of the TOE development process, produce the high-level design as a precursor for and input to the development of the TOE implementation.

ADV\_HLD-R.2.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

**Content and presentation of evidence elements:**

ADV\_HLD-R.2.1C The presentation of the high-level design shall be informal.

ADV\_HLD-R.2.2C The high-level design shall be internally consistent.

ADV\_HLD-R.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-R.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-R.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-R.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-R.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV\_HLD-R.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_HLD-R.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

**Evaluator action elements:**

ADV\_HLD-R.2.1E The evaluator shall confirm that the documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-R.2.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_HLD-R.2.3E(+) The evaluator shall confirm that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-R.2.4E(+) The evaluator shall confirm that the developer design process maintains the high-level design to reflect the actual implementation.

#### **4.5.8 ADV\_IMP-R.2 Implementation of the TSF**

Application notes: The ADV\_IMP-R.2.2E element defines a requirement that the evaluator determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements. This provides a direct correspondence between the TOE security functional requirements and the implementation representation, in addition to the pairwise correspondences required by the ADV\_RCR family. It is expected that the evaluator will use the evidence provided in ADV\_RCR as an input to making this determination.

Dependencies:

ADV\_LLD-R.1 Descriptive low-level design  
ADV\_RCR-R.1 Informal correspondence demonstration  
ALC\_TAT-R.1 Well-defined development tools

Developer action elements:

ADV\_IMP-R.2.1D The developer shall produce an implementation representation for the entire TSF that unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_IMP-R.2.1E The evaluator shall confirm that the implementation representation unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-R.2.2E The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.

#### **4.5.9 ADV\_INT-R.1 Modularity**

Dependencies:

ADV\_IMP-R.1 Subset of the implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design

Developer action elements:

ADV\_INT-R.1.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_INT-R.1.1E The evaluator shall determine that the developer designed, structured, and implemented the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design

#### **4.5.10 ADV\_LLD-R.1 Descriptive low-level design**

Dependencies:

ADV\_HLD-R.2 Security enforcing high-level design  
ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_LLD-R.1.1D The developer shall provide the low-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_LLD-R.1.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a low-level design.

ADV\_LLD-R.1.3D(+) The developer shall, as an essential part of the TOE development process, maintain the low-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_LLD-R.1.1C The presentation of the low-level design shall be informal.

ADV\_LLD-R.1.2C The low-level design shall be internally consistent.

ADV\_LLD-R.1.3C The low-level design shall describe the TSF in terms of modules.

ADV\_LLD-R.1.4C The low-level design shall describe the purpose of each module.

ADV\_LLD-R.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV\_LLD-R.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV\_LLD-R.1.7C The low-level design shall identify all interfaces to the modules of the TSF.



ADV\_LLD-R.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV\_LLD-R.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_LLD-R.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

ADV\_LLD-R.1.1E The evaluator shall confirm that the documented low-level design meets all requirements for content and presentation of evidence.

ADV\_LLD-R.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_LLD-R.1.3E(+) The evaluator shall confirm that the developer design process produces the low-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_LLD-R.1.4E(+) The evaluator shall confirm that the developer design process maintains the low-level design to reflect the actual implementation.

#### **4.5.11 ADV\_RCR-R.1 Informal correspondence demonstration**

Dependencies: No dependencies.

Developer action elements:

ADV\_RCR-R.1.1D The developer shall conduct an analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_RCR-R.1.1E The evaluator shall confirm that the results of the developer analysis of correspondence show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

#### **4.5.12 ADV\_SPM-R.1 Informal TOE security policy model**

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

ADV\_SPM-R.1.1D The developer shall produce an informal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.

ADV\_SPM-R.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_SPM-R.1.1E The evaluator shall confirm that the SPM is an informal TSP model and describes the rules and characteristics of the policies of the TSP.

ADV\_SPM-R.1.2E(+) The evaluator shall confirm that the developer has demonstrated correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

#### **4.5.13 AGD\_ADM-R.1 Administrator guidance**

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

AGD\_ADM-R.1.1D The developer shall provide administrator guidance addressed to system administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_ADM-R.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD\_ADM-R.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD\_ADM-R.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-R.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-R.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-R.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-R.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-R.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD\_ADM-R.1.1E The evaluator shall confirm that this guidance contains no errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### 4.5.14 AGD\_USR-R.1 User guidance

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

AGD\_USR-R.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_USR-R.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-R.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-R.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-R.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-R.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-R.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD\_USR-R.1.1E The evaluator shall confirm that this guidance contains no errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.5.15 ALC\_DVS-R.1 Identification of security measures**

Dependencies: No dependencies.

Developer action elements:

ALC\_DVS-R.1.1D The developer shall identify the physical, procedural, personnel, and other security measures that are deemed necessary by the developer to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-R.1.2D(+) The developer shall implement the measures identified.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_DVS-R.1.1E The evaluator shall confirm that the developer has identified the physical, procedural, personnel, and other security measures that are deemed necessary by the developer to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-R.1.2E The evaluator shall determine that the security measures are being applied.

#### **4.5.16 ALC\_FLR-R.2 Flaw reporting procedures**

Dependencies: No dependencies.

Developer action elements:

ALC\_FLR-R.2.1D The developer shall establish flaw remediation procedures that provide the capabilities included in the content and presentation section below.

ALC\_FLR-R.2.2D The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

**Content and presentation of evidence elements:**

ALC\_FLR-R.2.1C The flaw remediation procedures shall include tracking of all reported security flaws in each release of the TOE.

ALC\_FLR-R.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC\_FLR-R.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC\_FLR-R.2.4C The flaw remediation procedures shall include providing flaw information, corrections and guidance on corrective actions to TOE users.

ALC\_FLR-R.2.5C The procedures for processing reported security flaws shall include measures to ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC\_FLR-R.2.6C The procedures for processing reported security flaws shall include safeguards to reduce the potential for corrections to these security flaws to introduce new flaws.

**Evaluator action elements:**

ALC\_FLR-R.2.1E The evaluator shall confirm that the flaw remediation procedures meet all the requirements identified in the content and presentation of evidence section above.

ALC\_FLR-R.2.2E(+) The evaluator shall confirm that the flaw remediation procedures include provisions for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

**4.5.17 ALC\_LCD-R.1 Developer defined life-cycle model**

Dependencies: No dependencies.

**Developer action elements:**

ALC\_LCD-R.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ALC\_LCD-R.1.1E The evaluator shall determine that the developer has established a life-cycle model to be used in the development and maintenance of the TOE.

#### 4.5.18 ALC\_TAT-R.2 Compliance with implementation standards

Dependencies: ADV\_IMP-R.1 Subset of the implementation of the TSF

Developer action elements:

ALC\_TAT-R.2.1D The developer shall use well-defined development tools for the TOE.

ALC\_TAT-R.2.2D The developer shall identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-R.2.3D The developer shall apply identified implementation standards.

ALC\_TAT-R.2.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_TAT-R.2.1E The evaluator shall confirm that the developer is using well-defined development tools for the TOE.

ALC\_TAT-R.2.2E The evaluator shall confirm that the identified implementation standards have been applied.

ALC\_TAT-R.2.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-R.2.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### 4.5.19 ATE\_COV-R.2 Analysis of coverage

Objectives: In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved through an examination of analysis of correspondence.

Application notes: The developer is required to have conducted testing of all of the security functions as described in the functional specification on the basis of an analysis that shows the correspondence between tests and security functions and also provides sufficient

information for the evaluator to determine how the functions have been exercised. This information can be used in planning for additional evaluator tests. Although at this level the developer has to demonstrate that each of the functions within the functional specification has been tested, the amount of testing of each function need not be exhaustive.

Dependencies:

ADV\_FSP-R.1 Informal functional specification  
ATE\_FUN-R.1 Functional testing

Developer action elements:

ATE\_COV-R.2.1D The developer shall conduct testing on the basis of an analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

Content and presentation of evidence elements:

None

Evaluator action elements:

ATE\_COV-R.2.1E The evaluator shall confirm that the developer testing was performed on the basis of an analysis of test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

#### **4.5.20 ATE\_DPT-R.2 Testing: low-level design**

**Objectives:** The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized. The modules of a TSF provide a description of the internal workings of the TSF. Testing at the level of the modules, in order to demonstrate the presence of any flaws, provides assurance that the TSF modules have been correctly realized.

**Application notes:** The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts. The developer is expected to describe the testing of the low-level design of the TSF in terms of "modules". The term "modules" is used to express the notion of decomposing each of the "subsystems" of the TSF into a relatively small number of parts.

Dependencies:

ADV\_HLD-R.1 Descriptive high-level design  
ADV\_LLD-R.1 Descriptive low-level design  
ATE\_FUN-R.1 Functional testing

**Developer action elements:**

ATE\_DPT-R.2.1D The developer shall conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and its low-level design.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_DPT-R.2.1E The evaluator shall confirm that developer testing was conducted conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and its low-level design.

#### **4.5.21 ATE\_FUN-R.1 Functional testing**

**Objectives:** The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation.

**Dependencies:** No dependencies.

**Developer action elements:**

ATE\_FUN-R.1.1D The developer shall test the TSF.

ATE\_FUN-R.1.2D The developer shall produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

**Content and presentation of evidence elements:**

ATE\_FUN-R.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-R.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-R.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.



ATE\_FUN-R.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-R.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

**Evaluator action elements:**

ATE\_FUN-R.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the that the developer tested the TSF.

ATE\_FUN-R.1.2E(+) *placeholder for element appearing at higher component*

ATE\_FUN-R.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.

#### **4.5.22 ATE\_IND-R.2 Independent testing - sample**

**Objectives:** The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes selecting and repeating a sample of the developer tests.

**Application notes:** The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. This component contains a requirement that the evaluator has available test results from the developer to supplement the program of testing. The evaluator will repeat a sample of the developer's tests to gain confidence in the results obtained. Having established such confidence the evaluator will build upon the developer's testing by conducting additional tests that exercise the TOE in a different manner. By using a platform of validated developer test results the evaluator is able to gain confidence that the TOE operates correctly in a wider range of conditions than would be possible purely using the developer's own efforts, given a fixed level of resource. Having gained confidence that the developer has tested the TOE, the evaluator will also have more freedom, where appropriate, to concentrate testing in areas where examination of documentation or specialist knowledge has raised particular concerns.

**Dependencies:**

ADV\_FSP-R.1 Informal functional specification  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance  
ATE\_FUN-R.1 Functional testing

**Developer action elements:**

ATE\_IND-R.2.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Content and presentation of evidence elements:

None

Evaluator action elements:

ATE\_IND-R.2.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE\_IND-R.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE meets all functional requirements in the associated security target.

ATE\_IND-R.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

#### **4.5.23 AVA\_CCA-R.1 Covert channel analysis**

Objectives: The objective is to identify covert channels that are identifiable, through an informal search for covert channels.

Dependencies:

ADV\_FSP-R.2 Fully defined external interfaces

ADV\_IMP-R.2 Implementation of the TSF

AGD\_ADM-R.1 Administrator guidance

AGD\_USR-R.1 User guidance

Developer action elements:

AVA\_CCA-R.1.1D The developer shall conduct a search for covert channels for each information flow control policy; identifying each covert channel, estimating its capacity, and determining the worst-case exploitation scenario.

AVA\_CCA-R.1.2D The developer shall provide covert channel analysis documentation containing the information identified in the content and presentation section below.

Content and presentation of evidence elements:

AVA\_CCA-R.1.1C The analysis documentation shall identify covert channels and estimate their capacity.

AVA\_CCA-R.1.2C The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.

AVA\_CCA-R.1.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.

AVA\_CCA-R.1.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.

AVA\_CCA-R.1.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

**Evaluator action elements:**

AVA\_CCA-R.1.1E The evaluator shall confirm that the developer has identified covert channels and for each channel estimated its capacity, and determined the worst-case exploitation scenario.

AVA\_CCA-R.1.2E The evaluator shall confirm that the results of the covert channel analysis show that the TOE meets its functional requirements.

AVA\_CCA-R.1.3E The evaluator shall selectively validate the covert channel analysis through testing.

#### **4.5.24 AVA\_MSU-R.2 Validation of analysis**

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met.

**Dependencies:**

ADO\_IGS-R.1 Installation, generation, and start-up procedures

ADV\_FSP-R.1 Informal functional specification

AGD\_ADM-R.1 Administrator guidance

AGD\_USR-R.1 User guidance

**Developer action elements:**

AVA\_MSU-R.2.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-R.2.2D The developer shall analyze the guidance documentation to determine that the guidance documentation is complete.

**Content and presentation of evidence elements:**

AVA\_MSU-R.2.1C The guidance documentation shall identify possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-R.2.2C *CC element incorporated into developer elements*

AVA\_MSU-R.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA\_MSU-R.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

**Evaluator action elements:**

AVA\_MSU-R.2.1E The evaluator shall confirm that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-R.2.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-R.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA\_MSU-R.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

**4.5.25 AVA\_SOF-R.1 Strength of TOE security function evaluation****Dependencies:**

ADV\_FSP-R.1 Informal functional specification

ADV\_HLD-R.1 Descriptive high-level design

**Developer action elements:**

AVA\_SOF-R.1.1D The developer shall perform a strength of TOE security function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

AVA\_SOF-R.1.1E The evaluator shall confirm that the developer's strength of function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

AVA\_SOF-R.1.2E The evaluator shall determine that the strength claims are correct.

**4.5.26 AVA\_VLA-R.3 Moderately resistant**

Objectives. A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks using publicly known attacks and otherwise obvious attack methods and by attackers possessing a moderate attack potential.

**Dependencies:**

ADV\_FSP-R.1 Informal functional specification  
ADV\_HLD-R.2 Security enforcing high-level design  
ADV\_IMP-R.1 Subset of the implementation of the TSF  
ADV\_INT-R.1 Modularity  
ADV\_LLD-R.1 Descriptive low-level design  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance  
FPT\_RVM.1 Non-Bypassability of the TSP  
FPT\_SEP.2 SFP domain separation

**Developer action elements:**

AVA\_VLA-R.3.1D The developer shall perform and document a systematic analysis of the TOE searching for ways in which a user possessing a moderate attack potential can violate the TSP.

AVA\_VLA-R.3.2D The developer shall conduct penetration testing based upon this analysis to determine that the TOE is resistant to penetration attacks by attackers possessing a moderate attack potential.

AVA\_VLA-R.3.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

**Content and presentation of evidence elements:**

AVA\_VLA-R.3.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA\_VLA-R.3.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AL5- AVA\_VLA-R.3.3C The evidence shall show that the search for vulnerabilities is systematic.

**Evaluator action elements:**

AVA\_VLA-R.3.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing.

AVA\_VLA-R.3.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA\_VLA-R.3.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods and with respect to attackers with moderate attack capability.

AVA\_VLA-R.3.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods and to attackers with moderate attack capability.

AVA\_VLA-R.3.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods and by an attacker possessing a moderate attack potential.

AVA\_VLA-R.3.6E(+) The evaluator shall determine that the developer has performed penetration testing on the basis of a systematic vulnerability analysis to determine that the TOE is resistant to penetration attacks by attackers possessing moderate attack capability.

## 4.6 AL6 – RIGOROUS SECURITY ENGINEERING

### 4.6.1 ACM\_AUT-L.2 Complete CM automation

**Objectives:** In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorized. It is the objective of this component to ensure that the implementation representation is controlled through automated means. Providing an automated means of ascertaining changes between versions of the TOE and identifying which configuration items are affected by modifications to other configuration items assists in determining the impact of the changes between successive versions of the TOE. This in turn can provide valuable information in determining whether changes to the TOE result in all configuration items being consistent with one another.

**Dependencies:** ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ACM\_AUT-L.2.1D The developer shall use a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

ACM\_AUT-L.2.2D *placeholder for element appearing in higher component*

ACM\_AUT-L.2.3D(+) The developer shall use a CM system that provides automated means to support the generation of the TOE.

ACM\_AUT-L.2.4D(+) The developer shall use a CM system that provides automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-L.2.5D(+) The developer shall use a CM system that provides automated means to identify all other configuration items that are affected by the modification of a given configuration item.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_AUT-L.2.1E The evaluator shall check that the developer is using a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

ACM\_AUT-L.2.2E The evaluator shall confirm, at the level of appears to be true, that the developer is using a CM system that includes automated means to support the generation of the TOE.

ACM\_AUT-L.2.3E(+) The evaluator shall confirm, at the level of appears to be true, that the developer is using a CM system that includes automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-L.2.4E(+) The evaluator shall confirm, at the level of appears to be true, that the developer is using a CM system that includes automated means to identify all other configuration items that are affected by the modification of a given configuration item.

#### **4.6.2 ACM\_CAP-L.5 Generation support and acceptance procedures**

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE. The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized. Integration procedures help to ensure that generation of the TOE from a managed set of configuration items is correctly performed in an authorized manner. Requiring that the CM system be able to identify the master copy of the material used to generate the TOE helps to ensure that the integrity of this material is preserved by the appropriate technical, physical and procedural safeguards.

**Dependencies:**

ACM\_SCP-L.1 TOE CM coverage  
ALC\_DVS-L.2 Sufficiency of security measures

**Developer action elements:**

ACM\_CAP-L.5.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-L.5.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-L.5.3D The developer shall provide a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.5.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.



ACM\_CAP-L.5.5D The developer shall use a CM system that supports the generation of the TOE.

ACM\_CAP-L.5.6D The developer shall use a CM system that includes procedures used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-L.5.7D The developer shall use a CM system that applies to the TOE manufacturing process.

ACM\_CAP-L.5.8D The developer shall use a CM system that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-L.5.9D The developer shall use a CM system that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-L.5.10D The developer shall use a CM system that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-L.5.11D The developer shall use a CM system that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-L.5.12D The developer shall use a CM system that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-L.5.13D The developer shall use a CM system that provides for an adequate and appropriate review of changes to all configuration items.

Content and presentation of evidence elements:

None

Evaluator action elements:

ACM\_CAP-L.5.1E The evaluator shall check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-L.5.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.5.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to uniquely identify all configuration items.

ACM\_CAP-L.5.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to provide measures such that only authorized changes are made to the configuration items.

ACM\_CAP-L5.5E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to support generation of the TOE.

ACM\_CAP-L5.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-L5.7E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that applies to the TOE manufacturing process.

ACM\_CAP-L5.8E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-L5.9E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-L5.10E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-L5.11E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-L5.12E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-L5.13E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that provides for an adequate and appropriate review of changes to all configuration items.

#### **4.6.3 ACM\_SCP-L.3 Development tools CM coverage**

**Objectives:** A CM system can control changes only to those items that have been placed under CM. Placing the TOE implementation representation, design, tests, user and administrator documentation, and CM documentation under CM provides assurance that they have been modified in a controlled manner with proper authorizations. The ability to track security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution. Development tools play an important role in ensuring the production of a quality version of the TOE. Therefore, it is important to control modifications to these tools.

**Dependencies:** ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ACM\_SCP-L.3.1D The developer shall perform CM such that, as a minimum, the following are kept under CM: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_SCP-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the following are being configuration managed: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

**4.6.4 ADO\_DEL-L.2 Detection of modification**

Dependencies: ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ADO\_DEL-L.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures necessary to maintain security when distributing versions of the TOE to a user's site.

ADO\_DEL-L.2.2D *CC element deleted*

ADO\_DEL-L.2.3D(+) The developer shall explain how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-L.2.4D(+) The developer shall explain how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADO\_DEL-L.2.1E The evaluator shall check that the documented delivery procedures describe the procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.

ADO\_DEL-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has explained how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-L.2.3E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has explained how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

#### **4.6.5 ADO\_IGS-L.1 Installation generation and start-up procedures**

Dependencies: AGD\_ADM-L.1 Administrator guidance

Developer action elements:

ADO\_IGS-L.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_IGS-L.1.1E The evaluator shall check that the documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

#### **4.6.6 ADV\_FSP-L.2 Fully defined external interfaces**

Dependencies: ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:

ADV\_FSP-L.2.1D The developer shall provide an informal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-L.2.2D(+) The developer shall show that the TSF is a completely and correctly represented by this functional specification.

#### Content and presentation of evidence elements:

ADV\_FSP-L.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV\_FSP-L.2.2C The functional specification shall be internally consistent.

ADV\_FSP-L.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

#### Evaluator action elements:

ADV\_FSP-L.2.1E The evaluator shall confirm, to the level of rigor of no obvious errors or omissions and at a level of completeness sufficient to support effective functional testing against the functional requirements in the associated security target that, the functional specification meets all requirements for content and presentation of evidence.

ADV\_FSP-L.2.2E The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has shown that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

#### **4.6.7 ADV\_HLD-L.4 Semiformal or Informal high-level explanation**

##### Dependencies:

ADV\_FSP-L.1 Informal functional specification

ADV\_RCR-L.1 Informal correspondence demonstration

##### Developer action elements:

ADV\_HLD-L.4.1D The developer shall produce a high-level design of the TSF that provides the information identified in the content and presentation section below.

ADV\_HLD-L.4.2D(+) The developer shall, as an essential part of the TOE development process, produce the high-level design as a precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.4.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

#### Content and presentation of evidence elements:

ADV\_HLD-L.4.1C The presentation of the high-level design shall be semiformal or informal.

ADV\_HLD-L.4.2C The high-level design shall be internally consistent.

ADV\_HLD-L.4.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-L.4.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-L.4.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-L.4.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-L.4.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV\_HLD-L.4.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_HLD-L.4.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

ADV\_HLD-L.4.10C The high-level design shall justify that the identified means of achieving separation, including any protection mechanisms, are sufficient to ensure a clear and effective separation of TSP-enforcing from non-TSP-enforcing functions.

ADV\_HLD-L.4.11C The high-level design shall justify that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design.

Evaluator action elements:

ADV\_HLD-L.4.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-L.4.2E *placeholder for element appearing at higher component* ADV\_HLD-L.4.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.4.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process maintains the high-level design to reflect the actual implementation.

#### 4.6.8 ADV\_IMP-L.3 Structured Implementation of the TSF

Application notes

Dependencies:

ADV\_LLD-L.1 Descriptive low-level design

ADV\_RCR-L.1 Informal correspondence demonstration

ALC\_TAT-L.1 Well-defined development tools

Developer action elements:

ADV\_IMP-L.3.1D The developer shall produce an implementation representation for the entire TSF that unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-L.3.2D(+) The developer shall produce the implementation representation for the entire TSF such that this representation is structured into small and comprehensible sections.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_IMP-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the implementation representation unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-L.2.2E *placeholder for element appearing at higher component*

ADV\_IMP-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the implementation representation is structured into small and comprehensible sections.

#### 4.6.9 ADV\_INT-L.3 Minimization of complexity

Application notes: This component requires that the reference monitor property "simple enough to be analyzed" is fully addressed. When this component is combined with the functional requirements FPT\_RVM.1 and FPT\_SEP.3, the reference monitor concept would be fully realized.

Dependencies:

ADV\_IMP-L.2 Implementation of the TSF

ADV\_LLD-L.1 Descriptive low-level design

**Developer action elements:**

ADV\_INT-L.3.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

ADV\_INT-L.3.2D *CC element deleted*

ADV\_INT-L.3.3D The developer shall design and structure the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-L.3.4D The developer shall design and structure the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-L.3.5D The developer shall design and structure the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-L.3.6D The developer shall ensure that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADV\_INT-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed, structured, and implemented the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

ADV\_INT-L.3.2E *Intent of CC element incorporated into \*.1E above*

ADV\_INT-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed and structured the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-L.3.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed and structured the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-L.3.5E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed and structured the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-L.3.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer ensured that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.



#### 4.6.10 ADV\_LLD-L.1 Descriptive low-level design

Dependencies:

ADV\_HLD-L.2 Security enforcing high-level design  
ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:

ADV\_LLD-L.1.1D The developer shall provide the low-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_LLD-L.1.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a low-level design.

ADV\_LLD-L.1.3D(+) The developer shall, as an essential part of the TOE development process, maintain the low-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_LLD-L.1.1C The presentation of the low-level design shall be informal.

ADV\_LLD-L.1.2C The low-level design shall be internally consistent.

ADV\_LLD-L.1.3C The low-level design shall describe the TSF in terms of modules.

ADV\_LLD-L.1.4C The low-level design shall describe the purpose of each module.

ADV\_LLD-L.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV\_LLD-L.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV\_LLD-L.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV\_LLD-L.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV\_LLD-L.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_LLD-L.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

ADV\_LLD-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the documented low-level design meets all requirements for content and presentation of evidence.

ADV\_LLD-L.1.2E *CC element deleted* ADV\_LLD-L.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process produces the low-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_LLD-L.1.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process maintains the low-level design to reflect the actual implementation.

#### **4.6.11 ADV\_RCR-L.1 Informal correspondence demonstration**

Dependencies: No dependencies.

Developer action elements:

ADV\_RCR-L.1.1D The developer shall conduct an informal analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_RCR-L.1.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the results of the developer analysis of correspondence show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

#### **4.6.12 ADV\_SPM-L.1 Informal TOE security policy model**

Dependencies: ADV\_FSP-L.1 Informal functional specification

Developer action elements:

ADV\_SPM-L.1.1D The developer shall produce an informal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.

ADV\_SPM-L.1.2D The developer shall show correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_SPM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the SPM is an informal TSP model that describes the rules and characteristics of the policies of the TSP.

ADV\_SPM-L.1.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has shown correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

#### **4.6.13 AGD\_ADM-L.1 Administrator guidance**

Dependencies: None

Developer action elements:

AGD\_ADM-L.1.1D The developer shall provide administrator guidance addressed to system Administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_ADM-L.1.1C The administrator guidance shall describe the Administrative functions and interfaces available to the administrator of the TOE.

AGD\_ADM-L.1.2C The administrator guidance shall describe how to Administer the TOE in a secure manner.

AGD\_ADM-L.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-L.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-L.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-L.1.6C The administrator guidance shall describe each type of security-relevant event relative to the Administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-L.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-L.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD\_ADM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.6.14 AGD\_USR-L.1 User guidance**

Dependencies: None

Developer action elements:

AGD\_USR-L.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_USR-L.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-L.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-L.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-L.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-L.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-L.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD\_USR-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.6.15 ALC\_DVS-L.2 Sufficiency of security measures**

Dependencies: No dependencies.

Developer action elements:

ALC\_DVS-L.2.1D The developer shall identify the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-L.2.2D(+) The developer shall implement the measures identified.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_DVS-L.2.1E The evaluator shall confirm that the developer has identified the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-L.2.2E The evaluator shall confirm, to the level of rigor of appears to be true, that the security measures are being applied.

#### 4.6.16 ALC\_FLR-L.2 Flaw reporting procedures

Dependencies: No dependencies.

Developer action elements:

ALC\_FLR-L.2.1D The developer shall establish flaw remediation procedures that provide the capabilities included in the content and presentation section below.

ALC\_FLR-L.2.2D The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

Content and presentation of evidence elements:

ALC\_FLR-L.2.1C The flaw remediation procedures shall include tracking of all reported security flaws in each release of the TOE.

ALC\_FLR-L.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC\_FLR-L.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC\_FLR-L.2.4C The flaw remediation procedures shall include providing flaw information, corrections and guidance on corrective actions to TOE users.

ALC\_FLR-L.2.5C The procedures for processing reported security flaws shall include measures to ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC\_FLR-L.2.6C The procedures for processing reported security flaws shall include safeguards to reduce the potential for corrections to these security flaws to introduce new flaws.

Evaluator action elements:

ALC\_FLR-L.2.1E The evaluator shall confirm, to the level of rigor of appears true, that the flaw remediation procedures meet all the requirements identified in the content and presentation of evidence section above.

ALC\_FLR-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears true, that the flaw remediation procedures include provisions for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

#### 4.6.17 ALC\_LCD-L.2 Standardized life-cycle model

Dependencies: No dependencies.

Developer action elements:

ALC\_LCD-L.2.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC\_LCD-L.2.2D *CC element deleted*

ALC\_LCD-L.2.3D The developer shall use a standardized life-cycle model to develop and maintain the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_LCD-L.2.1E The evaluator shall confirm, to the level or rigor of appears to be true, that the developer is using a standardized life-cycle model for the development and maintenance of the TOE.

#### 4.6.18 ALC\_TAT-L.3 Compliance with implementation standards - all parts

Dependencies: ADV\_IMP-L.1 Subset of the implementation of the TSF

Developer action elements:

ALC\_TAT-L.3.1D The developer shall use well-defined development tools for the TOE.

ALC\_TAT-L.3.2D The developer shall identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.3.3D The developer shall apply identified implementation standards for all parts of the TOE as appropriate.

ALC\_TAT-L.3.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_TAT-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using well-defined development tools for the TOE.

ALC\_TAT-L.3.2E The evaluator shall confirm, to the level of rigor of appears to be true, that the identified implementation standards have been applied, as appropriate, to all parts of the TOE.

ALC\_TAT-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.3.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### **4.6.19 ATE\_COV-L.3 Analysis of coverage**

**Objectives:** In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved by the developer conducting testing on the basis of a rigorous analysis of correspondence.

**Application notes:**

**Dependencies:**

ADV\_FSP-L.1 Informal functional specification

ATE\_FUN-L.1 Functional testing

**Developer action elements:**

ATE\_COV-L.3.1D The developer shall conduct testing on the basis of an analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

ATE\_COV-L.3.2D(+) The developer shall conduct testing on the basis of a rigorous analysis of the test coverage that was used to ensure that the tests conducted completely tested all internal interfaces of the TSF identified in the functional specification.

**Content and presentation of evidence elements:**

None



**Evaluator action elements:**

ATE\_COV-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer testing was performed on the basis of an analysis of test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

ATE\_COV-L.3.2E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer testing was performed on the basis of a rigorous analysis of test coverage that was used to ensure that all external interfaces of the TSF identified in the functional specification have been completely tested.

**4.6.20 ATE\_DPT-L.3 Testing: implementation representation**

**Objectives:** The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized. The modules of a TSF provide a description of the internal workings of the TSF. Testing at the level of the modules, in order to demonstrate the presence of any flaws, provides assurance that the TSF modules have been correctly realized. The implementation representation of a TSF provides a detailed description of the internal workings of the TSF. Testing at the level of the implementation, in order to demonstrate the presence of any flaws, provides assurance that the TSF implementation has been correctly realized.

**Application notes:** The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts. The developer is expected to describe the testing of the low-level design of the TSF in terms of "modules". The term "modules" is used to express the notion of decomposing each of the "subsystems" of the TSF into a relatively small number of parts. The implementation representation is the one which is used to generate the TSF itself (e.g. source code which is then compiled).

**Dependencies:**

ADV\_HLD-L.2 Security enforcing high-level design  
ADV\_IMP-L.2 Implementation of the TSF  
ADV\_LLD-L.1 Descriptive low-level design  
ATE\_FUN-L.1 Functional testing

**Developer action elements:**

ATE\_DPT-L.3.1D The developer shall conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ATE\_DPT-L.3.1E The evaluator shall confirm that developer testing was conducted on the basis of an analysis of the depth of testing that demonstrates, to the level of rigor of appears reasonable, that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation.

#### **4.6.21 ATE\_FUN-L.2 Ordered functional testing**

**Objectives:** The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation. In this component, an additional objective is to ensure that testing is structured such as to avoid circular arguments about the correctness of the portions of the TSF being tested.

**Application notes:** Although the test procedures may state pre-requisite initial test conditions in terms of ordering of tests, they may not provide a rationale for the ordering. An analysis of test ordering is an important factor in determining the adequacy of testing, as there is a possibility of faults being concealed by the ordering of tests.

**Dependencies:** No dependencies.

Developer action elements:

ATE\_FUN-L.2.1D The developer shall test the TSF.

ATE\_FUN-L.2.2D The developer shall produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

ATE\_FUN-L.2.3D(+) The develop shall conduct testing on the basis of an analysis of the test procedure ordering dependencies.

Content and presentation of evidence elements:

ATE\_FUN-L.2.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-L.2.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-L.2.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE\_FUN-L.2.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-L.2.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE\_FUN-L.2.6C The test documentation shall include an analysis of the test procedure ordering dependencies.

**Evaluator action elements:**

ATE\_FUN-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the that the developer tested the TSF.

ATE\_FUN-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer conducted testing on the basis of an analysis of the test procedure ordering dependencies.

ATE\_FUN-L.2.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.

#### **4.6.22 ATE\_IND-L.2 Independent testing - sample**

**Objectives:** The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes selecting and repeating a sample of the developer tests.

**Application notes:** The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. This component contains a requirement that the evaluator has available test results from the developer to supplement the program of testing. The evaluator will repeat a sample of the developer's tests to gain confidence in the results obtained. Having established such confidence the evaluator will build upon the developer's testing, as necessary to confirm the developer testing, by conducting additional tests that exercise the TOE in a different manner. By using a platform of validated developer test results the evaluator is able to gain confidence that the TOE operates correctly in a wider range of conditions than would be possible purely using the developer's own efforts, given a fixed level of resource. Having gained confidence that the developer has tested the TOE, the evaluator will also have more freedom, where appropriate, to concentrate testing in areas where examination of documentation or specialist knowledge has raised particular concerns.

**Dependencies:**

ADV\_FSP-L.1 Informal functional specification  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance  
ATE\_FUN-L.1 Functional testing

**Developer action elements:**

ATE\_IND-L.2.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_IND-L.2.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE\_IND-L.2.2E The evaluator shall test the TSF only as necessary to confirm that the developer testing shows that the TOE meets all functional requirements in the associated security target.

ATE\_IND-L.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

#### **4.6.23 AVA\_CCA-L.2 Systematic covert channel analysis**

**Objectives:** The objective is to identify covert channels that are identifiable, through an informal search for covert channels.

**Application notes:** Performing a covert channel analysis in a systematic way requires that the developer identify covert channels in a structured and repeatable way, as opposed to identifying covert channels in an ad-hoc fashion.

**Dependencies:**

ADV\_FSP-L.2 Fully defined external interfaces  
ADV\_IMP-L.2 Implementation of the TSF  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

**Developer action elements:**

AVA\_CCA-L.2.1D The developer shall conduct a systematic search for covert channels for each information flow control policy; identifying each covert channel, estimating its capacity, and determining the worst-case exploitation scenario.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

AVA\_CCA-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has systematically identified covert channels and for each channel estimated its capacity, and determined the worst-case exploitation scenario.

**4.6.24 AVA\_MSU-L.3 Analysis and testing for insecure states**

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met, and this analysis is validated and confirmed through testing by the developer.

**Application notes:** In this component the developer is required to undertake testing to ensure that if and when the TOE enters an insecure state this may easily be detected. This testing may be considered as a specific aspect of penetration testing.

**Dependencies:**

ADO\_IGS-L.1 Installation, generation, and start-up procedures

ADV\_FSP-L.1 Informal functional specification

AGD\_ADM-L.1 Administrator guidance

AGD\_USR-L.1 User guidance

**Developer action elements:**

AVA\_MSU-L.3.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-L.3.2D The developer shall analyze the guidance documentation to determine that the guidance documentation is complete.

AVA\_MSU-L.3.3D(+) The developer shall perform testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

Content and presentation of evidence elements:

AVA\_MSU-L.3.1C The guidance documentation shall identify possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-L.3.2C *CC element incorporated into developer elements*

AVA\_MSU-L.3.3C The guidance documentation shall list assumptions about the intended environment.

AVA\_MSU-L.3.4C The guidance documentation shall list requirements for external security measures (including external procedural, physical and personnel controls).

Evaluator action elements:

AVA\_MSU-L.3.1E The evaluator shall check that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-L.3.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-L.3.3E The evaluator shall determine, to the level of rigor of appears to be true, that the use of the guidance documentation allows insecure states to be detected.

AVA\_MSU-L.3.4E The evaluator shall confirm, to the level of rigor of appears reasonable, that the analysis documentation shows that guidance is provided for secure operation in the modes of operation of the TOE.

AVA\_MSU-L.3.5E *CC element deleted*

AVA\_MSU-L.3.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer performed testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

#### 4.6.25 AVA\_SOF-L.1 Strength of TOE security function evaluation

Dependencies:

ADV\_FSP-L.1 Informal functional specification

Developer action elements:

AVA\_SOF-L.1.1D The developer shall perform a strength of TOE security function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

Content and presentation of evidence elements:

None

Evaluator action elements:

AVA\_SOF-L.1.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer's strength of function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

#### 4.6.26 AVA\_VLA-L.4 Highly resistant

Objectives. A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks using publicly known attacks and otherwise obvious attack methods. The developer performs penetration testing, supported by a vulnerability analysis, to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential.

Dependencies:

ADV\_FSP-L.1 Informal functional specification  
ADV\_HLD-L.2 Security enforcing high-level design  
ADV\_IMP-L.1 Subset of the implementation of the TSF  
ADV\_LLD-L.1 Descriptive low-level design  
ADV\_INT-L.3 Minimization of complexity  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance  
FPT\_RVM.1 Non-Bypassability of the TSP  
FPT\_SEP.3 Complete reference monitor

Developer action elements:

AVA\_VLA-L.4.1D The developer shall perform a systematic analysis of the TOE that completely addresses TOE deliverables searching for ways in which a user possessing a high attack potential can violate the TSP.

AVA\_VLA-L.4.2D The developer shall conduct penetration testing based upon this analysis to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential.

AVA\_VLA-L.4.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

Content and presentation of evidence elements:

None

Evaluator action elements:

AVA\_VLA-L.4.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing.

AVA\_VLA-L.4.2E *CC element deleted*

AVA\_VLA-L.4.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.4.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.4.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods.

AVA\_VLA-L.4.6E(+) The evaluator shall determine, to the level of rigor of appears reasonable, that the developer has performed penetration testing on the basis of a systematic vulnerability analysis, addressing all TOE deliverables, to determine that the TOE is resistant to penetration attacks by attackers possessing high attack capability.



## 4.7 AL7 – VERIFIED, RIGOROUS SECURITY ENGINEERING

### 4.7.1 ACM\_AUT-R.2 Complete CM automation

**Objectives:** In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorized. It is the objective of this component to ensure that the implementation representation is controlled through automated means. Providing an automated means of ascertaining changes between versions of the TOE and identifying which configuration items are affected by modifications to other configuration items assists in determining the impact of the changes between successive versions of the TOE. This in turn can provide valuable information in determining whether changes to the TOE result in all configuration items being consistent with one another.

**Dependencies:** ACM\_CAP-R.3 Authorization controls

**Developer action elements:**

ACM\_AUT-R.2.1D The developer shall use a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

ACM\_AUT-R.2.2D The developer shall provide a CM plan.

ACM\_AUT-R.2.3D(+) The developer shall use a CM system that provides automated means to support the generation of the TOE.

ACM\_AUT-R.2.4D(+) The developer shall use a CM system that provides automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-R.2.5D(+) The developer shall use a CM system that provides automated means to identify all other configuration items that are affected by the modification of a given configuration item.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_AUT-L.2.1E The evaluator shall confirm that the developer is using a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

ACM\_AUT-L.2.2E The evaluator shall confirm that the developer is using a CM system that includes automated means to support the generation of the TOE.

ACM\_AUT-L.2.3E(+) The evaluator shall confirm that the developer is using a CM system that includes automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-L.2.4E(+) The evaluator shall confirm that the developer is using a CM system that includes automated means to identify all other configuration items that are affected by the modification of a given configuration item.

#### **4.7.2 ACM\_CAP-R.5 Generation support and acceptance procedures**

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE. The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized. Integration procedures help to ensure that generation of the TOE from a managed set of configuration items is correctly performed in an authorized manner. Requiring that the CM system be able to identify the master copy of the material used to generate the TOE helps to ensure that the integrity of this material is preserved by the appropriate technical, physical and procedural safeguards.

**Dependencies:**

ACM\_SCP-R.1 TOE CM coverage  
ALC\_DVS-R.2 Sufficiency of security measures

**Developer action elements:**

ACM\_CAP-R.5.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-R.5.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-R.5.3D The developer shall provide a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-R.5.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

ACM\_CAP-R.5.5D The developer shall use a CM system that supports the generation of the TOE.

ACM\_CAP-R.5.6D The developer shall use a CM system that includes procedures used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-R.5.7D The developer shall use a CM system that applies to the TOE manufacturing process.

ACM\_CAP-R.5.8D The developer shall use a CM system that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-R.5.9D The developer shall use a CM system that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-R.5.10D The developer shall use a CM system that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-R.5.11D The developer shall use a CM system that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-R.5.12D The developer shall use a CM system that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-R.5.13D The developer shall use a CM system that provides for an adequate and appropriate review of changes to all configuration items.

Content and presentation of evidence elements:

None

Evaluator action elements:

ACM\_CAP-R.5.1E The evaluator shall check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-R.5.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-R.5.3E(+) The evaluator shall confirm that the CM system is being used to uniquely identify all configuration items.

ACM\_CAP-R.5.4E(+) The evaluator shall confirm that the CM system is being used to provide measures such that only authorized changes are made to the configuration items.

ACM\_CAP-R5.5E(+) The evaluator shall confirm that the CM system is being used to support generation of the TOE.

ACM\_CAP-R5.6E(+) The evaluator shall confirm that the CM system is being used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-R.5.7E(+) The evaluator shall confirm that the CM system is being used that applies to the TOE manufacturing process.

ACM\_CAP-R.5.8E(+) The evaluator shall confirm that the CM system is being used that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-R.5.9E(+) The evaluator shall confirm that the CM system is being used that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-R.5.10E(+) The evaluator shall confirm that the CM system is being used that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-R.5.11E(+) The evaluator shall confirm that the CM system is being used that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-R.5.12E(+) The evaluator shall determine that the CM system is being used that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-R.5.13E(+) The evaluator shall determine that the CM system is being used that provides for an adequate and appropriate review of changes to all configuration items.

#### **4.7.3 ACM\_SCP-R.3 Development tools CM coverage**

**Objectives:** A CM system can control changes only to those items that have been placed under CM. Placing the TOE implementation representation, design, tests, user and administrator documentation, and CM documentation under CM provides assurance that they have been modified in a controlled manner with proper authorizations. The ability to track security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution. Development tools play an important role in ensuring the production of a quality version of the TOE. Therefore, it is important to control modifications to these tools.

**Dependencies:** ACM\_CAP-R.3 Authorization controls

**Developer action elements:**

ACM\_SCP-R.3.1D The developer shall perform CM such that, as a minimum, the following are kept under CM: the TOE implementation representation, design documentation, test

documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

Content and presentation of evidence elements:

None

Evaluator action elements:

ACM\_SCP-R.3.1E The evaluator shall confirm that the following are being configuration managed: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

#### **4.7.4 ADO\_DEL-R.2 Detection of modification**

Dependencies: ACM\_CAP-R.3 Authorization controls

Developer action elements:

ADO\_DEL-R.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures necessary to maintain security when distributing versions of the TOE to a user's site.

ADO\_DEL-R.2.2D *CC element deleted*

ADO\_DEL-R.2.3D(+) The developer shall explain how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-R.2.4D(+) The developer shall explain how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_DEL-R.2.1E The evaluator shall confirm that the documented delivery procedures describe the procedures necessary to maintain security when distributing versions of the TOE to a user's site.

ADO\_DEL-R.2.2E(+) The evaluator shall confirm that the developer has explained how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-R.2.3E(+) The evaluator shall confirm that the developer has explained how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

#### **4.7.5 ADO\_IGS-R.1 Installation generation and start-up procedures**

Dependencies: AGD\_ADM-R.1 Administrator guidance

Developer action elements:

ADO\_IGS-R.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_IGS-R.1.1E The evaluator shall confirm that the documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

ADO\_IGS-R.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

#### **4.7.6 ADV\_FSP-R.2 Fully defined external interfaces**

Dependencies: ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_FSP-R.2.1D The developer shall provide an informal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-R.2.2D(+) The developer shall show that the TSF is a completely and correctly represented by this functional specification.

Content and presentation of evidence elements:

ADV\_FSP-R.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV\_FSP-R.2.2C The functional specification shall be internally consistent.

ADV\_FSP-R.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

**Evaluator action elements:**

ADV\_FSP-R.2.1E The evaluator shall confirm, to the level of rigor for completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target, that the functional specification meets all requirements for content and presentation of evidence.

ADV\_FSP-R.2.2E The evaluator shall determine that the developer has shown that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

**4.7.7 ADV\_HLD-R.4 Semiformal or Informal high-level explanation**

**Dependencies:**

ADV\_FSP-R.1 Informal functional specification

ADV\_RCR-R.1 Informal correspondence demonstration

**Developer action elements:**

ADV\_HLD-R.4.1D The developer shall provide the high-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_HLD-R.4.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a high-level design.

ADV\_HLD-R.4.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

**Content and presentation of evidence elements:**

ADV\_HLD-R.4.1C The presentation of the high-level design shall be semiformal or informal.

ADV\_HLD-R.4.2C The high-level design shall be internally consistent.

ADV\_HLD-R.4.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-R.4.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-R.4.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-R.4.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-R.4.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV\_HLD-R.4.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_HLD-R.4.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

ADV\_HLD-R.4.10C The high-level design shall justify that the identified means of achieving separation, including any protection mechanisms, are sufficient to ensure a clear and effective separation of TSP-enforcing from non-TSP-enforcing functions.

ADV\_HLD-R.4.11C The high-level design shall justify that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design.

**Evaluator action elements:**

ADV\_HLD-R.4.1E The evaluator shall confirm that the documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-R.4.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_HLD-R.4.3E(+) The evaluator shall confirm that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-R.4.4E(+) The evaluator shall confirm that the developer design process maintains the high-level design to reflect the actual implementation.

#### **4.7.8 ADV\_IMP-R.3 Structured Implementation of the TSF**

Application notes: The ADV\_IMP-R.2.2E element defines a requirement that the evaluator determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements. This provides a direct correspondence between the TOE security functional requirements and the implementation representation, in addition to the pairwise correspondences required by the ADV\_RCR family. It is expected that the evaluator will use the evidence provided in ADV\_RCR as an input to making this determination.



**Dependencies:**

ADV\_LLD-R.1 Descriptive low-level design  
ADV\_RCR-R.1 Informal correspondence demonstration  
ALC\_TAT-R.1 Well-defined development tools

**Developer action elements:**

ADV\_IMP-R.3.1D The developer shall produce an implementation representation for the entire TSF that unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-R.3.2D(+) The developer shall produce the implementation representation for the entire TSF such that this representation is structured into small and comprehensible sections.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADV\_IMP-R.3.1E The evaluator shall confirm that the implementation representation unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-R.3.2E The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_IMP-R.3.3E(+) The evaluator shall confirm that the implementation representation is structured into small and comprehensible sections.

**4.7.9 ADV\_INT-R.3 Minimization of complexity**

**Application notes:** This component requires that the reference monitor property "simple enough to be analyzed" is fully addressed. When this component is combined with the functional requirements FPT\_RVM.1 and FPT\_SEP.3, the reference monitor concept would be fully realized.

**Dependencies:**

ADV\_IMP-R.2 Implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design

**Developer action elements:**

ADV\_INT-R.3.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

ADV\_INT-R.3.2D *CC element deleted*

ADV\_INT-R.3.3D The developer shall design and structure the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-R.3.4D The developer shall design and structure the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-R.3.5D The developer shall design and structure the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-R.3.6D The developer shall ensure that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_INT-R.3.1E The evaluator shall determine that the developer designed and structured, and implemented the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design

ADV\_INT-R.3.2E *Intent of CC element incorporated into \*.1E above*

ADV\_INT-R.3.3E(+) The evaluator shall determine that the developer designed and structured the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-R.3.4E(+) The evaluator shall determine that the developer designed and structured the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-R.3.5E(+) The evaluator shall determine that the developer designed and structured the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-R.3.6E(+) The evaluator shall determine that the developer ensured that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

#### 4.7.10 ADV\_LLD-R.1 Descriptive low-level design

##### Dependencies:

ADV\_HLD-R.2 Security enforcing high-level design  
ADV\_RCR-R.1 Informal correspondence demonstration

##### Developer action elements:

ADV\_LLD-R.1.1D The developer shall provide the low-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_LLD-R.1.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a low-level design.

ADV\_LLD-R.1.3D(+) The developer shall, as an essential part of the TOE development process, maintain the low-level design to reflect the actual implementation.

##### Content and presentation of evidence elements:

ADV\_LLD-R.1.1C The presentation of the low-level design shall be informal.

ADV\_LLD-R.1.2C The low-level design shall be internally consistent.

ADV\_LLD-R.1.3C The low-level design shall describe the TSF in terms of modules.

ADV\_LLD-R.1.4C The low-level design shall describe the purpose of each module.

ADV\_LLD-R.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV\_LLD-R.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV\_LLD-R.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV\_LLD-R.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV\_LLD-R.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_LLD-R.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

##### Evaluator action elements:

ADV\_LLD-R.1.1E The evaluator shall confirm that the documented low-level design meets all requirements for content and presentation of evidence.

ADV\_LLD-R.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_LLD-R.1.3E(+) The evaluator shall confirm that the developer design process produces the low-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_LLD-R.1.4E(+) The evaluator shall confirm that the developer design process maintains the low-level design to reflect the actual implementation.

#### **4.7.11 ADV\_RCR-R.1 Informal correspondence demonstration**

Dependencies: No dependencies.

Developer action elements:

ADV\_RCR-R.1.1D The developer shall conduct an analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_RCR-R.1.1E The evaluator shall confirm that the results of the developer analysis of correspondence show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

#### **4.7.12 ADV\_SPM-R.1 Informal TOE security policy model**

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

ADV\_SPM-R.1.1D The developer shall produce an informal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.

ADV\_SPM-R.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

Content and presentation of evidence elements:

None

**Evaluator action elements:**

ADV\_SPM-R.1.1E The evaluator shall confirm that the SPM is an informal TSP model and describes the rules and characteristics of the policies of the TSP.

ADV\_SPM-R.1.2E(+) The evaluator shall confirm that the developer has demonstrated correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

**4.7.13 AGD\_ADM-R.1 Administrator guidance**

Dependencies: ADV\_FSP-R.1 Informal functional specification

**Developer action elements:**

AGD\_ADM-R.1.1D The developer shall provide administrator guidance addressed to system administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

**Content and presentation of evidence elements:**

AGD\_ADM-R.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD\_ADM-R.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD\_ADM-R.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-R.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-R.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-R.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-R.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-R.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

**Evaluator action elements:**

AGD\_ADM-R.1.1E The evaluator shall confirm that this guidance contains no errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.7.14 AGD\_USR-R.1 User guidance**

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

AGD\_USR-R.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_USR-R.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-R.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-R.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-R.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-R.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-R.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD\_USR-R.1.1E The evaluator shall confirm that this guidance contains no errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### 4.7.15 ALC\_DVS-R.2 Sufficiency of security measures

Dependencies: No dependencies.

Developer action elements:

ALC\_DVS-R.2.1D The developer shall identify the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-R.2.2D(+) The developer shall implement the measures identified.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_DVS-R.2.1E The evaluator shall confirm that the developer has identified the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-R.2.2E The evaluator shall confirm that the security measures are being applied.

#### 4.7.16 ALC\_FLR-R.2 Flaw reporting procedures

Dependencies: No dependencies.

Developer action elements:

ALC\_FLR-R.2.1D The developer shall establish flaw remediation procedures that provide the capabilities included in the content and presentation section below.

ALC\_FLR-R.2.2D The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

Content and presentation of evidence elements:

ALC\_FLR-R.2.1C The flaw remediation procedures shall include tracking of all reported security flaws in each release of the TOE.

ALC\_FLR-R.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC\_FLR-R.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC\_FLR-R.2.4C The flaw remediation procedures shall include providing flaw information, corrections and guidance on corrective actions to TOE users.

ALC\_FLR-R.2.5C The procedures for processing reported security flaws shall include measures to ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC\_FLR-R.2.6C The procedures for processing reported security flaws shall include safeguards to reduce the potential for corrections to these security flaws to introduce new flaws.

Evaluator action elements:

ALC\_FLR-R.2.1E The evaluator shall confirm that the flaw remediation procedures meet all the requirements identified in the content and presentation of evidence section above.

ALC\_FLR-R.2.2E(+) The evaluator shall confirm that the flaw remediation procedures include provisions for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

#### 4.7.17 ALC\_LCD-R.2 Standardized life-cycle model

Dependencies: No dependencies.

Developer action elements:

ALC\_LCD-R.2.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC\_LCD-R.2.2D *CC element deleted*

ALC\_LCD-R.2.3D The developer shall use a standardized life-cycle model to develop and maintain the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_LCD-R.2.1E The evaluator shall confirm that the developer is using a standardized life-cycle model for the development and maintenance of the TOE.



#### 4.7.18 ALC\_TAT-R.3 Compliance with implementation standards - all parts

Dependencies: ADV\_IMP-R.1 Subset of the implementation of the TSF

Developer action elements:

ALC\_TAT-L.3.1D The developer shall use well-defined development tools for the TOE.

ALC\_TAT-L.3.2D The developer shall identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.3.3D The developer shall apply identified implementation standards for all parts of the TOE as appropriate.

ALC\_TAT-L.3.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_TAT-R.3.1E The evaluator shall confirm that the developer is using well-defined development tools for the TOE.

ALC\_TAT-R.3.2E The evaluator shall confirm that the identified implementation standards have been applied, as appropriate, to all parts of the TOE.

ALC\_TAT-R.3.3E(+) The evaluator shall confirm that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-R.3.4E(+) The evaluator shall confirm that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### 4.7.19 ATE\_COV-R.3 Analysis of coverage

Objectives: In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved through an evaluator examination of and by the developer conducting testing on the basis of a rigorous analysis of correspondence.

Application notes: The developer is required to provide a convincing argument that the tests which have been identified cover all security functions, and that the testing of each security function is complete. There will remain little scope for the evaluator to devise additional

functional tests of the TSF interfaces based on the functional specification, as they will have been exhaustively tested. Nevertheless, the evaluator should strive to devise such tests.

**Dependencies:**

ADV\_FSP-R.1 Informal functional specification  
ATE\_FUN-R.1 Functional testing

**Developer action elements:**

ATE\_COV-R.3.1D The developer shall conduct testing on the basis of a documented analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

ATE\_COV-R.3.2D(+) The developer shall conduct testing on the basis of a rigorous analysis of the test coverage that was used to ensure that the tests conducted completely tested all internal interfaces of the TSF identified in the functional specification.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_COV-R.3.1E The evaluator shall confirm that the developer testing was performed on the basis of an analysis of test coverage that ensures that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

ATE\_COV-R.3.2E(+) The evaluator shall confirm that the developer testing was performed on the basis of a rigorous analysis of test coverage that ensures that all external interfaces of the TSF identified in the functional specification have been completely tested.

#### **4.7.20 ATE\_DPT-R.3 Testing: implementation representation**

**Objectives:** The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized. The modules of a TSF provide a description of the internal workings of the TSF. Testing at the level of the modules, in order to demonstrate the presence of any flaws, provides assurance that the TSF modules have been correctly realized. The implementation representation of a TSF provides a detailed description of the internal workings of the TSF. Testing at the level of the implementation, in order to demonstrate the presence of any flaws, provides assurance that the TSF implementation has been correctly realized.

Application notes: The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts. The developer is expected to describe the testing of the low-level design of the TSF in terms of "modules". The term "modules" is used to express the notion of decomposing each of the "subsystems" of the TSF into a relatively small number of parts. The implementation representation is the one which is used to generate the TSF itself (e.g. source code which is then compiled).

Dependencies:

ADV\_HLD-R.2 Security enforcing high-level design  
ADV\_IMP-R.2 Implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design  
ATE\_FUN-R.1 Functional testing

Developer action elements:

ATE\_DPT-R.3.1D The developer shall conduct testing on the basis of a documented analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ATE\_DPT-R.3.1E The evaluator shall confirm that developer testing was conducted on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation.

#### **4.7.21 ATE\_FUN-R.2 Ordered functional testing**

Objectives: The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation. In this component, an additional objective is to ensure that testing is structured such as to avoid circular arguments about the correctness of the portions of the TSF being tested.

Application notes: Although the test procedures may state pre-requisite initial test conditions in terms of ordering of tests, they may not provide a rationale for the ordering. An analysis of test ordering is an important factor in determining the adequacy of testing, as there is a possibility of faults being concealed by the ordering of tests.

Dependencies: No dependencies.

Developer action elements:

ATE\_FUN-R.2.1D The developer shall test the TSF.

ATE\_FUN-R.2.2D The developer shall produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

ATE\_FUN-R.2.3D(+) The developer shall conduct testing on the basis of an analysis of the test procedure ordering dependencies.

Content and presentation of evidence elements:

ATE\_FUN-R.2.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-R.2.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-R.2.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE\_FUN-R.2.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-R.2.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE\_FUN-R.2.6C The test documentation shall include an analysis of the test procedure ordering dependencies.

Evaluator action elements:

ATE\_FUN-R.2.1E The evaluator shall confirm that the developer tested the TSF.

ATE\_FUN-R.2.2E(+) The evaluator shall confirm that the developer conducted testing on the basis of an analysis of the test procedure ordering dependencies.

ATE\_FUN-R.2.3E(+) The evaluator shall confirm that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.

#### **4.7.22 ATE\_IND-R.2 Independent testing - sample**

**Objectives:** The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes selecting and repeating a sample of the developer tests.

**Application notes:** The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. This component contains a requirement that the evaluator has available test results from the developer to supplement the program of testing. The evaluator will repeat a sample of the developer's tests to gain confidence in the results obtained. Having established such confidence the evaluator will build upon the developer's testing by conducting additional tests that exercise the TOE in a different manner. By using a platform of validated developer test results the evaluator is able to gain confidence that the TOE operates correctly in a wider range of conditions than would be possible purely using the developer's own efforts, given a fixed level of resource. Having gained confidence that the developer has tested the TOE, the evaluator will also have more freedom, where appropriate, to concentrate testing in areas where examination of documentation or specialist knowledge has raised particular concerns.

**Dependencies:**

ADV\_FSP-R.1 Informal functional specification  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance  
ATE\_FUN-R.1 Functional testing

**Developer action elements:**

ATE\_IND-R.2.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_IND-R.2.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE\_IND-R.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE meets all functional requirements in the associated security target.

ATE\_IND-R.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

#### 4.7.23 AVA\_CCA-R.2 Systematic covert channel analysis

**Objectives:** The objective is to identify covert channels that are identifiable, through an informal search for covert channels.

**Application notes:** Performing a covert channel analysis in a systematic way requires that the developer identify covert channels in a structured and repeatable way, as opposed to identifying covert channels in an ad-hoc fashion.

**Dependencies:**

ADV\_FSP-R.2 Fully defined external interfaces  
ADV\_IMP-R.2 Implementation of the TSF  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance

**Developer action elements:**

AVA\_CCA-R.2.1D The developer shall conduct a systematic search for covert channels for each information flow control policy; identifying each covert channel, estimating its capacity, and determining the worst-case exploitation scenario.

AVA\_CCA-R.2.2D The developer shall provide covert channel analysis documentation containing the information identified in the content and presentation section below.

**Content and presentation of evidence elements:**

AVA\_CCA-R.2.1C The analysis documentation shall identify covert channels and estimate their capacity.

AVA\_CCA-R.2.2C The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.

AVA\_CCA-R.2.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.

AVA\_CCA-R.2.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.

AVA\_CCA-R.2.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

AVA\_CCA-R.2.6C The analysis documentation shall provide evidence that the method used to identify covert channels is systematic.

**Evaluator action elements:**

AVA\_CCA-R.2.1E The evaluator shall confirm that the developer has systematically identified covert channels and for each channel estimated its capacity, and determined the worst-case exploitation scenario.

AVA\_CCA-R.2.2E The evaluator shall confirm that the results of the covert channel analysis show that the TOE meets its functional requirements.

AVA\_CCA-R.2.3E The evaluator shall selectively validate the covert channel analysis through testing.

**4.7.24 AVA\_MSU-R.3 Analysis and testing for insecure states**

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met, and this analysis is validated and confirmed through testing by the developer and the evaluator.

**Application notes:** In this component the developer and the evaluator are required to undertake testing to ensure that if and when the TOE enters an insecure state this may easily be detected. This testing may be considered as a specific aspect of penetration testing.

**Dependencies:**

ADO\_IGS-R.1 Installation, generation, and start-up procedures

ADV\_FSP-R.1 Informal functional specification

AGD\_ADM-R.1 Administrator guidance

AGD\_USR-R.1 User guidance

**Developer action elements:**

AVA\_MSU-R.3.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-R.3.2D The developer shall analyze the guidance documentation to determine that the guidance documentation is complete.

AVA\_MSU-R.3.3D(+) The developer shall perform testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

**Content and presentation of evidence elements:**

AVA\_MSU-R.3.1C The guidance documentation shall identify possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-R.3.2C *CC element incorporated into developer elements*

AVA\_MSU-R.3.3C The guidance documentation shall list assumptions about the intended environment.

AVA\_MSU-R.3.4C The guidance documentation shall list requirements for external security measures (including external procedural, physical and personnel controls).

Evaluator action elements:

AVA\_MSU-R.3.1E The evaluator shall confirm that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-R.3.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-R.3.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA\_MSU-R.3.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

AVA\_MSU-R.3.5E The evaluator shall perform independent testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

AVA\_MSU-R.3.6E(+) The evaluator shall confirm that the developer performed testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.



#### 4.7.25 AVA\_SOF-R.1 Strength of TOE security function evaluation

Dependencies:

ADV\_FSP-R.1 Informal functional specification  
ADV\_HLD-R.1 Descriptive high-level design

Developer action elements:

AVA\_SOF-R.1.1D The developer shall perform a strength of TOE security function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

Content and presentation of evidence elements:

None

Evaluator action elements:

AVA\_SOF-R.1.1E The evaluator shall confirm that the developer's strength of function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

AVA\_SOF-R.1.2E The evaluator shall determine that the strength claims are correct.

#### 4.7.26 AVA\_VLA-R.4 Highly resistant

Objectives. A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks using publicly known attacks and otherwise obvious attack methods. The developer and the evaluator perform penetration testing, supported by a vulnerability analysis, to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential.

Dependencies:

ADV\_FSP-R.1 Informal functional specification  
ADV\_HLD-R.2 Security enforcing high-level design  
ADV\_IMP-R.1 Subset of the implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design  
ADV\_INT-R.3 Minimization of complexity  
AGD\_ADM-R.1 Administrator guidance

AGD\_USR-R.1 User guidance  
FPT\_RVM.1 Non-Bypassability of the TSP  
FPT\_SEP.3 Complete reference monitor

Developer action elements:

AVA\_VLA-R.4.1D The developer shall perform and document a systematic analysis of the TOE that completely addresses TOE deliverables searching for ways in which a user possessing a high attack potential can violate the TSP.

AVA\_VLA-R.4.2D The developer shall conduct penetration testing based upon this analysis to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential document the disposition of identified vulnerabilities.

AVA\_VLA-R.4.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

Content and presentation of evidence elements:

AVA\_VLA-R.4.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA\_VLA-R.4.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA\_VLA-R.4.3C The evidence shall show that the search for vulnerabilities is systematic.

AVA\_VLA-R.4.4C The analysis documentation shall provide a justification that the analysis completely addresses the TOE deliverables.

Evaluator action elements:

AVA\_VLA-R.4.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing.

AVA\_VLA-R.4.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA\_VLA-R.4.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods and with respect to attackers with high attack capability.

AVA\_VLA-R.4.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods and to attackers with high attack capability.

AVA\_VLA-R.4.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods and by an attacker possessing a high attack potential.

AVA\_VLA-R.4.6E(+) The evaluator shall determine that the developer has performed penetration testing on the basis of a systematic vulnerability analysis to determine that the TOE is resistant to penetration attacks by attackers possessing high attack capability.

DRAFT

## 4.8 AL8 – FORMAL METHODS

### 4.8.1 ACM\_AUT-L.2 Complete CM automation

**Objectives:** In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorized. It is the objective of this component to ensure that the implementation representation is controlled through automated means. Providing an automated means of ascertaining changes between versions of the TOE and identifying which configuration items are affected by modifications to other configuration items assists in determining the impact of the changes between successive versions of the TOE. This in turn can provide valuable information in determining whether changes to the TOE result in all configuration items being consistent with one another.

**Dependencies:** ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ACM\_AUT-L.2.1D The developer shall use a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

ACM\_AUT-L.2.2D *placeholder for element appearing in higher component*

ACM\_AUT-L.2.3D(+) The developer shall use a CM system that provides automated means to support the generation of the TOE.

ACM\_AUT-L.2.4D(+) The developer shall use a CM system that provides automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-L.2.5D(+) The developer shall use a CM system that provides automated means to identify all other configuration items that are affected by the modification of a given configuration item.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_AUT-L.2.1E The evaluator shall check that the developer is using a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

ACM\_AUT-L.2.2E The evaluator shall confirm, at the level of appears to be true, that the developer is using a CM system that includes automated means to support the generation of the TOE.

ACM\_AUT-L.2.3E(+) The evaluator shall confirm, at the level of appears to be true, that the developer is using a CM system that includes automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-L.2.4E(+) The evaluator shall confirm, at the level of appears to be true, that the developer is using a CM system that includes automated means to identify all other configuration items that are affected by the modification of a given configuration item.

#### **4.8.2 ACM\_CAP-L.5 Generation support and acceptance procedures**

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE. The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized. Integration procedures help to ensure that generation of the TOE from a managed set of configuration items is correctly performed in an authorized manner. Requiring that the CM system be able to identify the master copy of the material used to generate the TOE helps to ensure that the integrity of this material is preserved by the appropriate technical, physical and procedural safeguards.

**Dependencies:**

ACM\_SCP-L.1 TOE CM coverage  
ALC\_DVS-L.2 Sufficiency of security measures

**Developer action elements:**

ACM\_CAP-L.5.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-L.5.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-L.5.3D The developer shall provide a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.5.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

ACM\_CAP-L.5.5D The developer shall use a CM system that supports the generation of the TOE.

ACM\_CAP-L.5.6D The developer shall use a CM system that includes procedures used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-L.5.7D The developer shall use a CM system that applies to the TOE manufacturing process.

ACM\_CAP-L.5.8D The developer shall use a CM system that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-L.5.9D The developer shall use a CM system that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-L.5.10D The developer shall use a CM system that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-L.5.11D The developer shall use a CM system that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-L.5.12D The developer shall use a CM system that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-L.5.13D The developer shall use a CM system that provides for an adequate and appropriate review of changes to all configuration items.

Content and presentation of evidence elements:

None

Evaluator action elements:

ACM\_CAP-L.5.1E The evaluator shall check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-L.5.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.5.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to uniquely identify all configuration items.

ACM\_CAP-L.5.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to provide measures such that only authorized changes are made to the configuration items.

ACM\_CAP-L5.5E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to support generation of the TOE.

ACM\_CAP-L5.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-L5.7E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that applies to the TOE manufacturing process.

ACM\_CAP-L5.8E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-L5.9E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-L5.10E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-L5.11E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-L5.12E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-L5.13E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that provides for an adequate and appropriate review of changes to all configuration items.

#### **4.8.3 ACM\_SCP-L.3 Development tools CM coverage**

**Objectives:** A CM system can control changes only to those items that have been placed under CM. Placing the TOE implementation representation, design, tests, user and administrator documentation, and CM documentation under CM provides assurance that they have been modified in a controlled manner with proper authorizations. The ability to track security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution. Development tools play an important role in ensuring the production of a quality version of the TOE. Therefore, it is important to control modifications to these tools.

**Dependencies:** ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ACM\_SCP-L.3.1D The developer shall perform CM such that, as a minimum, the following are kept under CM: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_SCP-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the following are being configuration managed: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

**4.8.4 ADO\_DEL-L.3 Detection of modification**

Dependencies: ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ADO\_DEL-L.3.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures necessary to maintain security when distributing versions of the TOE to a user's site.

**ADO\_DEL-L.3.2D *CC element deleted***

ADO\_DEL-L.3.3D(+) The developer shall explain how the delivery procedures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-L.3.4D(+) The developer shall explain how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADO\_DEL-L.3.1E The evaluator shall check that the documented delivery procedures describe the procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.



ADO\_DEL-L.3.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has explained how the delivery procedures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has explained how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

#### **4.8.5 ADO\_IGS-L.1 Installation generation and start-up procedures**

Dependencies: AGD\_ADM-L.1 Administrator guidance

Developer action elements:

ADO\_IGS-L.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_IGS-L.1.1E The evaluator shall check that the documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

#### **4.8.6 ADV\_FSP-L.4 Formal functional specification**

Dependencies: ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:

ADV\_FSP-L.4.1D The developer shall provide a formal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-L.4.2D(+) The developer shall rigorously justify that the TSF is a completely and correctly represented by this functional specification.

Content and presentation of evidence elements:

ADV\_FSP-L.4.1C The functional specification shall describe the TSF and its external interfaces using an informal style or a semiformal style, supported by informal, explanatory text where appropriate.

ADV\_FSP-L.4.2C The functional specification shall be internally consistent.

ADV\_FSP-L.4.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

**Evaluator action elements:**

ADV\_FSP-L.4.1E The evaluator shall confirm, to the level of rigor of no obvious errors or omissions and at a level of completeness sufficient to support effective functional testing against the functional requirements in the associated security target, that the functional specification meets all requirements for content and presentation of evidence.

ADV\_FSP-L.4.2E The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has rigorously justified that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

#### **4.8.7 ADV\_HLD-L.4 Semiformal or Informal high-level explanation**

**Dependencies:**

ADV\_FSP-L.1 Informal functional specification

ADV\_RCR-L.1 Informal correspondence demonstration

**Developer action elements:**

ADV\_HLD-L.4.1D The developer shall produce a high-level design of the TSF that provides the information identified in the content and presentation section below.

ADV\_HLD-L.4.2D(+) The developer shall, as an essential part of the TOE development process, produce the high-level design as a precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.4.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

**Content and presentation of evidence elements:**

ADV\_HLD-L.4.1C The presentation of the high-level design shall be semiformal or informal.

ADV\_HLD-L.4.2C The high-level design shall be internally consistent.

ADV\_HLD-L.4.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-L.4.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-L.4.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-L.4.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-L.4.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV\_HLD-L.4.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_HLD-L.4.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

ADV\_HLD-L.4.10C The high-level design shall justify that the identified means of achieving separation, including any protection mechanisms, are sufficient to ensure a clear and effective separation of TSP-enforcing from non-TSP-enforcing functions.

ADV\_HLD-L.4.11C The high-level design shall justify that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design.

Evaluator action elements:

ADV\_HLD-L.4.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-L.4.2E *placeholder for element appearing at higher component* ADV\_HLD-L.4.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.4.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process maintains the high-level design to reflect the actual implementation.

#### 4.8.8 ADV\_IMP-L.3 Structured Implementation of the TSF

Application notes

Dependencies:

ADV\_LLD-L.1 Descriptive low-level design

ADV\_RCR-L.1 Informal correspondence demonstration

ALC\_TAT-L.1 Well-defined development tools

Developer action elements:

ADV\_IMP-L.3.1D The developer shall produce an implementation representation for the entire TSF that unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-L.3.2D(+) The developer shall produce the implementation representation for the entire TSF such that this representation is structured into small and comprehensible sections.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_IMP-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the implementation representation unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-L.2.2E *placeholder for element appearing at higher component*

ADV\_IMP-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the implementation representation is structured into small and comprehensible sections.

#### 4.8.9 ADV\_INT-L.3 Minimization of complexity

Application notes: This component requires that the reference monitor property "simple enough to be analyzed" is fully addressed. When this component is combined with the functional requirements FPT\_RVM.1 and FPT\_SEP.3, the reference monitor concept would be fully realized.

Dependencies:

ADV\_IMP-L.2 Implementation of the TSF

ADV\_LLD-L.1 Descriptive low-level design

**Developer action elements:**

ADV\_INT-L.3.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

ADV\_INT-L.3.2D *CC element deleted*

ADV\_INT-L.3.3D The developer shall design and structure the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-L.3.4D The developer shall design and structure the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-L.3.5D The developer shall design and structure the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-L.3.6D The developer shall ensure that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADV\_INT-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed, structured, and implemented the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

ADV\_INT-L.3.2E *Intent of CC element incorporated into \*.1E above*

ADV\_INT-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed and structured the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-L.3.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed and structured the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-L.3.5E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed and structured the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-L.3.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer ensured that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

#### 4.8.10 ADV\_LLD-L.1 Descriptive low-level design

Dependencies:

ADV\_HLD-L.2 Security enforcing high-level design

ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:

ADV\_LLD-L.1.1D The developer shall provide the low-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_LLD-L.1.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a low-level design.

ADV\_LLD-L.1.3D(+) The developer shall, as an essential part of the TOE development process, maintain the low-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_LLD-L.1.1C The presentation of the low-level design shall be informal.

ADV\_LLD-L.1.2C The low-level design shall be internally consistent.

ADV\_LLD-L.1.3C The low-level design shall describe the TSF in terms of modules.

ADV\_LLD-L.1.4C The low-level design shall describe the purpose of each module.

ADV\_LLD-L.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV\_LLD-L.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV\_LLD-L.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV\_LLD-L.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV\_LLD-L.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_LLD-L.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

ADV\_LLD-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the documented low-level design meets all requirements for content and presentation of evidence.

ADV\_LLD-L.1.2E *CC element deleted* ADV\_LLD-L.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process produces the low-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_LLD-L.1.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process maintains the low-level design to reflect the actual implementation.

#### 4.8.11 ADV\_RCR-L.3 Formal correspondence demonstration

Dependencies: No dependencies.

Developer action elements:

ADV\_RCR-L.3.1D The developer shall conduct an analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV\_RCR-L.3.2D(+) The developer shall conduct a formal proof of correspondence between adjacent pair of TSF representations whenever both representations are formally specified.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_RCR-L.3.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the results of the developer analysis of correspondence show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV\_RCR-L.3.2E *CC element deleted*

ADV\_RCR-L.3.3E(+) The evaluator shall check that the developer produced a formal proof of correspondence between adjacent pair of TSF representations whenever both representations are formally specified.

#### **4.8.12 ADV\_SPM-L.3 Formal TOE security policy model**

Dependencies: ADV\_FSP-L.1 Informal functional specification

Developer action elements:

ADV\_SPM-L.3.1D The developer shall produce a formal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.

ADV\_SPM-L.3.2D The developer shall prove correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_SPM-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the SPM a formal TSP model that describes the rules and characteristics of the policies of the TSP.

ADV\_SPM-L.3.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has proven correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

#### **4.8.13 AGD\_ADM-L.1 Administrator guidance**

Dependencies: None

Developer action elements:

AGD\_ADM-L.1.1D The developer shall provide administrator guidance addressed to system Administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_ADM-L.1.1C The administrator guidance shall describe the Administrative functions and interfaces available to the administrator of the TOE.



AGD\_ADM-L.1.2C The administrator guidance shall describe how to Administer the TOE in a secure manner.

AGD\_ADM-L.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-L.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-L.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-L.1.6C The administrator guidance shall describe each type of security-relevant event relative to the Administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-L.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-L.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD\_ADM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.8.14 AGD\_USR-L.1 User guidance**

Dependencies: None

Developer action elements:

AGD\_USR-L.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_USR-L.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-L.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-L.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-L.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-L.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-L.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD\_USR-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.8.15 ALC\_DVS-L.2 Sufficiency of security measures**

Dependencies: No dependencies.

Developer action elements:

ALC\_DVS-L.2.1D The developer shall identify the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-L.2.2D(+) The developer shall implement the measures identified.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_DVS-L.2.1E The evaluator shall confirm that the developer has identified the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-L.2.2E The evaluator shall confirm, to the level of rigor of appears to be true, that the security measures are being applied.

#### 4.8.16 ALC\_FLR-L.2 Flaw reporting procedures

Dependencies: No dependencies.

Developer action elements:

ALC\_FLR-L.2.1D The developer shall establish flaw remediation procedures that provide the capabilities included in the content and presentation section below.

ALC\_FLR-L.2.2D The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

Content and presentation of evidence elements:

ALC\_FLR-L.2.1C The flaw remediation procedures shall include tracking of all reported security flaws in each release of the TOE.

ALC\_FLR-L.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC\_FLR-L.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC\_FLR-L.2.4C The flaw remediation procedures shall include providing flaw information, corrections and guidance on corrective actions to TOE users.

ALC\_FLR-L.2.5C The procedures for processing reported security flaws shall include measures to ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC\_FLR-L.2.6C The procedures for processing reported security flaws shall include safeguards to reduce the potential for corrections to these security flaws to introduce new flaws.

Evaluator action elements:

ALC\_FLR-L.2.1E The evaluator shall confirm, to the level of rigor of appears true, that the flaw remediation procedures meet all the requirements identified in the content and presentation of evidence section above.

ALC\_FLR-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears true, that the flaw remediation procedures include provisions for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

#### 4.8.17 ALC\_LCD-L.3 Measurable life-cycle model

Dependencies: No dependencies.

Developer action elements:

ALC\_LCD-L.3.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC\_LCD-L.3.2D *CC element deleted*

ALC\_LCD-L.3.3D The developer shall use a standardized and measurable life-cycle model to develop and maintain the TOE.

ALC\_LCD-L.3.4D The developer shall measure the TOE development using the standardized and measurable life-cycle model.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_LCD-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using a standardized and measurable life-cycle model for the development and maintenance of the TOE.

ALC\_LCD-L.3.2E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has measured the TOE development using the standardized and measurable life-cycle model.

#### 4.8.18 ALC\_TAT-L.3 Compliance with implementation standards - all parts

Dependencies: ADV\_IMP-L.1 Subset of the implementation of the TSF

Developer action elements:

ALC\_TAT-L.3.1D The developer shall use well-defined development tools for the TOE.

ALC\_TAT-L.3.2D The developer shall identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.3.3D The developer shall apply identified implementation standards for all parts of the TOE as appropriate.

ALC\_TAT-L.3.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### Content and presentation of evidence elements:

None

#### Evaluator action elements:

ALC\_TAT-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using well-defined development tools for the TOE.

ALC\_TAT-L.3.2E The evaluator shall confirm, to the level of rigor of appears to be true, that the identified implementation standards have been applied, as appropriate, to all parts of the TOE.

ALC\_TAT-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.3.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### **4.8.19 ATE\_COV-L.3 Analysis of coverage**

**Objectives:** In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved by the developer conducting testing on the basis of a rigorous analysis of correspondence.

#### Application notes:

#### Dependencies:

ADV\_FSP-L.1 Informal functional specification

ATE\_FUN-L.1 Functional testing

#### Developer action elements:

ATE\_COV-L.3.1D The developer shall conduct testing on the basis of an analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

ATE\_COV-L.3.2D(+) The developer shall conduct testing on the basis of a rigorous analysis of the test coverage that was used to ensure that the tests conducted completely tested all internal interfaces of the TSF identified in the functional specification.

## Content and presentation of evidence elements:

None

## Evaluator action elements:

ATE\_COV-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer testing was performed on the basis of an analysis of test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

ATE\_COV-L.3.2E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer testing was performed on the basis of a rigorous analysis of test coverage that was used to ensure that all external interfaces of the TSF identified in the functional specification have been completely tested.

### 4.8.20 ATE\_DPT-L.3 Testing: implementation representation

**Objectives:** The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized. The modules of a TSF provide a description of the internal workings of the TSF. Testing at the level of the modules, in order to demonstrate the presence of any flaws, provides assurance that the TSF modules have been correctly realized. The implementation representation of a TSF provides a detailed description of the internal workings of the TSF. Testing at the level of the implementation, in order to demonstrate the presence of any flaws, provides assurance that the TSF implementation has been correctly realized.

**Application notes:** The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts. The developer is expected to describe the testing of the low-level design of the TSF in terms of "modules". The term "modules" is used to express the notion of decomposing each of the "subsystems" of the TSF into a relatively small number of parts. The implementation representation is the one which is used to generate the TSF itself (e.g. source code which is then compiled).

## Dependencies:

ADV\_HLD-L.2 Security enforcing high-level design

ADV\_IMP-L.2 Implementation of the TSF

ADV\_LLD-L.1 Descriptive low-level design

ATE\_FUN-L.1 Functional testing

**Developer action elements:**

ATE\_DPT-L.3.1D The developer shall conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_DPT-L.3.1E The evaluator shall confirm that developer testing was conducted on the basis of an analysis of the depth of testing that demonstrates, to the level of rigor of appears reasonable, that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation.

**4.8.21 ATE\_FUN-L.2 Ordered functional testing**

**Objectives:** The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation. In this component, an additional objective is to ensure that testing is structured such as to avoid circular arguments about the correctness of the portions of the TSF being tested.

**Application notes:** Although the test procedures may state pre-requisite initial test conditions in terms of ordering of tests, they may not provide a rationale for the ordering. An analysis of test ordering is an important factor in determining the adequacy of testing, as there is a possibility of faults being concealed by the ordering of tests.

**Dependencies:** No dependencies.

**Developer action elements:**

ATE\_FUN-L.2.1D The developer shall test the TSF.

ATE\_FUN-L.2.2D The developer shall produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

ATE\_FUN-L.2.3D(+) The developer shall conduct testing on the basis of an analysis of the test procedure ordering dependencies.

**Content and presentation of evidence elements:**

ATE\_FUN-L.2.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-L.2.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-L.2.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE\_FUN-L.2.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-L.2.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE\_FUN-L.2.6C The test documentation shall include an analysis of the test procedure ordering dependencies.

**Evaluator action elements:**

ATE\_FUN-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the that the developer tested the TSF.

ATE\_FUN-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer conducted testing on the basis of an analysis of the test procedure ordering dependencies.

ATE\_FUN-L.2.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.

#### **4.8.22 ATE\_IND-L.3 Independent testing - complete**

**Objectives:** The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes repeating all of the developer tests.

**Application notes:** The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. In this component the evaluator must repeat all of the developer's tests as part of the program of testing. As in the previous component the evaluator will also conduct tests that aim to exercise the TOE in a different manner from that achieved by the developer. In cases where developer testing has been exhaustive, there may remain little scope for this.



**Dependencies:**

ADV\_FSP-L.1 Informal functional specification  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance  
ATE\_FUN-L.1 Functional testing

**Developer action elements:**

ATE\_IND-L.3.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_IND-L.3.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE\_IND-L.3.2E The evaluator shall test the TSF only as necessary to confirm that the developer testing shows that the TOE meets all functional requirements in the associated security target..

ATE\_IND-L.3.3E The evaluator shall execute all of tests in the test documentation to verify the developer test results.

**4.8.23 AVA\_CCA-L.2 Systematic covert channel analysis**

**Objectives:** The objective is to identify covert channels that are identifiable, through an informal search for covert channels.

**Application notes:** Performing a covert channel analysis in a systematic way requires that the developer identify covert channels in a structured and repeatable way, as opposed to identifying covert channels in an ad-hoc fashion.

**Dependencies:**

ADV\_FSP-L.2 Fully defined external interfaces  
ADV\_IMP-L.2 Implementation of the TSF  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

**Developer action elements:**

AVA\_CCA-L.2.1D The developer shall conduct a systematic search for covert channels for each information flow control policy; identifying each covert channel, estimating its capacity, and determining the worst-case exploitation scenario.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

AVA\_CCA-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has systematically identified covert channels and for each channel estimated its capacity, and determined the worst-case exploitation scenario.

**4.8.24 AVA\_MSU-L.3 Analysis and testing for insecure states**

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met, and this analysis is validated and confirmed through testing by the developer.

**Application notes:** In this component the developer is required to undertake testing to ensure that if and when the TOE enters an insecure state this may easily be detected. This testing may be considered as a specific aspect of penetration testing.

**Dependencies:**

ADO\_IGS-L.1 Installation, generation, and start-up procedures

ADV\_FSP-L.1 Informal functional specification

AGD\_ADM-L.1 Administrator guidance

AGD\_USR-L.1 User guidance

**Developer action elements:**

AVA\_MSU-L.3.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-L.3.2D The developer shall analyze the guidance documentation to determine that the guidance documentation is complete.

AVA\_MSU-L.3.3D(+) The developer shall perform testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

Content and presentation of evidence elements:

AVA\_MSU-L.3.1C The guidance documentation shall identify possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-L.3.2C *CC element incorporated into developer elements*

AVA\_MSU-L.3.3C The guidance documentation shall list assumptions about the intended environment.

AVA\_MSU-L.3.4C The guidance documentation shall list requirements for external security measures (including external procedural, physical and personnel controls).

Evaluator action elements:

AVA\_MSU-L.3.1E The evaluator shall check that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-L.3.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-L.3.3E The evaluator shall determine, to the level of rigor of appears to be true, that the use of the guidance documentation allows insecure states to be detected.

AVA\_MSU-L.3.4E The evaluator shall confirm, to the level of rigor of appears reasonable, that the analysis documentation shows that guidance is provided for secure operation in the modes of operation of the TOE.

AVA\_MSU-L.3.5E *CC element deleted*

AVA\_MSU-L.3.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer performed testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

#### 4.8.25 AVA\_SOF-L.1 Strength of TOE security function evaluation

Dependencies:

## ADV\_FSP-L.1 Informal functional specification

### Developer action elements:

AVA\_SOF-L.1.1D The developer shall perform a strength of TOE security function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

### Content and presentation of evidence elements:

None

### Evaluator action elements:

AVA\_SOF-L.1.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer's strength of function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

## 4.8.26 AVA\_VLA-L.4 **Highly resistant**

Objectives. A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks using publicly known attacks and otherwise obvious attack methods. The developer performs penetration testing, supported by a vulnerability analysis, to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential.

### Dependencies:

ADV\_FSP-L.1 Informal functional specification  
ADV\_HLD-L.2 Security enforcing high-level design  
ADV\_IMP-L.1 Subset of the implementation of the TSF  
ADV\_LLD-L.1 Descriptive low-level design  
ADV\_INT-L.3 Minimization of complexity  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance  
FPT\_RVM.1 Non-Bypassability of the TSP  
FPT\_SEP.3 Complete reference monitor

Developer action elements:

AVA\_VLA-L.4.1D The developer shall perform a systematic analysis of the TOE that completely addresses TOE deliverables searching for ways in which a user possessing a high attack potential can violate the TSP.

AVA\_VLA-L.4.2D The developer shall conduct penetration testing based upon this analysis to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential.

AVA\_VLA-L.4.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

Content and presentation of evidence elements:

None

Evaluator action elements:

AVA\_VLA-L.4.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing.

AVA\_VLA-L.4.2E *CC element deleted*

AVA\_VLA-L.4.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.4.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.4.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods.

AVA\_VLA-L.4.6E(+) The evaluator shall determine, to the level of rigor of appears reasonable, that the developer has performed penetration testing on the basis of a systematic vulnerability analysis, addressing all TOE deliverables, to determine that the TOE is resistant to penetration attacks by attackers possessing high attack capability.

## 4.9 AL9 – VERIFIED FORMAL METHODS

### 4.9.1 ACM\_AUT-R.2 Complete CM automation

**Objectives:** In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorized. It is the objective of this component to ensure that the implementation representation is controlled through automated means. Providing an automated means of ascertaining changes between versions of the TOE and identifying which configuration items are affected by modifications to other configuration items assists in determining the impact of the changes between successive versions of the TOE. This in turn can provide valuable information in determining whether changes to the TOE result in all configuration items being consistent with one another.

**Dependencies:** ACM\_CAP-R.3 Authorization controls

**Developer action elements:**

ACM\_AUT-R.2.1D The developer shall use a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

ACM\_AUT-R.2.2D The developer shall provide a CM plan.

ACM\_AUT-R.2.3D(+) The developer shall use a CM system that provides automated means to support the generation of the TOE.

ACM\_AUT-R.2.4D(+) The developer shall use a CM system that provides automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-R.2.5D(+) The developer shall use a CM system that provides automated means to identify all other configuration items that are affected by the modification of a given configuration item.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ACM\_AUT-L.2.1E The evaluator shall confirm that the developer is using a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

ACM\_AUT-L.2.2E The evaluator shall confirm that the developer is using a CM system that includes automated means to support the generation of the TOE.

ACM\_AUT-L.2.3E(+) The evaluator shall confirm that the developer is using a CM system that includes automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-L.2.4E(+) The evaluator shall confirm that the developer is using a CM system that includes automated means to identify all other configuration items that are affected by the modification of a given configuration item.

#### **4.9.2 ACM\_CAP-R.5 Generation support and acceptance procedures**

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE. The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized. Integration procedures help to ensure that generation of the TOE from a managed set of configuration items is correctly performed in an authorized manner. Requiring that the CM system be able to identify the master copy of the material used to generate the TOE helps to ensure that the integrity of this material is preserved by the appropriate technical, physical and procedural safeguards.

**Dependencies:**

ACM\_SCP-R.1 TOE CM coverage  
ALC\_DVS-R.2 Sufficiency of security measures

**Developer action elements:**

ACM\_CAP-R.5.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-R.5.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-R.5.3D The developer shall provide a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-R.5.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

ACM\_CAP-R.5.5D The developer shall use a CM system that supports the generation of the TOE.

ACM\_CAP-R.5.6D The developer shall use a CM system that includes procedures used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-R.5.7D The developer shall use a CM system that applies to the TOE manufacturing process.

ACM\_CAP-R.5.8D The developer shall use a CM system that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-R.5.9D The developer shall use a CM system that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-R.5.10D The developer shall use a CM system that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-R.5.11D The developer shall use a CM system that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-R.5.12D The developer shall use a CM system that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-R.5.13D The developer shall use a CM system that provides for an adequate and appropriate review of changes to all configuration items.

Content and presentation of evidence elements:

None

Evaluator action elements:

ACM\_CAP-R.5.1E The evaluator shall check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-R.5.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-R.5.3E(+) The evaluator shall confirm that the CM system is being used to uniquely identify all configuration items.

ACM\_CAP-R.5.4E(+) The evaluator shall confirm that the CM system is being used to provide measures such that only authorized changes are made to the configuration items.



ACM\_CAP-R5.5E(+) The evaluator shall confirm that the CM system is being used to support generation of the TOE.

ACM\_CAP-R5.6E(+) The evaluator shall confirm that the CM system is being used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-R.5.7E(+) The evaluator shall confirm that the CM system is being used that applies to the TOE manufacturing process.

ACM\_CAP-R.5.8E(+) The evaluator shall confirm that the CM system is being used that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-R.5.9E(+) The evaluator shall confirm that the CM system is being used that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-R.5.10E(+) The evaluator shall confirm that the CM system is being used that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-R.5.11E(+) The evaluator shall confirm that the CM system is being used that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-R.5.12E(+) The evaluator shall determine that the CM system is being used that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-R.5.13E(+) The evaluator shall determine that the CM system is being used that provides for an adequate and appropriate review of changes to all configuration items.

#### **4.9.3 ACM\_SCP-R.3 Development tools CM coverage**

**Objectives:** A CM system can control changes only to those items that have been placed under CM. Placing the TOE implementation representation, design, tests, user and administrator documentation, and CM documentation under CM provides assurance that they have been modified in a controlled manner with proper authorizations. The ability to track security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution. Development tools play an important role in ensuring the production of a quality version of the TOE. Therefore, it is important to control modifications to these tools.

**Dependencies:** ACM\_CAP-R.3 Authorization controls

**Developer action elements:**

ACM\_SCP-R.3.1D The developer shall perform CM such that, as a minimum, the following are kept under CM: the TOE implementation representation, design documentation, test

documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

Content and presentation of evidence elements:

None

Evaluator action elements:

ACM\_SCP-R.3.1E The evaluator shall confirm that the following are being configuration managed: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

#### **4.9.4 ADO\_DEL-R.3 Detection of modification**

Dependencies: ACM\_CAP-R.3 Authorization controls

Developer action elements:

ADO\_DEL-R.3.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures necessary to maintain security when distributing versions of the TOE to a user's site.

ADO\_DEL-R.3.2D *CC element deleted*

ADO\_DEL-R.3.3D(+) The developer shall explain how the delivery procedures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-R.3.4D(+) The developer shall explain how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_DEL-R.3.1E The evaluator shall confirm that the documented delivery procedures describe the procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.

ADO\_DEL-R.3.2E(+) The evaluator shall confirm that the developer has explained how the delivery procedures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-R.3.3E(+) The evaluator shall confirm that the developer has explained how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

#### **4.9.5 ADO\_IGS-R.1 Installation generation and start-up procedures**

Dependencies: AGD\_ADM-R.1 Administrator guidance

Developer action elements:

ADO\_IGS-R.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADO\_IGS-R.1.1E The evaluator shall confirm that the documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

ADO\_IGS-R.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

#### **4.9.6 ADV\_FSP-R.4 Formal functional specification**

Dependencies: ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_FSP-R.4.1D The developer shall provide a formal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-R.4.2D(+) The developer shall rigorously justify that the TSF is a completely and correctly represented by this functional specification.

Content and presentation of evidence elements:

ADV\_FSP-R.4.1C The functional specification shall describe the TSF and its external interfaces using an informal style or a semiformal style, supported by informal, explanatory text where appropriate.

ADV\_FSP-R.4.2C The functional specification shall be internally consistent.

ADV\_FSP-R.4.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

Evaluator action elements:

ADV\_FSP-R.4.1E The evaluator shall confirm that the functional specification meets all requirements for content and presentation of evidence to the level of rigor of no errors or omissions and at a level of completeness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-R.4.2E The evaluator shall determine that the developer has rigorously justified that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

#### **4.9.7 ADV\_HLD-R.4 Semiformal or Informal high-level explanation**

Dependencies:

ADV\_FSP-R.1 Informal functional specification

ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_HLD-R.4.1D The developer shall provide the high-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_HLD-R.4.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a high-level design.

ADV\_HLD-R.4.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_HLD-R.4.1C The presentation of the high-level design shall be semiformal or informal.

ADV\_HLD-R.4.2C The high-level design shall be internally consistent.

ADV\_HLD-R.4.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-R.4.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-R.4.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-R.4.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-R.4.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV\_HLD-R.4.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_HLD-R.4.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

ADV\_HLD-R.4.10C The high-level design shall justify that the identified means of achieving separation, including any protection mechanisms, are sufficient to ensure a clear and effective separation of TSP-enforcing from non-TSP-enforcing functions.

ADV\_HLD-R.4.11C The high-level design shall justify that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design.

#### Evaluator action elements:

ADV\_HLD-R.4.1E The evaluator shall confirm that the documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-R.4.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_HLD-R.4.3E(+) The evaluator shall confirm that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-R.4.4E(+) The evaluator shall confirm that the developer design process maintains the high-level design to reflect the actual implementation.

#### **4.9.8 ADV\_IMP-R.3 Structured Implementation of the TSF**

Application notes: The ADV\_IMP-R.2.2E element defines a requirement that the evaluator determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements. This provides a direct correspondence between the TOE security functional requirements and the implementation representation, in addition to

the pairwise correspondences required by the ADV\_RCR family. It is expected that the evaluator will use the evidence provided in ADV\_RCR as an input to making this determination.

**Dependencies:**

ADV\_LLD-R.1 Descriptive low-level design  
ADV\_RCR-R.1 Informal correspondence demonstration  
ALC\_TAT-R.1 Well-defined development tools

**Developer action elements:**

ADV\_IMP-R.3.1D The developer shall produce an implementation representation for the entire TSF that unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-R.3.2D(+) The developer shall produce the implementation representation for the entire TSF such that this representation is structured into small and comprehensible sections.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADV\_IMP-R.3.1E The evaluator shall confirm that the implementation representation unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-R.3.2E The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_IMP-R.3.3E(+) The evaluator shall confirm that the implementation representation is structured into small and comprehensible sections.

#### **4.9.9 ADV\_INT-R.3 Minimization of complexity**

**Application notes:** This component requires that the reference monitor property "simple enough to be analyzed" is fully addressed. When this component is combined with the functional requirements FPT\_RVM.1 and FPT\_SEP.3, the reference monitor concept would be fully realized.

**Dependencies:**

ADV\_IMP-R.2 Implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design

**Developer action elements:**

ADV\_INT-R.3.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

ADV\_INT-R.3.2D *CC element deleted*

ADV\_INT-R.3.3D The developer shall design and structure the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-R.3.4D The developer shall design and structure the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-R.3.5D The developer shall design and structure the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-R.3.6D The developer shall ensure that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADV\_INT-R.3.1E The evaluator shall determine that the developer designed and structured, and implemented the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design

ADV\_INT-R.3.2E *Intent of CC element incorporated into \*.1E above*

ADV\_INT-R.3.3E(+) The evaluator shall determine that the developer designed and structured the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-R.3.4E(+) The evaluator shall determine that the developer designed and structured the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-R.3.5E(+) The evaluator shall determine that the developer designed and structured the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-R.3.6E(+) The evaluator shall determine that the developer ensured that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

#### 4.9.10 ADV\_LLD-R.1 Descriptive low-level design

Dependencies:

ADV\_HLD-R.2 Security enforcing high-level design

ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_LLD-R.1.1D The developer shall provide the low-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_LLD-R.1.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a low-level design.

ADV\_LLD-R.1.3D(+) The developer shall, as an essential part of the TOE development process, maintain the low-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_LLD-R.1.1C The presentation of the low-level design shall be informal.

ADV\_LLD-R.1.2C The low-level design shall be internally consistent.

ADV\_LLD-R.1.3C The low-level design shall describe the TSF in terms of modules.

ADV\_LLD-R.1.4C The low-level design shall describe the purpose of each module.

ADV\_LLD-R.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV\_LLD-R.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV\_LLD-R.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV\_LLD-R.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV\_LLD-R.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_LLD-R.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.



**Evaluator action elements:**

ADV\_LLD-R.1.1E The evaluator shall confirm that the documented low-level design meets all requirements for content and presentation of evidence.

ADV\_LLD-R.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_LLD-R.1.3E(+) The evaluator shall confirm that the developer design process produces the low-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_LLD-R.1.4E(+) The evaluator shall confirm that the developer design process maintains the low-level design to reflect the actual implementation.

**4.9.11 ADV\_RCR-R.3 Formal correspondence demonstration**

Dependencies: No dependencies.

**Developer action elements:**

ADV\_RCR-R.3.1D The developer shall conduct an analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV\_RCR-R.3.2D(+) The developer shall conduct a formal proof of correspondence between adjacent pair of TSF representations whenever both representations are formally specified.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ADV\_RCR-R.3.1E The evaluator shall confirm that the results of the developer analysis of correspondence show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV\_RCR-R.3.2E The evaluator shall determine the accuracy of the proofs of correspondence by selectively verifying the formal analysis.

ADV\_RCR-R.3.3E(+) The evaluator shall check that the developer produced a formal proof of correspondence between adjacent pair of TSF representations whenever both representations are formally specified.

#### **4.9.12 ADV\_SPM-R.3 Formal TOE security policy model**

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

ADV\_SPM-R.1.1D The developer shall produce a formal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.

ADV\_SPM-R.1.2D The developer shall prove correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

Content and presentation of evidence elements:

None

Evaluator action elements:

ADV\_SPM-R.3.1E The evaluator shall confirm that the SPM a formal TSP model that describes the rules and characteristics of the policies of the TSP.

ADV\_SPM-R.3.2E(+) The evaluator shall confirm that the developer has proven correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

#### **4.9.13 AGD\_ADM-R.1 Administrator guidance**

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

AGD\_ADM-R.1.1D The developer shall provide administrator guidance addressed to system administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_ADM-R.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD\_ADM-R.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD\_ADM-R.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-R.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-R.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-R.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-R.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-R.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD\_ADM-R.1.1E The evaluator shall confirm that this guidance contains no errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### 4.9.14 AGD\_USR-R.1 User guidance

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

AGD\_USR-R.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_USR-R.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-R.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-R.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-R.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-R.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-R.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD\_USR-R.1.1E The evaluator shall confirm that this guidance contains no errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

#### **4.9.15 ALC\_DVS-R.2 Sufficiency of security measures**

Dependencies: No dependencies.

Developer action elements:

ALC\_DVS-R.2.1D The developer shall identify the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-R.2.2D(+) The developer shall implement the measures identified.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_DVS-R.2.1E The evaluator shall confirm that the developer has identified the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-R.2.2E The evaluator shall confirm that the security measures are being applied.

#### **4.9.16 ALC\_FLR-R.2 Flaw reporting procedures**

Dependencies: No dependencies.

Developer action elements:

ALC\_FLR-R.2.1D The developer shall establish flaw remediation procedures that provide the capabilities included in the content and presentation section below.

ALC\_FLR-R.2.2D The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

#### Content and presentation of evidence elements:

ALC\_FLR-R.2.1C The flaw remediation procedures shall include tracking of all reported security flaws in each release of the TOE.

ALC\_FLR-R.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC\_FLR-R.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC\_FLR-R.2.4C The flaw remediation procedures shall include providing flaw information, corrections and guidance on corrective actions to TOE users.

ALC\_FLR-R.2.5C The procedures for processing reported security flaws shall include measures to ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC\_FLR-R.2.6C The procedures for processing reported security flaws shall include safeguards to reduce the potential for corrections to these security flaws to introduce new flaws.

#### Evaluator action elements:

ALC\_FLR-R.2.1E The evaluator shall confirm that the flaw remediation procedures meet all the requirements identified in the content and presentation of evidence section above.

ALC\_FLR-R.2.2E(+) The evaluator shall confirm that the flaw remediation procedures include provisions for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

#### **4.9.17 ALC\_LCD-R.3 Measurable life-cycle model**

Dependencies: No dependencies.

#### Developer action elements:

ALC\_LCD-R.3.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC\_LCD-R.3.2D *CC element deleted*

ALC\_LCD-R.3.3D The developer shall use a standardized and measurable life-cycle model to develop and maintain the TOE.

ALC\_LCD-R.3.4D The developer shall measure the TOE development using the standardized and measurable life-cycle model.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_LCD-R.3.1E The evaluator shall confirm that the developer is using a standardized and measurable life-cycle model for the development and maintenance of the TOE.

ALC\_LCD-R.3.2E(+) The evaluator shall confirm that the developer has measured the TOE development using the standardized and measurable life-cycle model.

#### **4.9.18 ALC\_TAT-R.3 Compliance with implementation standards - all parts**

Dependencies: ADV\_IMP-R.1 Subset of the implementation of the TSF

Developer action elements:

ALC\_TAT-L.3.1D The developer shall use well-defined development tools for the TOE.

ALC\_TAT-L.3.2D The developer shall identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.3.3D The developer shall apply identified implementation standards for all parts of the TOE as appropriate.

ALC\_TAT-L.3.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

None

Evaluator action elements:

ALC\_TAT-R.3.1E The evaluator shall confirm that the developer is using well-defined development tools for the TOE.

ALC\_TAT-R.3.2E The evaluator shall confirm that the identified implementation standards have been applied, as appropriate, to all parts of the TOE.

ALC\_TAT-R.3.3E(+) The evaluator shall confirm that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-R.3.4E(+) The evaluator shall confirm that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### 4.9.19 ATE\_COV-R.3 Analysis of coverage

**Objectives:** In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved through an evaluator examination of and by the developer conducting testing on the basis of a rigorous analysis of correspondence.

**Application notes:** The developer is required to provide a convincing argument that the tests which have been identified cover all security functions, and that the testing of each security function is complete. There will remain little scope for the evaluator to devise additional functional tests of the TSF interfaces based on the functional specification, as they will have been exhaustively tested. Nevertheless, the evaluator should strive to devise such tests.

**Dependencies:**

ADV\_FSP-R.1 Informal functional specification  
ATE\_FUN-R.1 Functional testing

**Developer action elements:**

ATE\_COV-R.3.1D The developer shall conduct testing on the basis of a documented analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

ATE\_COV-R.3.2D(+) The developer shall conduct testing on the basis of a rigorous analysis of the test coverage that was used to ensure that the tests conducted completely tested all internal interfaces of the TSF identified in the functional specification.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_COV-R.3.1E The evaluator shall confirm that the developer testing was performed on the basis of an analysis of test coverage that ensures that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

ATE\_COV-R.3.2E(+) The evaluator shall confirm that the developer testing was performed on the basis of a rigorous analysis of test coverage that ensures that all external interfaces of the TSF identified in the functional specification have been completely tested.

#### **4.9.20 ATE\_DPT-R.3 Testing: implementation representation**

**Objectives:** The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized. The modules of a TSF provide a description of the internal workings of the TSF. Testing at the level of the modules, in order to demonstrate the presence of any flaws, provides assurance that the TSF modules have been correctly realized. The implementation representation of a TSF provides a detailed description of the internal workings of the TSF. Testing at the level of the implementation, in order to demonstrate the presence of any flaws, provides assurance that the TSF implementation has been correctly realized.

**Application notes:** The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts. The developer is expected to describe the testing of the low-level design of the TSF in terms of "modules". The term "modules" is used to express the notion of decomposing each of the "subsystems" of the TSF into a relatively small number of parts. The implementation representation is the one which is used to generate the TSF itself (e.g. source code which is then compiled).

**Dependencies:**

ADV\_HLD-R.2 Security enforcing high-level design  
ADV\_IMP-R.2 Implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design  
ATE\_FUN-R.1 Functional testing

**Developer action elements:**

ATE\_DPT-R.3.1D The developer shall conduct testing on the basis of a documented analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_DPT-R.3.1E The evaluator shall confirm that developer testing was conducted on the basis of an analysis of the depth of testing that demonstrates that the tests identified in



the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation.

#### **4.9.21 ATE\_FUN-R.2 Ordered functional testing**

**Objectives:** The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation. In this component, an additional objective is to ensure that testing is structured such as to avoid circular arguments about the correctness of the portions of the TSF being tested.

**Application notes:** Although the test procedures may state pre-requisite initial test conditions in terms of ordering of tests, they may not provide a rationale for the ordering. An analysis of test ordering is an important factor in determining the adequacy of testing, as there is a possibility of faults being concealed by the ordering of tests.

**Dependencies:** No dependencies.

**Developer action elements:**

ATE\_FUN-R.2.1D The developer shall test the TSF.

ATE\_FUN-R.2.2D The developer shall produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

ATE\_FUN-R.2.3D(+) The developer shall conduct testing on the basis of an analysis of the test procedure ordering dependencies.

**Content and presentation of evidence elements:**

ATE\_FUN-R.2.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-R.2.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-R.2.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE\_FUN-R.2.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-R.2.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE\_FUN-R.2.6C The test documentation shall include an analysis of the test procedure ordering dependencies.

**Evaluator action elements:**

ATE\_FUN-R.2.1E The evaluator shall confirm that the that the developer tested the TSF.

ATE\_FUN-R.2.2E(+) The evaluator shall confirm that the developer conducted testing on the basis of an analysis of the test procedure ordering dependencies.

ATE\_FUN-R.2.3E(+) The evaluator shall confirm that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.

**4.9.22 ATE\_IND-R.3 Independent testing - complete**

**Objectives:** The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes repeating all of the developer tests.

**Application notes:** The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. In this component the evaluator must repeat all of the developer's tests as part of the program of testing. As in the previous component the evaluator will also conduct tests that aim to exercise the TOE in a different manner from that achieved by the developer. In cases where developer testing has been exhaustive, there may remain little scope for this.

**Dependencies:**

ADV\_FSP-R.1 Informal functional specification

AGD\_ADM-R.1 Administrator guidance

AGD\_USR-R.1 User guidance

ATE\_FUN-R.1 Functional testing

**Developer action elements:**

ATE\_IND-R.3.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

**Content and presentation of evidence elements:**

None

**Evaluator action elements:**

ATE\_IND-R.3.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE\_IND-R.3.2E The evaluator shall test the TSF as appropriate to confirm that the developer testing shows that the TOE meets all functional requirements in the associated security target..

ATE\_IND-R.3.3E The evaluator shall execute all of tests in the test documentation to verify the developer test results.

#### **4.9.23 AVA\_CCA-R.2 Systematic covert channel analysis**

**Objectives:** The objective is to identify covert channels that are identifiable, through an informal search for covert channels.

**Application notes:** Performing a covert channel analysis in a systematic way requires that the developer identify covert channels in a structured and repeatable way, as opposed to identifying covert channels in an ad-hoc fashion.

**Dependencies:**

ADV\_FSP-R.2 Fully defined external interfaces  
ADV\_IMP-R.2 Implementation of the TSF  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance

**Developer action elements:**

AVA\_CCA-R.2.1D The developer shall conduct a systematic search for covert channels for each information flow control policy; identifying each covert channel, estimating its capacity, and determining the worst-case exploitation scenario.

AVA\_CCA-R.2.2D The developer shall provide covert channel analysis documentation containing the information identified in the content and presentation section below.

**Content and presentation of evidence elements:**

AVA\_CCA-R.2.1C The analysis documentation shall identify covert channels and estimate their capacity.

AVA\_CCA-R.2.2C The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.

AVA\_CCA-R.2.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.

AVA\_CCA-R.2.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.

AVA\_CCA-R.2.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

AVA\_CCA-R.2.6C The analysis documentation shall provide evidence that the method used to identify covert channels is systematic.

Evaluator action elements:

AVA\_CCA-R.2.1E The evaluator shall confirm that the developer has systematically identified covert channels and for each channel estimated its capacity, and determined the worst-case exploitation scenario.

AVA\_CCA-R.2.2E The evaluator shall confirm that the results of the covert channel analysis show that the TOE meets its functional requirements.

AVA\_CCA-R.2.3E The evaluator shall selectively validate the covert channel analysis through testing.

#### **4.9.24 AVA\_MSU-R.3 Analysis and testing for insecure states**

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met, and this analysis is validated and confirmed through testing by the developer and the evaluator.

**Application notes:** In this component the developer and the evaluator are required to undertake testing to ensure that if and when the TOE enters an insecure state this may easily be detected. This testing may be considered as a specific aspect of penetration testing.

Dependencies:

ADO\_IGS-R.1 Installation, generation, and start-up procedures

ADV\_FSP-R.1 Informal functional specification

AGD\_ADM-R.1 Administrator guidance

AGD\_USR-R.1 User guidance

Developer action elements:

AVA\_MSU-R.3.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-R.3.2D The developer shall analyze the guidance documentation to determine that the guidance documentation is complete.

AVA\_MSU-R.3.3D(+) The developer shall perform testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

### Content and presentation of evidence elements:

AVA\_MSU-R.3.1C The guidance documentation shall identify possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-R.3.2C *CC element incorporated into developer elements*

AVA\_MSU-R.3.3C The guidance documentation shall list assumptions about the intended environment.

AVA\_MSU-R.3.4C The guidance documentation shall list requirements for external security measures (including external procedural, physical and personnel controls).

### Evaluator action elements:

AVA\_MSU-R.3.1E The evaluator shall confirm that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-R.3.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-R.3.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA\_MSU-R.3.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

AVA\_MSU-R.3.5E The evaluator shall perform independent testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

AVA\_MSU-R.3.6E(+) The evaluator shall confirm that the developer performed testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

#### 4.9.25 AVA\_SOF-R.1 Strength of TOE security function evaluation

Dependencies:

ADV\_FSP-R.1 Informal functional specification  
ADV\_HLD-R.1 Descriptive high-level design

Developer action elements:

AVA\_SOF-R.1.1D The developer shall perform a strength of TOE security function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

Content and presentation of evidence elements:

None

Evaluator action elements:

AVA\_SOF-R.1.1E The evaluator shall confirm that the developer's strength of function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

AVA\_SOF-R.1.2E The evaluator shall determine that the strength claims are correct.

#### 4.9.26 AVA\_VLA-R.4 Highly resistant

Objectives. A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks using publicly known attacks and otherwise obvious attack methods. The developer and the evaluator perform penetration testing, supported by a vulnerability analysis, to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential.

Dependencies:

ADV\_FSP-R.1 Informal functional specification  
ADV\_HLD-R.2 Security enforcing high-level design  
ADV\_IMP-R.1 Subset of the implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design  
ADV\_INT-R.3 Minimization of complexity  
AGD\_ADM-R.1 Administrator guidance

AGD\_USR-R.1 User guidance  
FPT\_RVM.1 Non-Bypassability of the TSP  
FPT\_SEP.3 Complete reference monitor

**Developer action elements:**

AVA\_VLA-R.4.1D The developer shall perform and document a systematic analysis of the TOE that completely addresses TOE deliverables searching for ways in which a user possessing a high attack potential can violate the TSP.

AVA\_VLA-R.4.2D The developer shall conduct penetration testing based upon this analysis to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential document the disposition of identified vulnerabilities.

AVA\_VLA-R.4.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

**Content and presentation of evidence elements:**

AVA\_VLA-R.4.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA\_VLA-R.4.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA\_VLA-R.4.3C The evidence shall show that the search for vulnerabilities is systematic.

AVA\_VLA-R.4.4C The analysis documentation shall provide a justification that the analysis completely addresses the TOE deliverables.

**Evaluator action elements:**

AVA\_VLA-R.4.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing.

AVA\_VLA-R.4.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA\_VLA-R.4.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods and with respect to attackers with high attack capability.

AVA\_VLA-R.4.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods and to attackers with high attack capability.

AVA\_VLA-R.4.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods and by an attacker possessing a high attack potential.

AVA\_VLA-R.4.6E(+) The evaluator shall determine that the developer has performed penetration testing on the basis of a systematic vulnerability analysis to determine that the TOE is resistant to penetration attacks by attackers possessing high attack capability.

DRAFT



## APPENDIX A



## A. CATALOG OF ASSURANCE COMPONENTS

COMPONENTS EXPRESSED AS ADDITIONS AND DELETIONS FROM EXISTING CC COMPONENTS

In the following, the similar Common Criteria component is used as the starting point with changes marked as follows:

Additions: additions are shown as underline

Deletions: ~~deletions are shown as strikethrough~~

### A.1 CONFIGURATION MANAGEMENT

#### A.1.1 ACM\_AUT-L.1 Partial CM automation

Objectives: In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorized. It is the objective of this component to ensure that the implementation representation is controlled through automated means.

Dependencies: ACM\_CAP-L.3 Authorization controls

Developer action elements:

ACM\_AUT-L.1.1D The developer shall use a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation.

ACM\_AUT-L.1.2D *placeholder for element appearing in higher component* ~~The developer shall provide a CM plan.~~

ACM\_AUT-L.1.3D(+) The developer shall use a CM system that provides automated means to support the generation of the TOE.

Content and presentation of evidence elements:

~~ACM\_AUT-L.1.1C The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.~~

~~ACM\_AUT-L.1.2C The CM system shall provide an automated means to support the generation of the TOE.~~

~~ACM\_AUT-L.1.3C The CM plan shall describe the automated tools used in the CM system.~~

~~ACM\_AUT-L.1.4C The CM plan shall describe how the automated tools are used in the CM system.~~

None

Evaluator action elements:

~~ACM\_AUT-L.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence check that the developer is using a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation~~

ACM\_AUT-L.1.2E(+) The evaluator shall check that the developer is using a CM system that includes automated means to support the generation of the TOE.

### **A.1.2 ACM\_AUT-R.1 Partial CM automation**

Objectives: In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorized. It is the objective of this component to ensure that the implementation representation is controlled through automated means.

Dependencies: ACM\_CAP-R.3 Authorization controls

Developer action elements:

ACM\_AUT-R.1.1D The developer shall use a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation.

ACM\_AUT-R.1.2D The developer shall provide a CM plan.

ACM\_AUT-R.1.3D(+) The developer shall use a CM system that provides automated means to support the generation of the TOE.

Content and presentation of evidence elements:

~~ACM\_AUT-R.1.1C The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.~~

~~ACM\_AUT-R.1.2C The CM system shall provide an automated means to support the generation of the TOE.~~

~~ACM\_AUT-R.1.3C The CM plan shall describe the automated tools used in the CM system.~~

~~ACM\_AUT-R.1.4C The CM plan shall describe how the automated tools are used in the CM system.~~

None

Evaluator action elements:

ACM\_AUT-R.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence that the developer is using a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation.

ACM\_AUT-R.1.2E(+) The evaluator shall confirm that the developer is using a CM system that includes automated means to support the generation of the TOE.

### **A.1.3 ACM\_AUT-L.2 Complete CM automation**

Objectives: In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorized. It is the objective of this component to ensure that the implementation representation is controlled through automated means. Providing an automated means of ascertaining changes between versions of the TOE and identifying which configuration items are affected by modifications to other configuration items assists in determining the impact of the changes between successive versions of the TOE. This in turn can provide valuable information in determining whether changes to the TOE result in all configuration items being consistent with one another.

Dependencies: ACM\_CAP-L.3 Authorization controls

Developer action elements:

ACM\_AUT-L.2.1D The developer shall use a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

*ACM\_AUT-L.2.2D placeholder for element appearing in higher component* ~~The developer shall provide a CM plan.~~

ACM\_AUT-L.2.3D(+) The developer shall use a CM system that provides automated means to support the generation of the TOE.

ACM\_AUT-L.2.4D(+) The developer shall use a CM system that provides automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-L.2.5D(+) The developer shall use a CM system that provides automated means to identify all other configuration items that are affected by the modification of a given configuration item.

Content and presentation of evidence elements:

~~ACM\_AUT-L.2.1C~~ The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation, and to all other configuration items.

~~ACM\_AUT-L.2.2C~~ The CM system shall provide an automated means to support the generation of the TOE.

~~ACM\_AUT-L.2.3C~~ The CM plan shall describe the automated tools used in the CM system.

~~ACM\_AUT-L.2.4C~~ The CM plan shall describe how the automated tools are used in the CM system.

~~ACM\_AUT-L.2.5C~~ The CM system shall provide an automated means to ascertain the changes between the TOE and its preceding version.

~~ACM\_AUT-L.2.6C~~ The CM system shall provide an automated means to identify all other configuration items that are affected by the modification of a given configuration item.

None

Evaluator action elements:

ACM\_AUT-L.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence check that the developer is using a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

ACM\_AUT-L.2.2E The evaluator shall confirm, at the level of appears to be true, that the developer is using a CM system that includes automated means to support the generation of the TOE.

ACM\_AUT-L.2.3E(+) The evaluator shall confirm, at the level of appears to be true, that the developer is using a CM system that includes automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-L.2.4E(+) The evaluator shall confirm, at the level of appears to be true, that the developer is using a CM system that includes automated means to identify all other configuration items that are affected by the modification of a given configuration item.

#### A.1.4 ACM\_AUT-R.2 Complete CM automation

**Objectives:** In development environments where the implementation representation is complex or is being developed by multiple developers, it is difficult to control changes without the support of automated tools. In particular, these automated tools need to be able to support the numerous changes that occur during development and ensure that those changes are authorized. It is the objective of this component to ensure that the implementation representation is controlled through automated means. Providing an automated means of ascertaining changes between versions of the TOE and identifying which configuration items are affected by modifications to other configuration items assists in determining the impact of the changes between successive versions of the TOE. This in turn can provide valuable information in determining whether changes to the TOE result in all configuration items being consistent with one another.

**Dependencies:** ACM\_CAP-R.3 Authorization controls

**Developer action elements:**

ACM\_AUT-R.2.1D The developer shall use a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.

ACM\_AUT-R.2.2D The developer shall provide a CM plan.

ACM\_AUT-R.2.3D(+) The developer shall use a CM system that provides automated means to support the generation of the TOE.

ACM\_AUT-R.2.4D(+) The developer shall use a CM system that provides automated means to ascertain the changes between the TOE and its preceding version.

ACM\_AUT-R.2.5D(+) The developer shall use a CM system that provides automated means to identify all other configuration items that are affected by the modification of a given configuration item.

**Content and presentation of evidence elements:**

~~ACM\_AUT-R.2.1C The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation, and to all other configuration items.~~

~~ACM\_AUT-R.2.2C The CM system shall provide an automated means to support the generation of the TOE.~~

~~ACM\_AUT-R.2.3C The CM plan shall describe the automated tools used in the CM system.~~

~~ACM\_AUT-R.2.4C The CM plan shall describe how the automated tools are used in the CM system.~~

~~ACM\_AUT-R.2.5C The CM system shall provide an automated means to ascertain the changes between the TOE and its preceding version.~~

~~ACM\_AUT-R.2.6C The CM system shall provide an automated means to identify all other configuration items that are affected by the modification of a given configuration item.~~

None

Evaluator action elements:

~~ACM\_AUT-L.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence developer is using a CM system that includes automated means to help ensure that only authorized changes are made to the TOE implementation representation and to all other configuration items.~~

~~ACM\_AUT-L.2.2E The evaluator shall confirm that the developer is using a CM system that includes automated means to support the generation of the TOE.~~

~~ACM\_AUT-L.2.3E(+) The evaluator shall confirm that the developer is using a CM system that includes automated means to ascertain the changes between the TOE and its preceding version.~~

~~ACM\_AUT-L.2.4E(+) The evaluator shall confirm that the developer is using a CM system that includes automated means to identify all other configuration items that are affected by the modification of a given configuration item.~~

#### **A.1.5 ACM\_CAP-L.1 Version numbers**

Objectives. A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using.

Dependencies: No dependencies.

Developer action elements:

~~ACM\_CAP-L.1.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.~~

Content and presentation of evidence elements:

~~ACM\_CAP-L.1.1C The reference for the TOE shall be unique to each version of the TOE.~~  
~~ACM\_CAP-L.1.2C The TOE shall be labeled with its reference.~~

None

Evaluator action elements:

~~ACM\_CAP-L.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ check that the TOE is labeled with a reference that can reasonably be expected to be unique to each version of the TOE.

#### **A.1.6 ACM\_CAP-L.3 Authorization controls**

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE.

**Dependencies:**

ACM\_SCP-L.1 TOE CM coverage  
ALC\_DVS-L.1 Identification of security measures

**Developer action elements:**

ACM\_CAP-L.3.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-L.3.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-L.3.3D The developer shall provide ~~CM documentation~~ a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.3.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

**Content and presentation of evidence elements:**

~~ACM\_CAP-L.3.1C The reference for the TOE shall be unique to each version of the TOE.~~

~~ACM\_CAP-L.3.2C The TOE shall be labeled with its reference.~~

~~ACM\_CAP-L.3.3C The CM documentation shall include a configuration list and a CM plan.~~

~~ACM\_CAP-L.3.4C The configuration list shall describe the configuration items that comprise the TOE.~~

~~ACM\_CAP-L.3.5C The CM documentation shall describe the method used to uniquely identify the configuration items.~~

~~ACM\_CAP-L.3.6C The CM system shall uniquely identify all configuration items.~~

~~ACM\_CAP-L.3.7C The CM plan shall describe how the CM system is used.~~

~~ACM\_CAP-L.3.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.~~

~~ACM\_CAP-L.3.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.~~

~~ACM\_CAP-L.3.10C The CM system shall provide measures such that only authorized changes are made to the configuration items.~~

None

Evaluator action elements:

~~ACM\_CAP-L.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-L.3.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to uniquely identify all configuration items and provide measures such that only authorized changes are made to the configuration items.

#### **A.1.7 ACM\_CAP-L.4 Generation support and acceptance procedures**

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE. The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized.

Dependencies:

ACM\_SCP-L.1 TOE CM coverage

ALC\_DVS-L.1 Identification of security measures

Developer action elements:

ACM\_CAP-L.4.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.



ACM\_CAP-L.4.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-L.4.3D The developer shall provide CM documentation a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.4.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

ACM\_CAP-L.4.5D The developer shall use a CM system that supports the generation of the TOE.

ACM\_CAP-L.4.6D The developer shall use a CM system that includes procedures used to accept modified or newly created configuration items as part of the TOE.

Content and presentation of evidence elements:

~~ACM\_CAP-L.4.1C The reference for the TOE shall be unique to each version of the TOE.~~

~~ACM\_CAP-L.4.2C The TOE shall be labeled with its reference.~~

~~ACM\_CAP-L.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.~~

~~ACM\_CAP-L.4.4C The configuration list shall describe the configuration items that comprise the TOE.~~

~~ACM\_CAP-L.4.5C The CM documentation shall describe the method used to uniquely identify the configuration items.~~

~~ACM\_CAP-L.4.6C The CM system shall uniquely identify all configuration items.~~

~~ACM\_CAP-L.4.7C The CM plan shall describe how the CM system is used.~~

~~ACM\_CAP-L.4.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.~~

~~ACM\_CAP-L.4.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.~~

~~ACM\_CAP-L.4.10C The CM system shall provide measures such that only authorized changes are made to the configuration items.~~

~~ACM\_CAP-L.4.11C The CM system shall support the generation of the TOE.~~

~~ACM\_CAP-L.4.12C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.~~

None

Evaluator action elements:

ACM\_CAP-L.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-L.4.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L4.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to uniquely identify all configuration items.

ACM\_CAP-L4.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to provide measures such that only authorized changes are made to the configuration items.

ACM\_CAP-L4.5E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to support generation of the TOE.

ACM\_CAP-L4.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to accept modified or newly created configuration items as part of the TOE.

#### **A.1.8 ACM\_CAP-R.4 Generation support and acceptance procedures**

Objectives: A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE. The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized.

Dependencies:

ACM\_SCP-R.1 TOE CM coverage  
ALC\_DVS-R.1 Identification of security measures

Developer action elements:

ACM\_CAP-R.4.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-R.4.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-R.4.3D The developer shall provide CM documentation a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-R.4.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

ACM\_CAP-R.4.5D The developer shall use a CM system that supports the generation of the TOE.

ACM\_CAP-R.4.6D The developer shall use a CM system that includes procedures used to accept modified or newly created configuration items as part of the TOE.

Content and presentation of evidence elements:

~~ACM\_CAP-R.4.1C The reference for the TOE shall be unique to each version of the TOE.~~

~~ACM\_CAP-R.4.2C The TOE shall be labeled with its reference.~~

~~ACM\_CAP-R.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.~~

~~ACM\_CAP-R.4.4C The configuration list shall describe the configuration items that comprise the TOE.~~

~~ACM\_CAP-R.4.5C The CM documentation shall describe the method used to uniquely identify the configuration items.~~

~~ACM\_CAP-R.4.6C The CM system shall uniquely identify all configuration items.~~

~~ACM\_CAP-R.4.7C The CM plan shall describe how the CM system is used.~~

~~ACM\_CAP-R.4.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.~~

~~ACM\_CAP-R.4.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.~~

~~ACM\_CAP-R.4.10C The CM system shall provide measures such that only authorized changes are made to the configuration items.~~

~~ACM\_CAP-R.4.11C The CM system shall support the generation of the TOE.~~

~~ACM\_CAP-R.4.12C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.~~

None

Evaluator action elements:

~~ACM\_CAP-R.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-R.4.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-R.4.3E(+) The evaluator shall confirm that the CM system is being used to uniquely identify all configuration items.

ACM\_CAP-R4.4E(+) The evaluator shall confirm that the CM system is being used to provide measures such that only authorized changes are made to the configuration items.

ACM\_CAP-R4.5E(+) The evaluator shall confirm that the CM system is being used to support generation of the TOE.

ACM\_CAP-R4.6E(+) The evaluator shall confirm that the CM system is being used to accept modified or newly created configuration items as part of the TOE.

#### **A.1.9 ACM\_CAP-L.5 Generation support and acceptance procedures**

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE. The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized. Integration procedures help to ensure that generation of the TOE from a managed set of configuration items is correctly performed in an authorized manner. Requiring that the CM system be able to identify the master copy of the material used to generate the TOE helps to ensure that the integrity of this material is preserved by the appropriate technical, physical and procedural safeguards.

**Dependencies:**

ACM\_SCP-L.1 TOE CM coverage  
ALC\_DVS-L.2 Sufficiency of security measures

**Developer action elements:**

ACM\_CAP-L.5.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-L.5.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-L.5.3D The developer shall provide CM documentation a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.5.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

ACM\_CAP-L.5.5D The developer shall use a CM system that supports the generation of the TOE.

ACM\_CAP-L.5.6D The developer shall use a CM system that includes procedures used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-L.5.7D The developer shall use a CM system that applies to the TOE manufacturing process.

ACM\_CAP-L.5.8D The developer shall use a CM system that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-L.5.9D The developer shall use a CM system that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-L.5.10D The developer shall use a CM system that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-L.5.11D The developer shall use a CM system that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-L.5.12D The developer shall use a CM system that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-L.5.13D The developer shall use a CM system that provides for an adequate and appropriate review of changes to all configuration items.

Content and presentation of evidence elements:

~~ACM\_CAP-L.5.1C The reference for the TOE shall be unique to each version of the TOE.~~

~~ACM\_CAP-L.5.2C The TOE shall be labeled with its reference.~~

~~ACM\_CAP-L.5.3C The CM documentation shall include a configuration list, a CM plan, an acceptance plan, and integration procedures.~~

~~ACM\_CAP-L.5.4C The configuration list shall describe the configuration items that comprise the TOE.~~

~~ACM\_CAP-L.5.5C The CM documentation shall describe the method used to uniquely identify the configuration items.~~

~~ACM\_CAP-L.5.6C The CM system shall uniquely identify all configuration items.~~

~~ACM\_CAP-L.5.7C The CM plan shall describe how the CM system is used.~~

~~ACM\_CAP-L.5.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.~~

~~ACM\_CAP-L.5.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.~~

~~ACM\_CAP-L.5.10C The CM system shall provide measures such that only authorized changes are made to the configuration items.~~

~~ACM\_CAP-L.5.11C The CM system shall support the generation of the TOE.~~

~~ACM\_CAP-L.5.12C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.~~

~~ACM\_CAP-L.5.13C The integration procedures shall describe how the CM system is applied in the TOE manufacturing process.~~

~~ACM\_CAP-L.5.14C The CM system shall require that the person responsible for accepting a configuration item into CM is not the person who developed it.~~

~~ACM\_CAP-L.5.15C The CM system shall clearly identify the configuration items that comprise the TSF.~~

~~ACM\_CAP-L.5.16C The CM system shall support the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.~~

~~ACM\_CAP-L.5.17C The CM system shall be able to identify the master copy of all material used to generate the TOE.~~

~~ACM\_CAP-L.5.18C The CM documentation shall demonstrate that the use of the CM system, together with the development security measures, allow only authorized changes to be made to the TOE.~~

~~ACM\_CAP-L.5.19C The CM documentation shall demonstrate that the use of the integration procedures ensures that the generation of the TOE is correctly performed in an authorized manner.~~

~~ACM\_CAP-L.5.20C The CM documentation shall demonstrate that the CM system is sufficient to ensure that the person responsible for accepting a configuration item into CM is not the person who developed it.~~

~~ACM\_CAP-L.5.21C The CM documentation shall justify that the acceptance procedures provide for an adequate and appropriate review of changes to all configuration items.~~

None

Evaluator action elements:

ACM\_CAP-L.5.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-L.5.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-L.5.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to uniquely identify all configuration items.

ACM\_CAP-L.5.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to provide measures such that only authorized changes are made to the configuration items.

ACM\_CAP-L.5.5E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to support generation of the TOE.

ACM\_CAP-L.5.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-L.5.7E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that applies to the TOE manufacturing process.

ACM\_CAP-L.5.8E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-L.5.9E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-L.5.10E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-L.5.11E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-L.5.12E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-L.5.13E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the CM system is being used that provides for an adequate and appropriate review of changes to all configuration items.

#### **A.1.10 ACM\_CAP-R.5 Generation support and acceptance procedures**

**Objectives:** A unique reference is required to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with its reference ensures that users of the TOE can be aware of which instance of the TOE they are using. Unique identification of the configuration items leads to a clearer understanding of the composition of the TOE, which in turn helps to determine those items which are subject to the evaluation requirements for the TOE. Providing controls to ensure that unauthorized modifications are not made to the TOE, and ensuring proper functionality and use of the CM system, helps to maintain the integrity of the TOE. The purpose of acceptance procedures is to confirm that any creation or modification of configuration items is authorized. Integration procedures help to ensure that generation of the TOE from a managed set of configuration items is correctly performed in an authorized manner. Requiring that the CM system be able to identify the master copy of the material used to generate the TOE helps to ensure that the integrity of this material is preserved by the appropriate technical, physical and procedural safeguards.

**Dependencies:**

ACM\_SCP-R.1 TOE CM coverage  
ALC\_DVS-R.2 Sufficiency of security measures

Developer action elements:

ACM\_CAP-R.5.1D The developer shall label the TOE with a reference for the TOE that is unique to each version of the TOE.

ACM\_CAP-R.5.2D The developer shall use a CM system that uniquely identifies all configuration items.

ACM\_CAP-R.5.3D The developer shall provide ~~CM documentation~~ a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-R.5.4D The developer shall use a CM system that provides measures such that only authorized changes are made to the configuration items.

ACM\_CAP-R.5.5D The developer shall use a CM system that supports the generation of the TOE.

ACM\_CAP-R.5.6D The developer shall use a CM system that includes procedures used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-R.5.7D The developer shall use a CM system that applies to the TOE manufacturing process.

ACM\_CAP-R.5.8D The developer shall use a CM system that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-R.5.9D The developer shall use a CM system that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-R.5.10D The developer shall use a CM system that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-R.5.11D The developer shall use a CM system that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-R.5.12D The developer shall use a CM system that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-R.5.13D The developer shall use a CM system that provides for an adequate and appropriate review of changes to all configuration items.

Content and presentation of evidence elements:



~~ACM\_CAP R.5.1C The reference for the TOE shall be unique to each version of the TOE.~~

~~ACM\_CAP R.5.2C The TOE shall be labeled with its reference.~~

~~AL7 CM\_CAP R.5.3C The CM documentation shall include a configuration list, a CM plan, an acceptance plan, and integration procedures.~~

~~ACM\_CAP R.5.4C The configuration list shall describe the configuration items that comprise the TOE.~~

~~ACM\_CAP R.5.5C The CM documentation shall describe the method used to uniquely identify the configuration items.~~

~~ACM\_CAP R.5.6C The CM system shall uniquely identify all configuration items.~~

~~ACM\_CAP R.5.7C The CM plan shall describe how the CM system is used.~~

~~ACM\_CAP R.5.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.~~

~~ACM\_CAP R.5.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.~~

~~ACM\_CAP R.5.10C The CM system shall provide measures such that only authorized changes are made to the configuration items.~~

~~ACM\_CAP R.5.11C The CM system shall support the generation of the TOE.~~

~~ACM\_CAP R.5.12C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.~~

~~ACM\_CAP R.5.13C The integration procedures shall describe how the CM system is applied in the TOE manufacturing process.~~

~~ACM\_CAP R.5.14C The CM system shall require that the person responsible for accepting a configuration item into CM is not the person who developed it.~~

~~ACM\_CAP R.5.15C The CM system shall clearly identify the configuration items that comprise the TSF.~~

~~ACM\_CAP R.5.16C The CM system shall support the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.~~

~~ACM\_CAP R.5.17C The CM system shall be able to identify the master copy of all material used to generate the TOE.~~

~~ACM\_CAP R.5.18C The CM documentation shall demonstrate that the use of the CM system, together with the development security measures, allow only authorized changes to be made to the TOE.~~

~~ACM\_CAP R.5.19C The CM documentation shall demonstrate that the use of the integration procedures ensures that the generation of the TOE is correctly performed in an authorized manner.~~

~~ACM\_CAP R.5.20C The CM documentation shall demonstrate that the CM system is sufficient to ensure that the person responsible for accepting a configuration item into CM is not the person who developed it.~~

~~ACM\_CAP-R.5.21C The CM documentation shall justify that the acceptance procedures provide for an adequate and appropriate review of changes to all configuration items.~~

None

Evaluator action elements:

~~ACM\_CAP-R.5.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ check that the TOE is labeled with a reference to that is unique to that version of the TOE.

ACM\_CAP-R.5.2E(+) The evaluator shall check that the developer has provided a configuration list describing the configuration items that comprise the TOE.

ACM\_CAP-R.5.3E(+) The evaluator shall confirm that the CM system is being used to uniquely identify all configuration items.

ACM\_CAP-R.5.4E(+) The evaluator shall confirm that the CM system is being used to provide measures such that only authorized changes are made to the configuration items.

ACM\_CAP-R.5.5E(+) The evaluator shall confirm that the CM system is being used to support generation of the TOE.

ACM\_CAP-R.5.6E(+) The evaluator shall confirm that the CM system is being used to accept modified or newly created configuration items as part of the TOE.

ACM\_CAP-R.5.7E(+) The evaluator shall confirm that the CM system is being used that applies to the TOE manufacturing process.

ACM\_CAP-R.5.8E(+) The evaluator shall confirm that the CM system is being used that ensures that the person responsible for accepting a configuration item into CM is not the person who developed it.

ACM\_CAP-R.5.9E(+) The evaluator shall confirm that the CM system is being used that clearly identifies the configuration items that comprise the TSF.

ACM\_CAP-R.5.10E(+) The evaluator shall confirm that the CM system is being used that supports the audit of all modifications to the TOE, including as a minimum the originator, date, and time in the audit trail.

ACM\_CAP-R.5.11E(+) The evaluator shall confirm that the CM system is being used that is able to identify the master copy of all material used to generate the TOE.

ACM\_CAP-R.5.12E(+) The evaluator shall determine that the CM system is being used that ensures that the generation of the TOE is correctly performed in an authorized manner.

ACM\_CAP-R.5.13E(+) The evaluator shall determine that the CM system is being used that provides for an adequate and appropriate review of changes to all configuration items.

#### **A.1.11 ACM\_SCP-L.2 Problem tracking CM coverage**

**Objectives:** A CM system can control changes only to those items that have been placed under CM. Placing the TOE implementation representation, design, tests, user and administrator documentation, and CM documentation under CM provides assurance that they have been modified in a controlled manner with proper authorizations. The ability to track security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution.

**Dependencies:** ACM\_CAP-L.3 Authorization controls

**Developer action elements:**

ACM\_SCP-L.2.1D The developer shall ~~provide CM documentation~~ perform CM such that, as a minimum, the following are kept under CM: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.

**Content and presentation of evidence elements:**

~~ACM\_SCP-L.2.1C~~ The CM documentation shall show that the CM system, as a minimum, ~~tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.~~

~~ACM\_SCP-L.2.2C~~ The CM documentation shall describe how configuration items are tracked by the CM system.

None

**Evaluator action elements:**

ACM\_SCP-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the following are being configuration managed: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws. ~~that the information provided meets all requirements for content and presentation of evidence~~

#### **A.1.12 ACM\_SCP-L.3 Development tools CM coverage**

**Objectives:** A CM system can control changes only to those items that have been placed under CM. Placing the TOE implementation representation, design, tests, user and administrator documentation, and CM documentation under CM provides assurance that they have been modified in a controlled manner with proper authorizations. The ability to track security

flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution. Development tools play an important role in ensuring the production of a quality version of the TOE. Therefore, it is important to control modifications to these tools.

Dependencies: ACM\_CAP-L.3 Authorization controls

Developer action elements:

ACM\_SCP-L.3.1D The developer shall ~~provide CM documentation~~ perform CM such that, as a minimum, the following are kept under CM: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

Content and presentation of evidence elements:

~~ACM\_SCP-L.3.1C The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.~~

~~ACM\_SCP-L.3.2C The CM documentation shall describe how configuration items are tracked by the CM system.~~

None

Evaluator action elements:

ACM\_SCP-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the following are being configuration managed: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information. ~~that the information provided meets all requirements for content and presentation of evidence~~

### **A.1.13 ACM\_SCP-R.3 Development tools CM coverage**

Objectives: A CM system can control changes only to those items that have been placed under CM. Placing the TOE implementation representation, design, tests, user and administrator documentation, and CM documentation under CM provides assurance that they have been modified in a controlled manner with proper authorizations. The ability to track security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution. Development tools play an important role in ensuring the production of a quality version of the TOE. Therefore, it is important to control modifications to these tools.

Dependencies: ACM\_CAP-R.3 Authorization controls

Developer action elements:

ACM\_SCP-R.3.1D The developer shall ~~provide CM documentation~~ perform CM such that, as a minimum, the following are kept under CM: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.

Content and presentation of evidence elements:

~~ACM\_SCP-R.3.1C The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information.~~

~~ACM\_SCP-R.3.2C The CM documentation shall describe how configuration items are tracked by the CM system.~~

None

Evaluator action elements:

ACM\_SCP-R.3.1E The evaluator shall confirm that the following are being configuration managed: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, security flaws, and development tools and related information. ~~the information provided meets all requirements for content and presentation of evidence~~

## A.2 DELIVERY AND OPERATION

### A.1.14 ADO\_DEL-L.1 Delivery procedures

Dependencies: No dependencies.

Developer action elements:

ADO\_DEL-L.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.

~~ADO\_DEL-L.1.2D The developer shall use the delivery procedures.~~

Content and presentation of evidence elements:

~~ADO\_DEL-L.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.~~

None

Evaluator action elements:

~~ADO\_DEL-L.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ check that the documented delivery procedures describe the procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.

### **A.1.15 ADO\_DEL-L.2 Detection of modification**

Dependencies: ACM\_CAP-L.3 Authorization controls

Developer action elements:

~~ADO\_DEL-L.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures necessary to maintain security when distributing versions of the TOE to a user's site.~~

~~ADO\_DEL-L.2.2D CC element deleted The developer shall use the delivery procedures.~~

ADO\_DEL-L.2.3D(+) The developer shall explain how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-L.2.4D(+) The developer shall explain how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

Content and presentation of evidence elements:

~~ADO\_DEL-L.2.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.~~

~~ADO\_DEL-L.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.~~

~~ADO\_DEL-L.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.~~

None

Evaluator action elements:

~~ADO\_DEL-L.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ check that the documented delivery procedures describe the procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.

~~ADO\_DEL-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has explained how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.~~

~~ADO\_DEL-L.2.3E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has explained how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.~~

#### **A.1.16 ADO\_DEL-R.2 Detection of modification**

Dependencies: ACM\_CAP-R.3 Authorization controls

Developer action elements:

~~ADO\_DEL-R.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures necessary to maintain security when distributing versions of the TOE to a user's site.~~

~~ADO\_DEL-R.2.2D CC element deleted The developer shall use the delivery procedures.~~

~~ADO\_DEL-R.2.3D(+) The developer shall explain how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.~~

~~ADO\_DEL-R.2.4D(+) The developer shall explain how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.~~

Content and presentation of evidence elements:

~~ADO\_DEL-R.2.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.~~

~~ADO\_DEL-R.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.~~

~~ADO\_DEL-R.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.~~

None

Evaluator action elements:

~~ADO\_DEL-R.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ documented delivery procedures describe the procedures necessary to maintain security when distributing versions of the TOE to a user's site.

~~ADO\_DEL-R.2.2E(+) The evaluator shall confirm that the developer has explained how the delivery procedures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.~~

~~ADO\_DEL-R.2.3E(+) The evaluator shall confirm that the developer has explained how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.~~

#### **A.1.17 ADO\_DEL-L.3 Detection of modification**

Dependencies: ACM\_CAP-L.3 Authorization controls

Developer action elements:

~~ADO\_DEL-L.3.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures necessary to maintain security when distributing versions of the TOE to a user's site.~~

~~ADO\_DEL-L.3.2D CC element deleted The developer shall use the delivery procedures.~~

~~ADO\_DEL-L.3.3D(+) The developer shall explain how the delivery procedures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.~~

~~ADO\_DEL-L.3.4D(+) The developer shall explain how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.~~

Content and presentation of evidence elements:

~~ADO\_DEL-L.3.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.~~



~~ADO\_DEL-L.3.2C The delivery documentation shall describe how the various procedures and technical measures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.~~

~~ADO\_DEL-L.3.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.~~

None

Evaluator action elements:

~~ADO\_DEL-L.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ check that the documented delivery procedures describe the procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.

ADO\_DEL-L.3.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has explained how the delivery procedures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has explained how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

#### **A.1.18 ADO\_DEL-R.3 Detection of modification**

Dependencies: ACM\_CAP-R.3 Authorization controls

Developer action elements:

~~ADO\_DEL-R.3.1D The developer shall document procedures for delivery of the TOE or parts of it to the user; describing all procedures necessary to maintain security when distributing versions of the TOE to a user's site.~~

~~ADO\_DEL-R.3.2D CC element deleted The developer shall use the delivery procedures.~~

ADO\_DEL-R.3.3D(+) The developer shall explain how the delivery procedures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO\_DEL-R.3.4D(+) The developer shall explain how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

Content and presentation of evidence elements:

~~ADO\_DEL-R.3.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.~~

~~ADO\_DEL-R.3.2C The delivery documentation shall describe how the various procedures and technical measures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.~~

~~ADO\_DEL-R.3.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.~~

None

Evaluator action elements:

~~ADO\_DEL-R.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence documented delivery procedures describe the procedures that the developer deems necessary to maintain security when distributing versions of the TOE to a user's site.~~

~~ADO\_DEL-R.3.2E(+) The evaluator shall confirm that the developer has explained how the delivery procedures provide for the prevention of modifications, or any discrepancy between the developer's master copy and the version received at the user site.~~

~~ADO\_DEL-R.3.3E(+) The evaluator shall confirm that the developer has explained how the deliver procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.~~

#### **A.1.19 ADO\_IGS-L.1 Installation generation and start-up procedures**

Dependencies: AGD\_ADM-L.1 Administrator guidance

Developer action elements:

~~ADO\_IGS-L.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.~~

Content and presentation of evidence elements:

~~ADO\_IGS-L.1.1C The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.~~

None

Evaluator action elements:

~~ADO\_IGS-L.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ check that the documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

~~ADO\_IGS-L.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.~~

### **A.1.20 ADO\_IGS-R.1 Installation generation and start-up procedures**

Dependencies: AGD\_ADM-R.1 Administrator guidance

Developer action elements:

ADO\_IGS-R.1.1D The developer shall document procedures describing the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

~~ADO\_IGS-R.1.1C The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.~~

None

Evaluator action elements:

~~ADO\_IGS-R.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ documented procedures describe the steps deemed by the developer to be necessary for the secure installation, generation, and start-up of the TOE.

ADO\_IGS-R.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

## **A.3 DEVELOPMENT**

### **A.1.21 ADV\_FSP-L.1 Informal functional specification**

Dependencies: ~~ADV\_RCR.1 Informal correspondence demonstration~~ None

Developer action elements:

ADV\_FSP-L.1.1D The developer shall provide an informal functional specification with the information content described in the content and presentation section below at a level of

completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

Content and presentation of evidence elements:

ADV\_FSP-L.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV\_FSP-L.1.2C The functional specification shall be internally consistent.

ADV\_FSP-L.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV\_FSP-L.1.4C The functional specification shall completely represent the TSF.

Evaluator action elements:

~~ADV\_FSP-L.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ confirm, to the level of rigor of no obvious errors or omissions and at a level of completeness sufficient to support effective functional testing against the functional requirements in the associated security target, that the functional specification meets all requirements for content and presentation of evidence.

~~ADV\_FSP-L.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.~~

#### **A.1.22 ADV\_FSP-L.2 Fully defined external interfaces**

Dependencies: ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:

ADV\_FSP-L.2.1D The developer shall provide an informal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-L.2.2D(+) The developer shall show that the TSF is a completely and correctly represented by this functional specification.

Content and presentation of evidence elements:

ADV\_FSP-L.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV\_FSP-L.2.2C The functional specification shall be internally consistent.

ADV\_FSP-L.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

~~ADV\_FSP-L.2.4C The functional specification shall completely represent the TSF.~~

~~ADV\_FSP-L.2.5C The functional specification shall include rationale that the TSF is completely represented.~~

Evaluator action elements:

ADV\_FSP-L.2.1E The evaluator shall ~~confirm that the information provided meets all requirements for content and presentation of evidence~~ confirm, to the level of rigor of no obvious errors or omissions and at a level of completeness sufficient to support effective functional testing against the functional requirements in the associated security target that, the functional specification meets all requirements for content and presentation of evidence.

ADV\_FSP-L.2.2E The evaluator shall ~~determine~~ confirm, to the level of rigor of appears reasonable, that the developer has shown that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

### **A.1.23 ADV\_FSP-R.2 Fully defined external interfaces**

Dependencies: ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_FSP-R.2.1D The developer shall provide an informal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-R.2.2D(+) The developer shall show that the TSF is a completely and correctly represented by this functional specification.

Content and presentation of evidence elements:

ADV\_FSP-R.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV\_FSP-R.2.2C The functional specification shall be internally consistent.

ADV\_FSP-R.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

~~ADV\_FSP-R.2.4C The functional specification shall completely represent the TSF.~~

~~ADV\_FSP-R.2.5C The functional specification shall include rationale that the TSF is completely represented.~~

Evaluator action elements:

ADV\_FSP-R.2.1E The evaluator shall confirm, to the level of rigor for completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target, that the ~~information provided meets all requirements for content and presentation of evidence~~ functional specification meets all requirements for content and presentation of evidence.

ADV\_FSP-R.2.2E The evaluator shall determine that the developer has shown that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

**A.1.24 ADV\_FSP-L.4 Formal functional specification**

Dependencies: ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:

ADV\_FSP-L.4.1D The developer shall provide a formal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-L.4.2D(+) The developer shall rigorously justify that the TSF is a completely and correctly represented by this functional specification.

Content and presentation of evidence elements:

ADV\_FSP-L.4.1C The functional specification shall describe the TSF and its external interfaces using an informal style or a semiformal style, supported by informal, explanatory text where appropriate.

ADV\_FSP-L.4.2C The functional specification shall be internally consistent.

ADV\_FSP-L.4.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

~~ADV\_FSP-L.4.4C The functional specification shall completely represent the TSF.~~

~~ADV\_FSP-L.4.5C The functional specification shall include rationale that the TSF is completely represented.~~

Evaluator action elements:

~~ADV\_FSP-L.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ confirm, to the level of rigor of no obvious errors or omissions and at a level of completeness sufficient to support effective functional testing against the functional requirements in the associated security target, that the functional specification meets all requirements for content and presentation of evidence.

~~ADV\_FSP-L.4.2E The evaluator shall determine~~ confirm, to the level of rigor of appears reasonable, that the developer has rigorously justified that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

#### **A.1.25 ADV\_FSP-R.4 Formal functional specification**

Dependencies: ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

~~ADV\_FSP-R.4.1D The developer shall provide a formal functional specification with the information content described in the content and presentation section below at a level of completeness and correctness sufficient to support effective functional testing against the functional requirements in the associated security target.~~

~~ADV\_FSP-R.4.2D(+) The developer shall rigorously justify that the TSF is a completely and correctly represented by this functional specification.~~

Content and presentation of evidence elements:

~~ADV\_FSP-R.4.1C The functional specification shall describe the TSF and its external interfaces using an informal style or a semiformal style, supported by informal, explanatory text where appropriate.~~

~~ADV\_FSP-R.4.2C The functional specification shall be internally consistent.~~

~~ADV\_FSP-R.4.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.~~

~~ADV\_FSP-R.4.4C The functional specification shall completely represent the TSF.~~

~~ADV\_FSP-R.4.5C The functional specification shall include rationale that the TSF is completely represented.~~

Evaluator action elements:

~~ADV\_FSP-R.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ functional specification meets all requirements for content and presentation of evidence to the level of rigor of no errors or omissions and at a level of completeness sufficient to support effective functional testing against the functional requirements in the associated security target.

ADV\_FSP-R.4.2E The evaluator shall determine that the developer has rigorously justified that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

#### **A.1.26 ADV\_HLD-L.1 Descriptive high-level design**

Dependencies:

ADV\_FSP-L.1 Informal functional specification

ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:

ADV\_HLD-L.1.1D The developer shall ~~provide the~~ produce a high-level design of the TSF that provides the information identified in the content and presentation section below.

ADV\_HLD-L.1.2D(+) The developer shall, as an essential part of the TOE development process, produce the high-level design as a precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.1.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_HLD-L.1.1C The presentation of the high-level design shall be informal.

ADV\_HLD-L.1.2C The high-level design shall be internally consistent.

ADV\_HLD-L.1.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-L.1.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.



ADV\_HLD-L.1.5C The high-level design shall identify any underlying hardware, firm-ware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-L.1.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-L.1.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

#### Evaluator action elements:

ADV\_HLD-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the ~~information provided meets all requirements for content and presentation of evidence~~ documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-L.1.2E *placeholder for element appearing at higher component* ~~The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.~~

ADV\_HLD-L.1.3E(+) The evaluator shall confirm, the level of rigor of appears to be true, that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.1.4E(+) The evaluator shall confirm, the level of rigor of appears to be true, that the developer design process maintains the high-level design to reflect the actual implementation.

### A.1.27 ADV\_HLD-L.2 Security enforcing high-level design

#### Dependencies:

ADV\_FSP-L.1 Informal functional specification

ADV\_RCR-L.1 Informal correspondence demonstration

#### Developer action elements:

ADV\_HLD-L.2.1D The developer shall ~~provide the~~ produce a high-level design of the TSF that provides the information identified in the content and presentation section below.

ADV\_HLD-L.2.2D(+) The developer shall, as an essential part of the TOE development process, produce the high-level design as a precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.2.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_HLD-L.2.1C The presentation of the high-level design shall be informal.

ADV\_HLD-L.2.2C The high-level design shall be internally consistent.

ADV\_HLD-L.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-L.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-L.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-L.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-L.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV\_HLD-L.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_HLD-L.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements:

ADV\_HLD-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, ~~that the information provided meets all requirements for content and presentation of evidence~~ documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-L.1.2E *placeholder for element appearing at higher component* ~~The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.~~

ADV\_HLD-L.2.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.2.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process maintains the high-level design to reflect the actual implementation.

#### **A.1.28 ADV\_HLD-R.2 Security enforcing high-level design**

Dependencies:

ADV\_FSP-R.1 Informal functional specification

ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_HLD-R.2.1D The developer shall ~~provide the~~ produce a high-level design of the TSF that provides the information identified in the content and presentation section below.

ADV\_HLD-R.2.2D(+) The developer shall, as an essential part of the TOE development process, produce the high-level design as a precursor for and input to the development of the TOE implementation.

ADV\_HLD-R.2.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_HLD-R.2.1C The presentation of the high-level design shall be informal.

ADV\_HLD-R.2.2C The high-level design shall be internally consistent.

ADV\_HLD-R.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-R.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-R.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-R.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-R.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV\_HLD-R.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_HLD-R.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements:

ADV\_HLD-R.2.1E The evaluator shall confirm that the ~~information provided meets all requirements for content and presentation of evidence~~ documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-R.2.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_HLD-R.2.3E(+) The evaluator shall confirm that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-R.2.4E(+) The evaluator shall confirm that the developer design process maintains the high-level design to reflect the actual implementation.

**A.1.29 ADV\_HLD-L.4 Semiformal or Informal high-level explanation**

Dependencies:

ADV\_FSP-L.1 Informal functional specification

ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:

ADV\_HLD-L.4.1D The developer shall ~~provide the~~ produce a high-level design of the TSF that provides the information identified in the content and presentation section below.

ADV\_HLD-L.4.2D(+) The developer shall, as an essential part of the TOE development process, produce the high-level design as a precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.4.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_HLD-L.4.1C The presentation of the high-level design shall be semiformal or informal.

ADV\_HLD-L.4.2C The high-level design shall be internally consistent.

ADV\_HLD-L.4.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-L.4.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-L.4.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-L.4.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-L.4.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV\_HLD-L.4.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_HLD-L.4.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

ADV\_HLD-L.4.10C The high-level design shall justify that the identified means of achieving separation, including any protection mechanisms, are sufficient to ensure a clear and effective separation of TSP-enforcing from non-TSP-enforcing functions.

ADV\_HLD-L.4.11C The high-level design shall justify that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design.

Evaluator action elements:

ADV\_HLD-L.4.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the ~~information provided meets all requirements for content and presentation of evidence~~ documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-L.4.2E *placeholder for element appearing at higher component* ~~The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.~~

ADV\_HLD-L.4.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-L.4.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process maintains the high-level design to reflect the actual implementation.

**A.1.30 ADV\_HLD-R.4 Semiformal or Informal high-level explanation**

Dependencies:

ADV\_FSP-R.1 Informal functional specification

ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_HLD-R.4.1D The developer shall provide the high-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_HLD-R.4.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a high-level design.

ADV\_HLD-R.4.3D(+) The developer shall, as an essential part of the TOE development process, maintain the high-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_HLD-R.4.1C The presentation of the high-level design shall be semiformal or informal.

ADV\_HLD-R.4.2C The high-level design shall be internally consistent.

ADV\_HLD-R.4.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV\_HLD-R.4.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV\_HLD-R.4.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV\_HLD-R.4.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV\_HLD-R.4.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV\_HLD-R.4.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_HLD-R.4.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

ADV\_HLD-R.4.10C The high-level design shall justify that the identified means of achieving separation, including any protection mechanisms, are sufficient to ensure a clear and effective separation of TSP-enforcing from non-TSP-enforcing functions.

ADV\_HLD-R.4.11C The high-level design shall justify that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design.

#### Evaluator action elements:

ADV\_HLD-R.4.1E The evaluator shall confirm that the ~~information provided meets all requirements for content and presentation of evidence~~ documented high-level design meets all requirements for content and presentation of evidence.

ADV\_HLD-R.4.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_HLD-R.4.3E(+) The evaluator shall confirm that the developer design process produces the high-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_HLD-R.4.4E(+) The evaluator shall confirm that the developer design process maintains the high-level design to reflect the actual implementation.

### A.1.31 ADV\_IMP-L.2 Implementation of the TSF

Application notes: The ~~ADV\_IMP-L.2.2E element defines a requirement that the evaluator determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements. This provides a direct correspondence between the TOE security functional requirements and the implementation representation, in addition to the pairwise correspondences required by the ADV\_RCR family. It is expected that the evaluator will use the evidence provided in ADV\_RCR as an input to making this determination.~~

#### Dependencies:

ADV\_LLD-L.1 Descriptive low-level design

ADV\_RCR-L.1 Informal correspondence demonstration

ALC\_TAT-L.1 Well-defined development tools

#### Developer action elements:

ADV\_IMP-L.2.1D The developer shall ~~provide the~~ produce an implementation representation for the entire TSF that unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

Content and presentation of evidence elements:

~~ADV\_IMP-L.2.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.~~

~~ADV\_IMP-L.2.2C The implementation representation shall be internally consistent.~~

~~ADV\_IMP-L.2.3C The implementation representation shall describe the relationships between all portions of the implementation.~~

None

Evaluator action elements:

~~ADV\_IMP-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the implementation representation unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation that the information provided meets all requirements for content and presentation of evidence.~~

~~ADV\_IMP-L.2.2E The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.~~

#### **A.1.32 ADV\_IMP-R.2 Implementation of the TSF**

Application notes: The ADV\_IMP-R.2.2E element defines a requirement that the evaluator determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements. This provides a direct correspondence between the TOE security functional requirements and the implementation representation, in addition to the pairwise correspondences required by the ADV\_RCR family. It is expected that the evaluator will use the evidence provided in ADV\_RCR as an input to making this determination.

Dependencies:

ADV\_LLD-R.1 Descriptive low-level design

ADV\_RCR-R.1 Informal correspondence demonstration

ALC\_TAT-R.1 Well-defined development tools

Developer action elements:

~~ADV\_IMP-R.2.1D The developer shall provide the produce an implementation representation for the entire TSF that unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.~~



Content and presentation of evidence elements:

~~ADV\_IMP-R.2.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.~~

~~ADV\_IMP-R.2.2C The implementation representation shall be internally consistent.~~

~~ADV\_IMP-R.2.3C The implementation representation shall describe the relationships between all portions of the implementation.~~

None

Evaluator action elements:

~~ADV\_IMP-R.2.1E The evaluator shall confirm that the implementation representation unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation that the information provided meets all requirements for content and presentation of evidence.~~

ADV\_IMP-R.2.2E The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.

### **A.1.33 ADV\_IMP-L.3 Structured Implementation of the TSF**

Application notes: ~~The ADV\_IMP-L.2.2E element defines a requirement that the evaluator determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements. This provides a direct correspondence between the TOE security functional requirements and the implementation representation, in addition to the pairwise correspondences required by the ADV\_RCR family. It is expected that the evaluator will use the evidence provided in ADV\_RCR as an input to making this determination.~~

Dependencies:

ADV\_LLD-L.1 Descriptive low-level design

ADV\_RCR-L.1 Informal correspondence demonstration

ALC\_TAT-L.1 Well-defined development tools

Developer action elements:

ADV\_IMP-L.3.1D The developer shall ~~provide the~~ produce an implementation representation for the entire TSF that unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-L.3.2D(+) The developer shall produce the implementation representation for the entire TSF such that this representation is structured into small and comprehensible sections.

Content and presentation of evidence elements:

~~ADV\_IMP-L.3.1C~~ The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

~~ADV\_IMP-L.3.2C~~ The implementation representation shall be internally consistent.

~~ADV\_IMP-L.2.3C~~ The implementation representation shall describe the relationships between all portions of the implementation.

~~ADV\_IMP-L.3.4C~~ The implementation representation shall be structured into small and comprehensible sections.

None

Evaluator action elements:

ADV\_IMP-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the implementation representation unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation that the information provided meets all requirements for content and presentation of evidence.

~~ADV\_IMP-L.2.2E~~ *placeholder for element appearing at higher component*—The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_IMP-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the implementation representation is structured into small and comprehensible sections.

#### **A.1.34 ADV\_IMP-R.3 Structured Implementation of the TSF**

Application notes: The ADV\_IMP-R.2.2E element defines a requirement that the evaluator determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements. This provides a direct correspondence between the TOE security functional requirements and the implementation representation, in addition to the pairwise correspondences required by the ADV\_RCR family. It is expected that the evaluator will use the evidence provided in ADV\_RCR as an input to making this determination.

Dependencies:

ADV\_LLD-R.1 Descriptive low-level design

ADV\_RCR-R.1 Informal correspondence demonstration  
ALC\_TAT-R.1 Well-defined development tools

Developer action elements:

ADV\_IMP-R.3.1D ~~The developer shall provide the~~ produce an implementation representation for the entire TSF that unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation.

ADV\_IMP-R.3.2D(+) The developer shall produce the implementation representation for the entire TSF such that this representation is structured into small and comprehensible sections.

Content and presentation of evidence elements:

~~ADV\_IMP-R.3.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.~~

~~ADV\_IMP-R.3.2C The implementation representation shall be internally consistent.~~

~~ADV\_IMP-R.3.3C The implementation representation shall describe the relationships between all portions of the implementation.~~

~~ADV\_IMP-R.3.4C The implementation representation shall be structured into small and comprehensible sections.~~

None

Evaluator action elements:

ADV\_IMP-R.3.1E The evaluator shall confirm that the implementation representation unambiguously defines the TSF to a level of detail such that the TSF can be generated without further design decisions, is internally consistent, and describes the relationships between all portions of the implementation that the information provided meets all requirements for content and presentation of evidence.

ADV\_IMP-R.3.2E The evaluator shall determine that the implementation representation is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_IMP-R.3.3E(+) The evaluator shall confirm that the implementation representation is structured into small and comprehensible sections.

### A.1.35 ADV\_INT-L.1 Modularity

Dependencies:

ADV\_IMP-L.1 Subset of the implementation of the TSF  
ADV\_LLD-L.1 Descriptive low-level design

Developer action elements:

ADV\_INT-L.1.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

~~ADV\_INT-L.1.2D The developer shall provide an architectural description.~~

Content and presentation of evidence elements:

~~ADV\_INT-L.1.1C The architectural description shall identify the modules of the TSF.~~

~~ADV\_INT-L.1.2C The architectural description shall describe the purpose, interface, parameters, and effects of each module of the TSF.~~

~~ADV\_INT-L.1.3C The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions.~~

None

Evaluator action elements:

~~ADV\_INT-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed, structured, and implemented the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design that the information provided meets all requirements for content and presentation of evidence.~~

~~ADV\_INT-L.1.2E The evaluator shall determine that both the low-level design and the implementation representation are in compliance with the architectural description.~~

### **A.1.36 ADV\_INT-R.1 Modularity**

Dependencies:

ADV\_IMP-R.1 Subset of the implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design

Developer action elements:

ADV\_INT-R.1.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

~~ADV\_INT-R.1.2D The developer shall provide an architectural description.~~

Content and presentation of evidence elements:

~~ADV\_INT-R.1.1C The architectural description shall identify the modules of the TSF.~~

~~ADV\_INT-R.1.2C The architectural description shall describe the purpose, interface, parameters, and effects of each module of the TSF.~~

~~ADV\_INT-R.1.3C The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions.~~

None

Evaluator action elements:

~~ADV\_INT-R.1.1E The evaluator shall confirm determine that the developer designed, structured, and implemented the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design that the information provided meets all requirements for content and presentation of evidence.~~

~~ADV\_INT-R.1.2E The evaluator shall determine that both the low-level design and the implementation representation are in compliance with the architectural description.~~

### **A.1.37 ADV\_INT-L.3 Minimization of complexity**

Application notes: This component requires that the reference monitor property "simple enough to be analyzed" is fully addressed. When this component is combined with the functional requirements FPT\_RVM.1 and FPT\_SEP.3, the reference monitor concept would be fully realized.

Dependencies:

ADV\_IMP-L.2 Implementation of the TSF

ADV\_LLD-L.1 Descriptive low-level design

Developer action elements:

ADV\_INT-L.3.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

~~ADV\_INT-L.3.2D CC element deleted The developer shall provide an architectural description.~~

ADV\_INT-L.3.3D The developer shall design and structure the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-L.3.4D The developer shall design and structure the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-L.3.5D The developer shall design and structure the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-L.3.6D The developer shall ensure that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

Content and presentation of evidence elements:

~~ADV\_INT-L.3.1C The architectural description shall identify the modules of the TSF and shall specify which portions of the TSF enforce the access control and/or information flow control policies.~~

~~ADV\_INT-L.3.2C The architectural description shall describe the purpose, interface, parameters, and side effects of each module of the TSF.~~

~~ADV\_INT-L.3.3C The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions.~~

~~ADV\_INT-L.3.4C The architectural description shall describe the layering architecture.~~

~~ADV\_INT-L.3.5C The architectural description shall show that mutual interactions have been minimized, and justify those that remain.~~

~~ADV\_INT-L.3.6C The architectural description shall describe how the entire TSF has been structured to minimize complexity.~~

~~ADV\_INT-L.3.7C The architectural description shall justify the inclusion of any non-TSP-enforcing modules in the TSF.~~

None

Evaluator action elements:

~~ADV\_INT-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed, structured, and implemented the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design that the information provided meets all requirements for content and presentation of evidence.~~

~~ADV\_INT-L.3.2E Intent of CC element incorporated into \*.1E above The evaluator shall determine that both the low-level design and the implementation representation are in compliance with the architectural description.~~

ADV\_INT-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed and structured the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-L.3.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed and structured the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-L.3.5E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer designed and structured the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-L.3.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer ensured that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

### **A.1.38 ADV\_INT-R.3 Minimization of complexity**

Application notes: This component requires that the reference monitor property "simple enough to be analyzed" is fully addressed. When this component is combined with the functional requirements FPT\_RVM.1 and FPT\_SEP.3, the reference monitor concept would be fully realized.

Dependencies:

ADV\_IMP-R.2 Implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design

Developer action elements:

ADV\_INT-R.3.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

~~ADV\_INT-R.3.2D CC element deleted The developer shall provide an architectural description.~~

ADV\_INT-R.3.3D The developer shall design and structure the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-R.3.4D The developer shall design and structure the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-R.3.5D The developer shall design and structure the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-R.3.6D The developer shall ensure that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

Content and presentation of evidence elements:

~~ADV\_INT-R.3.1C The architectural description shall identify the modules of the TSF and shall specify which portions of the TSF enforce the access control and/or information flow control policies.~~

~~ADV\_INT-R.3.2C The architectural description shall describe the purpose, interface, parameters, and side effects of each module of the TSF.~~

~~ADV\_INT-R.3.3C The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions.~~

~~ADV\_INT-R.3.4C The architectural description shall describe the layering architecture.~~

~~ADV\_INT-R.3.5C The architectural description shall show that mutual interactions have been minimized, and justify those that remain.~~

~~ADV\_INT-R.3.6C The architectural description shall describe how the entire TSF has been structured to minimize complexity.~~

~~ADV\_INT-R.3.7C The architectural description shall justify the inclusion of any non-TSP-enforcing modules in the TSF.~~

None

Evaluator action elements:

~~ADV\_INT-R.3.1E The evaluator shall confirm determine that the developer designed and structured, and implemented the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design that the information provided meets all requirements for content and presentation of evidence.~~

~~ADV\_INT-R.3.2E *Intent of CC element incorporated into \*.1E above* The evaluator shall determine that both the low-level design and the implementation representation are in compliance with the architectural description.~~

ADV\_INT-R.3.3E(+) The evaluator shall determine that the developer designed and structured the TSF in a layered fashion that minimizes mutual interactions between the layers of the design.

ADV\_INT-R.3.4E(+) The evaluator shall determine that the developer designed and structured the TSF in such a way that minimizes the complexity of the entire TSF.

ADV\_INT-R.3.5E(+) The evaluator shall determine that the developer designed and structured the portions of the TSF that enforce any access control and/or information flow control policies such that they are simple enough to be analyzed.

ADV\_INT-R.3.6E(+) The evaluator shall determine that the developer ensured that functions whose objectives are not relevant for the TSF are excluded from the TSF modules.

### **A.1.39 ADV\_LLD-L.1 Descriptive low-level design**

Dependencies:

ADV\_HLD-L.2 Security enforcing high-level design

ADV\_RCR-L.1 Informal correspondence demonstration

Developer action elements:



ADV\_LLD-L.1.1D The developer shall provide the low-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_LLD-L.1.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a low-level design.

ADV\_LLD-L.1.3D(+) The developer shall, as an essential part of the TOE development process, maintain the low-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_LLD-L.1.1C The presentation of the low-level design shall be informal.

ADV\_LLD-L.1.2C The low-level design shall be internally consistent.

ADV\_LLD-L.1.3C The low-level design shall describe the TSF in terms of modules.

ADV\_LLD-L.1.4C The low-level design shall describe the purpose of each module.

ADV\_LLD-L.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV\_LLD-L.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV\_LLD-L.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV\_LLD-L.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV\_LLD-L.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_LLD-L.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

ADV\_LLD-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the documented low-level design meets all requirements for content and presentation of evidence.

~~ADV\_LLD-L.1.2E CC element deleted The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.~~

ADV\_LLD-L.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process produces the low-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_LLD-L.1.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer design process maintains the low-level design to reflect the actual implementation.

#### **A.1.40 ADV\_LLD-R.1 Descriptive low-level design**

Dependencies:

ADV\_HLD-R.2 Security enforcing high-level design  
ADV\_RCR-R.1 Informal correspondence demonstration

Developer action elements:

ADV\_LLD-R.1.1D The developer shall provide the low-level design of the TSF that represents the current TOE implementation and provides the information identified in the content and presentation section below.

ADV\_LLD-R.1.2D(+) The developer shall, as an essential precursor for and input to the development of the TOE implementation, produce a low-level design.

ADV\_LLD-R.1.3D(+) The developer shall, as an essential part of the TOE development process, maintain the low-level design to reflect the actual implementation.

Content and presentation of evidence elements:

ADV\_LLD-R.1.1C The presentation of the low-level design shall be informal.

ADV\_LLD-R.1.2C The low-level design shall be internally consistent.

ADV\_LLD-R.1.3C The low-level design shall describe the TSF in terms of modules.

ADV\_LLD-R.1.4C The low-level design shall describe the purpose of each module.

ADV\_LLD-R.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV\_LLD-R.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV\_LLD-R.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV\_LLD-R.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV\_LLD-R.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV\_LLD-R.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

ADV\_LLD-R.1.1E The evaluator shall confirm that the documented low-level design meets all requirements for content and presentation of evidence.

ADV\_LLD-R.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV\_LLD-R.1.3E(+) The evaluator shall confirm that the developer design process produces the low-level design as an essential precursor for and input to the development of the TOE implementation.

ADV\_LLD-R.1.4E(+) The evaluator shall confirm that the developer design process maintains the low-level design to reflect the actual implementation.

**A.1.41 ADV\_RCR-L.1 Informal correspondence demonstration**

Dependencies: No dependencies.

Developer action elements:

ADV\_RCR-L.1.1D The developer shall ~~provide an~~ conduct an informal analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Content and presentation of evidence elements:

~~ADV\_RCR-L.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.~~

None

Evaluator action elements:

ADV\_RCR-L.1.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the ~~information provided meets all requirements for content and presentation of evidence~~ results of the developer analysis of correspondence show that the relevant security

functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

#### **A.1.42 ADV\_RCR-R.1 Informal correspondence demonstration**

Dependencies: No dependencies.

Developer action elements:

ADV\_RCR-R.1.1D The developer shall ~~provide an~~ conduct an analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Content and presentation of evidence elements:

~~ADV\_RCR-L.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.~~

None

Evaluator action elements:

ADV\_RCR-R.1.1E The evaluator shall confirm that the ~~information provided meets all requirements for content and presentation of evidence~~ results of the developer analysis of correspondence show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

#### **A.1.43 ADV\_RCR-L.3 Formal correspondence demonstration**

Dependencies: No dependencies.

Developer action elements:

ADV\_RCR-L.3.1D The developer shall ~~provide an~~ conduct an analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV\_RCR-L.3.2D(+) The developer shall conduct a formal proof of correspondence between adjacent pair of TSF representations whenever both representations are formally specified.

Content and presentation of evidence elements:

~~ADV\_RCR-L.3.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.~~

~~ADV\_RCR-L.3.2C For each adjacent pair of provided TSF representations, where portions of one representation are semiformally specified and the other at least semiformally specified, the demonstration of correspondence between those portions of the representations shall be semiformal.~~

~~ADV\_RCR-L.3.3C For each adjacent pair of provided TSF representations, where portions of both representations are formally specified, the proof of correspondence between those portions of the representations shall be formal.~~

None

Evaluator action elements:

~~ADV\_RCR-L.3.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the information provided meets all requirements for content and presentation of evidence results of the developer analysis of correspondence show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.~~

~~ADV\_RCR-L.3.2E CC element deleted The evaluator shall determine the accuracy of the proofs of correspondence by selectively verifying the formal analysis.~~

ADV\_RCR-L.3.3E(+) The evaluator shall check that the developer produced a formal proof of correspondence between adjacent pair of TSF representations whenever both representations are formally specified.

#### **A.1.44 ADV\_RCR-R.3 Formal correspondence demonstration**

Dependencies: No dependencies.

Developer action elements:

ADV\_RCR-R.3.1D The developer shall provide an conduct an analysis of correspondence between all adjacent pairs of TSF representations that are provided to verify that relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

ADV\_RCR-R.3.2D(+) The developer shall conduct a formal proof of correspondence between adjacent pair of TSF representations whenever both representations are formally specified.

Content and presentation of evidence elements:

~~ADV\_RCR-R.3.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.~~

~~ADV\_RCR-R.3.2C For each adjacent pair of provided TSF representations, where portions of one representation are semiformaly specified and the other at least semiformaly specified, the demonstration of correspondence between those portions of the representations shall be semiformal.~~

~~ADV\_RCR-R.3.3C For each adjacent pair of provided TSF representations, where portions of both representations are formally specified, the proof of correspondence between those portions of the representations shall be formal.~~

None

Evaluator action elements:

~~ADV\_RCR-R.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence results of the developer analysis of correspondence show that the relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.~~

~~ADV\_RCR-R.3.2E The evaluator shall determine the accuracy of the proofs of correspondence by selectively verifying the formal analysis.~~

~~ADV\_RCR-R.3.3E(+) The evaluator shall check that the developer produced a formal proof of correspondence between adjacent pair of TSF representations whenever both representations are formally specified.~~

#### **A.1.45 ADV\_SPM-L.1 Informal TOE security policy model**

Dependencies: ADV\_FSP-L.1 Informal functional specification

Developer action elements:

~~ADV\_SPM-L.1.1D The developer shall provide produce an informal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.~~

~~ADV\_SPM-L.1.2D The developer shall demonstrate show correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.~~

Content and presentation of evidence elements:

~~ADV\_SPM-L.1.1C The TSP model shall be informal.~~

~~ADV\_SPM-L.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.~~

~~ADV\_SPM-L.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.~~

~~ADV\_SPM-L.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.~~

None

Evaluator action elements:

ADV\_SPM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the information provided meets all requirements for content and presentation of evidence SPM is an informal TSP model that describes the rules and characteristics of the policies of the TSP.

ADV\_SPM-L.1.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has shown correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

#### **A.1.46 ADV\_SPM-R.1 Informal TOE security policy model**

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

ADV\_SPM-R.1.1D The developer shall provide produce an informal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.

ADV\_SPM-R.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

Content and presentation of evidence elements:

~~ADV\_SPM-R.1.1C The TSP model shall be informal.~~

~~ADV\_SPM-R.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.~~

~~ADV\_SPM-R.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.~~

~~ADV\_SPM-R.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.~~

None

### Evaluator action elements:

~~ADV\_SPM-R.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ SPM is an informal TSP model and describes the rules and characteristics of the policies of the TSP.

~~ADV\_SPM-R.1.2E(+) The evaluator shall confirm that the developer has demonstrated correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.~~

### A.1.47 ADV\_SPM-L.3 Formal TOE security policy model

Dependencies: ADV\_FSP-L.1 Informal functional specification

### Developer action elements:

~~ADV\_SPM-L.3.1D The developer shall provide produce a formal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.~~

~~ADV\_SPM-L.3.2D The developer shall demonstrate or prove, as appropriate, correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.~~

### Content and presentation of evidence elements:

~~ADV\_SPM-L.3.1C The TSP model shall be informal.~~

~~ADV\_SPM-L.3.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.~~

~~ADV\_SPM-L.3.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.~~

~~ADV\_SPM-L.3.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.~~

~~ADV\_SPM-L.3.5C Where the functional specification is semiformal, the demonstration of correspondence between the TSP model and the functional specification shall be semiformal.~~

~~ADV\_SPM-L.3.6C Where the functional specification is formal, the proof of correspondence between the TSP model and the functional specification shall be formal.~~

None

### Evaluator action elements:



~~ADV\_SPM-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the information provided meets all requirements for content and presentation of evidence SPM a formal TSP model that describes the rules and characteristics of the policies of the TSP.~~

~~ADV\_SPM-L.3.2E(+) The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer has proven correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.~~

#### **A.1.48 ADV\_SPM-R.3 Formal TOE security policy model**

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

~~ADV\_SPM-R.1.1D The developer shall provide produce a formal TSP model (SPM) describing the rules and characteristics of the policies of the TSP.~~

~~ADV\_SPM-R.1.2D The developer shall demonstrate or prove, as appropriate, correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.~~

Content and presentation of evidence elements:

~~ADV\_SPM-R.3.1C The TSP model shall be informal.~~

~~ADV\_SPM-R.3.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.~~

~~ADV\_SPM-R.3.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.~~

~~ADV\_SPM-R.3.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.~~

~~ADV\_SPM-R.3.5C Where the functional specification is semiformal, the demonstration of correspondence between the TSP model and the functional specification shall be semiformal.~~

~~ADV\_SPM-R.3.6C Where the functional specification is formal, the proof of correspondence between the TSP model and the functional specification shall be formal.~~

None

Evaluator action elements:

ADV\_SPM-R.3.1E The evaluator shall confirm that the ~~information provided meets all requirements for content and presentation of evidence~~ SPM a formal TSP model that describes the rules and characteristics of the policies of the TSP.

ADV\_SPM-R.3.2E(+) The evaluator shall confirm that the developer has proven correspondence between the functional specification and the TSP model and that the security functions in the functional specification are consistent and complete with respect to the TSP model.

## A.4 GUIDANCE DOCUMENTATION

### A.1.49 AGD\_ADM-L.1 Administrator guidance

Dependencies: ADV\_FSP.1 Informal functional specification None

Developer action elements:

AGD\_ADM-L.1.1D The developer shall provide administrator guidance addressed to system Administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_ADM-L.1.1C The administrator guidance shall describe the Administrative functions and interfaces available to the administrator of the TOE.

AGD\_ADM-L.1.2C The administrator guidance shall describe how to Administer the TOE in a secure manner.

AGD\_ADM-L.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-L.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-L.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-L.1.6C The administrator guidance shall describe each type of security-relevant event relative to the Administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-L.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-L.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD\_ADM-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE that the information provided meets all requirements for content and presentation of evidence.

**A.1.50 AGD\_ADM-R.1 Administrator guidance**

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

AGD\_ADM-R.1.1D The developer shall provide administrator guidance addressed to system administrative personnel providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_ADM-R.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD\_ADM-R.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD\_ADM-R.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD\_ADM-R.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD\_ADM-R.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD\_ADM-R.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD\_ADM-R.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_ADM-R.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD\_ADM-R.1.1E The evaluator shall confirm that this guidance contains no errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE ~~that the information provided meets all requirements for content and presentation of evidence.~~

**A.1.51 AGD\_USR-L.1 User guidance**

Dependencies: ~~ADV\_FSP.1 Informal functional specification~~ None

Developer action elements:

AGD\_USR-L.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_USR-L.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-L.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-L.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-L.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-L.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-L.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD\_USR-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that this guidance contains no obvious errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable con-

figuring and operating the TOE in a manner consistent with the security claims made about the TOE ~~that the information provided meets all requirements for content and presentation of evidence.~~

#### **A.1.52 AGD\_USR-R.1 User guidance**

Dependencies: ADV\_FSP-R.1 Informal functional specification

Developer action elements:

AGD\_USR-R.1.1D The developer shall provide user guidance providing the information identified in the content and presentation section below in a manner that is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE.

Content and presentation of evidence elements:

AGD\_USR-R.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD\_USR-R.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD\_USR-R.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD\_USR-R.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD\_USR-R.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD\_USR-R.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD\_USR-R.1.1E The evaluator shall confirm that this guidance contains no errors, is suitable to conduct the other evaluator actions to be performed as a part of the evaluation of the TOE, and is sufficient to enable configuring and operating the TOE in a manner consistent with the security claims made about the TOE ~~that the information provided meets all requirements for content and presentation of evidence.~~

### **A.5 LIFE CYCLE SUPPORT**

#### **A.1.53 ALC\_DVS-L.1 Identification of security measures**

Dependencies: No dependencies.

Developer action elements:

ALC\_DVS-L.1.1D The developer shall produce development security documentation identify the physical, procedural, personnel, and other security measures that are deemed necessary by the developer to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-L.1.2D(+) The developer shall implement the measures identified.

Content and presentation of evidence elements:

~~ALC\_DVS-L.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.~~

~~ALC\_DVS-L.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.~~

None

Evaluator action elements:

ALC\_DVS-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the information provided meets all requirements for content and presentation of evidence the developer has identified the physical, procedural, personnel, and other security measures that are deemed necessary by the developer to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-L.1.2E The evaluator shall confirm, to the level of rigor of appears to be true, that the security measures are being applied.

**A.1.54 ALC\_DVS-R.1 Identification of security measures**

Dependencies: No dependencies.

Developer action elements:

ALC\_DVS-R.1.1D The developer shall produce development security documentation identify the physical, procedural, personnel, and other security measures that are deemed necessary by the developer to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC\_DVS-R.1.2D(+) The developer shall implement the measures identified.

Content and presentation of evidence elements:

~~ALC\_DVS-R.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.~~

~~ALC\_DVS-R.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.~~

None

Evaluator action elements:

~~ALC\_DVS-R.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ the developer has identified the physical, procedural, personnel, and other security measures that are deemed necessary by the developer to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

~~ALC\_DVS-R.1.2E The evaluator shall determine that the security measures are being applied.~~

#### **A.1.55 ALC\_DVS-L.2 Sufficiency of security measures**

Dependencies: No dependencies.

Developer action elements:

~~ALC\_DVS-L.2.1D The developer shall produce development security documentation identify the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.~~

ALC\_DVS-L.2.2D(+) The developer shall implement the measures identified.

Content and presentation of evidence elements:

~~ALC\_DVS-L.2.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.~~

~~ALC\_DVS-L.2.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.~~

~~ALC\_DVS-L.2.3C The evidence shall justify that the security measures provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.~~

None

Evaluator action elements:

~~ALC\_DVS-L.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ developer has identified the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

~~ALC\_DVS-L.2.2E The evaluator shall confirm, to the level of rigor of appears to be true,~~  
that the security measures are being applied.

#### **A.1.56 ALC\_DVS-R.2 Sufficiency of security measures**

Dependencies: No dependencies.

Developer action elements:

~~ALC\_DVS-R.2.1D The developer shall produce development security documentation identify the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.~~

~~ALC\_DVS-R.2.2D(+) The developer shall implement the measures identified.~~

Content and presentation of evidence elements:

~~ALC\_DVS-R.2.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.~~

~~ALC\_DVS-R.2.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.~~

~~ALC\_DVS-R.2.3C The evidence shall justify that the security measures provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.~~

None

Evaluator action elements:

~~ALC\_DVS-R.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ developer has identified the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.



ALC\_DVS-R.2.2E The evaluator shall confirm that the security measures are being applied.

#### **A.1.57 ALC\_FLR-L.2 Flaw reporting procedures**

Dependencies: No dependencies.

Developer action elements:

ALC\_FLR-L.2.1D The developer shall ~~document~~ establish flaw remediation procedures that provide the capabilities included in the content and presentation section below.

ALC\_FLR-L.2.2D The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

Content and presentation of evidence elements:

ALC\_FLR-L.2.1C The flaw remediation procedures ~~documentation~~ shall ~~describe the procedures used to track~~ include tracking of all reported security flaws in each release of the TOE.

ALC\_FLR-L.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC\_FLR-L.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC\_FLR-L.2.4C The flaw remediation procedures ~~documentation~~ shall ~~describe the methods used to provide~~ include providing flaw information, corrections and guidance on corrective actions to TOE users.

ALC\_FLR-L.2.5C The procedures for processing reported security flaws shall include measures to ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC\_FLR-L.2.6C The procedures for processing reported security flaws shall include safeguards to reduce the potential for ~~that any~~ corrections to these security flaws to ~~do not~~ introduce ~~any~~ new flaws.

Evaluator action elements:

ALC\_FLR-L.2.1E The evaluator shall confirm, to the level of rigor of appears true, that the flaw remediation procedures meet all the requirements identified in the content and presentation of evidence section above.

ALC\_FLR-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears true, that the flaw remediation procedures include provisions for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

#### **A.1.58 ALC\_FLR-R.2 Flaw reporting procedures**

Dependencies: No dependencies.

Developer action elements:

ALC\_FLR-R.2.1D The developer shall ~~document~~ establish flaw remediation procedures that provide the capabilities included in the content and presentation section below.

ALC\_FLR-R.2.2D The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

Content and presentation of evidence elements:

ALC\_FLR-R.2.1C The flaw remediation procedures ~~documentation~~ shall ~~describe the procedures used to track~~ include tracking of all reported security flaws in each release of the TOE.

ALC\_FLR-R.2.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC\_FLR-R.2.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC\_FLR-R.2.4C The flaw remediation procedures ~~documentation~~ shall ~~describe the methods used to provide~~ include providing flaw information, corrections and guidance on corrective actions to TOE users.

ALC\_FLR-R.2.5C The procedures for processing reported security flaws shall include measures to ensure that any reported flaws are corrected and the correction issued to TOE users.

ALC\_FLR-R.2.6C The procedures for processing reported security flaws shall include safeguards to reduce the potential for ~~that any~~ corrections to these security flaws to ~~do not~~ introduce ~~any~~ new flaws.

Evaluator action elements:

ALC\_FLR-R.2.1E The evaluator shall confirm that the flaw remediation procedures meet all the requirements identified in the content and presentation of evidence section above.

ALC\_FLR-R.2.2E(+) The evaluator shall confirm that the flaw remediation procedures include provisions for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.

#### **A.1.59 ALC\_LCD-L.1 Developer defined life-cycle model**

Dependencies: No dependencies.

Developer action elements:

ALC\_LCD-L.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

~~ALC\_LCD-L.1.2D The developer shall provide life-cycle definition documentation.~~

Content and presentation of evidence elements:

~~ALC\_LCD-L.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.~~

~~ALC\_LCD-L.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.~~

None

Evaluator action elements:

ALC\_LCD-L.1.1E The evaluator shall confirm, to the level of rigor of appears true, that the information provided meets all requirements for content and presentation of evidence developer has established a life-cycle model to be used in the development and maintenance of the TOE.

#### **A.1.60 ALC\_LCD-R.1 Developer defined life-cycle model**

Dependencies: No dependencies.

Developer action elements:

ALC\_LCD-R.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

~~ALC\_LCD-R.1.2D The developer shall provide life-cycle definition documentation.~~

Content and presentation of evidence elements:

~~ALC\_LCD-R.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.~~

~~ALC\_LCD-R.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.~~

None

Evaluator action elements:

~~ALC\_LCD-R.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ determine that the developer has established a life-cycle model to be used in the development and maintenance of the TOE.

#### **A.1.61 ALC\_LCD-L.2 Standardized life-cycle model**

Dependencies: No dependencies.

Developer action elements:

ALC\_LCD-L.2.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

~~ALC\_LCD-L.2.2D CC element deleted The developer shall provide life cycle definition documentation.~~

ALC\_LCD-L.2.3D The developer shall use a standardized life-cycle model to develop and maintain the TOE.

Content and presentation of evidence elements:

~~ALC\_LCD-L.2.1C The life cycle definition documentation shall describe the model used to develop and maintain the TOE.~~

~~ALC\_LCD-L.2.2C The life cycle model shall provide for the necessary control over the development and maintenance of the TOE.~~

~~ALC\_LCD-L.2.3C The life cycle definition documentation shall explain why the model was chosen.~~

~~ALC\_LCD-L.2.4C The life cycle definition documentation shall explain how the model is used to develop and maintain the TOE.~~

~~ALC\_LCD-L.2.5C The life cycle definition documentation shall demonstrate compliance with the standardized life cycle model.~~

None

Evaluator action elements:

ALC\_LCD-L.2.1E The evaluator shall confirm to the level or rigor of appears to be true, ~~that the information provided meets all requirements for content and presentation of evi-~~

ence developer is using a standardized life-cycle model for the development and maintenance of the TOE.

#### **A.1.62 ALC\_LCD-R.2 Standardized life-cycle model**

Dependencies: No dependencies.

Developer action elements:

ALC\_LCD-R.2.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

~~ALC\_LCD-R.2.2D CC element deleted The developer shall provide life cycle definition documentation.~~

ALC\_LCD-R.2.3D The developer shall use a standardized life-cycle model to develop and maintain the TOE.

Content and presentation of evidence elements:

~~ALC\_LCD-R.2.1C The life cycle definition documentation shall describe the model used to develop and maintain the TOE.~~

~~ALC\_LCD-R.2.2C The life cycle model shall provide for the necessary control over the development and maintenance of the TOE.~~

~~ALC\_LCD-R.2.3C The life cycle definition documentation shall explain why the model was chosen.~~

~~ALC\_LCD-R.2.4C The life cycle definition documentation shall explain how the model is used to develop and maintain the TOE.~~

~~ALC\_LCD-R.2.5C The life cycle definition documentation shall demonstrate compliance with the standardized life cycle model.~~

None

Evaluator action elements:

~~ALC\_LCD-R.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence~~ developer is using a standardized life-cycle model for the development and maintenance of the TOE.

#### **A.1.63 ALC\_LCD-L.3 Measurable life-cycle model**

Dependencies: No dependencies.

Developer action elements:

ALC\_LCD-L.3.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC\_LCD-L.3.2D *CC element deleted* ~~The developer shall provide life-cycle definition documentation.~~

ALC\_LCD-L.3.3D The developer shall use a standardized and measurable life-cycle model to develop and maintain the TOE.

ALC\_LCD-L.3.4D The developer shall measure the TOE development using the standardized and measurable life-cycle model.

Content and presentation of evidence elements:

~~ALC\_LCD-L.3.1C The life cycle definition documentation shall describe the model used to develop and maintain the TOE, including the details of its arithmetic parameters and/or metrics used to measure the TOE development against the model.~~

~~ALC\_LCD-L.3.2C The life cycle model shall provide for the necessary control over the development and maintenance of the TOE.~~

~~ALC\_LCD-L.3.3C The life cycle definition documentation shall explain why the model was chosen.~~

~~ALC\_LCD-L.3.4C The life cycle definition documentation shall explain how the model is used to develop and maintain the TOE.~~

~~ALC\_LCD-L.3.5C The life cycle definition documentation shall demonstrate compliance with the standardized and measurable life cycle model.~~

~~ALC\_LCD-L.3.6C The life cycle documentation shall provide the results of the measurements of the TOE development using the standardized and measurable life cycle model.~~

None

Evaluator action elements:

ALC\_LCD-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the information provided meets all requirements for content and presentation of evidence developer is using a standardized and measurable life-cycle model for the development and maintenance of the TOE.

ALC\_LCD-L.3.2E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has measured the TOE development using the standardized and measurable life-cycle model.

#### **A.1.64 ALC\_LCD-R.3 Measurable life-cycle model**

Dependencies: No dependencies.

## Developer action elements:

ALC\_LCD-R.3.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC\_LCD-R.3.2D *CC element deleted* ~~The developer shall provide life-cycle definition documentation.~~

ALC\_LCD-R.3.3D The developer shall use a standardized and measurable life-cycle model to develop and maintain the TOE.

ALC\_LCD-R.3.4D The developer shall measure the TOE development using the standardized and measurable life-cycle model.

## Content and presentation of evidence elements:

~~ALC\_LCD-R.3.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE, including the details of its arithmetic parameters and/or metrics used to measure the TOE development against the model.~~

~~ALC\_LCD-R.3.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.~~

~~ALC\_LCD-R.3.3C The life-cycle definition documentation shall explain why the model was chosen.~~

~~ALC\_LCD-R.3.4C The life-cycle definition documentation shall explain how the model is used to develop and maintain the TOE.~~

~~ALC\_LCD-R.3.5C The life-cycle definition documentation shall demonstrate compliance with the standardized and measurable life-cycle model.~~

~~ALC\_LCD-R.3.6C The life-cycle documentation shall provide the results of the measurements of the TOE development using the standardized and measurable life-cycle model.~~

None

## Evaluator action elements:

ALC\_LCD-R.3.1E The evaluator shall confirm that the developer is using a standardized and measurable life-cycle model for the development and maintenance of the TOE.

ALC\_LCD-R.3.2E(+) The evaluator shall confirm that the developer has measured the TOE development using the standardized and measurable life-cycle model.

**A.1.65 ALC\_TAT-L.1 Well-defined development tools**

Dependencies: ~~ADV\_IMP.1 Subset of the implementation of the TSF~~ None

Developer action elements:

ALC\_TAT-L.1.1D The developer shall ~~identify the~~ use well-defined development tools ~~being used~~ for the TOE.

ALC\_TAT-L.1.2D The developer shall ~~document~~ identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.1.3D(+) placeholder for element appearing in higher component

ALC\_TAT-L.1.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

~~ALC\_TAT-L.1.1C All development tools used for implementation shall be well defined.~~

~~ALC\_TAT-L.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.~~

~~ALC\_TAT-L.1.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.~~

None

Evaluator action elements:

ALC\_TAT-L.1.1E The evaluator shall ~~confirm, to the level of rigor of appears to be true, that the information provided meets all requirements for content and presentation of evidence~~ developer is using well-defined development tools for the TOE.

ALC\_TAT-L.1.2E(+) placeholder for element appearing in higher component

ALC\_TAT-L.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.1.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.



**A.1.66 ALC\_TAT-L.2 Compliance with implementation standards**

Dependencies: ADV\_IMP-L.1 Subset of the implementation of the TSF

Developer action elements:

ALC\_TAT-L.2.1D The developer shall ~~identify the~~ use well-defined development tools ~~being used~~ for the TOE.

ALC\_TAT-L.2.2D The developer shall ~~document~~ identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.2.3D The developer shall ~~describe~~ apply identified implementation standards ~~to be applied.~~

ALC\_TAT-L.2.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

~~ALC\_TAT-L.2.1C All development tools used for implementation shall be well-defined.~~

~~ALC\_TAT-L.2.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.~~

~~ALC\_TAT-L.2.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.~~

~~ALC\_TAT-L.2.3D The developer shall describe the implementation standards to be applied.~~

None

Evaluator action elements:

ALC\_TAT-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the ~~information provided meets all requirements for content and presentation of evidence~~ developer is using well-defined development tools for the TOE.

ALC\_TAT-L.2.2E The evaluator shall confirm, to the level of rigor of appears to be true, that the identified implementation standards have been applied.

ALC\_TAT-L.2.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.2.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### **A.1.67 ALC\_TAT-R.2 Compliance with implementation standards**

Dependencies: ADV\_IMP-R.1 Subset of the implementation of the TSF

Developer action elements:

~~ALC\_TAT-R.2.1D~~ The developer shall ~~identify the~~ use well-defined development tools ~~being used~~ for the TOE.

~~ALC\_TAT-R.2.2D~~ The developer shall ~~document~~ identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

~~ALC\_TAT-R.2.3D~~ The developer shall ~~describe~~ apply identified implementation standards ~~to be applied.~~

ALC\_TAT-R.2.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

~~ALC\_TAT-R.2.1C~~ All development tools used for implementation shall be well-defined.

~~ALC\_TAT-R.2.2C~~ The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

~~ALC\_TAT-R.2.3C~~ The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

~~ALC\_TAT-R.2.3D~~ The developer shall describe the implementation standards to be applied.

None

Evaluator action elements:

~~ALC\_TAT-R.2.1E~~ The evaluator shall confirm that the ~~information provided meets all requirements for content and presentation of evidence~~ developer is using well-defined development tools for the TOE.

~~ALC\_TAT-R.2.2E~~ The evaluator shall confirm that the identified implementation standards have been applied.

ALC\_TAT-R.2.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the implementation-dependent options of the devel-

opment tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-R.2.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### **A.1.68 ALC\_TAT-L.3 Compliance with implementation standards - all parts**

Dependencies: ADV\_IMP-L.1 Subset of the implementation of the TSF

Developer action elements:

~~ALC\_TAT-L.3.1D The developer shall identify the~~ use well-defined development tools ~~being used~~ for the TOE.

~~ALC\_TAT-L.3.2D The developer shall document~~ identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

~~ALC\_TAT-L.3.3D The developer shall describe~~ apply identified implementation standards ~~to be applied~~ for all parts of the TOE as appropriate.

ALC\_TAT-L.3.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

~~ALC\_TAT-L.3.1C All development tools used for implementation shall be well-defined.~~

~~ALC\_TAT-L.3.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.~~

~~ALC\_TAT-L.3.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.~~

None

Evaluator action elements:

~~ALC\_TAT-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the information provided meets all requirements for content and presentation of evidence~~ developer is using well-defined development tools for the TOE.

ALC\_TAT-L.3.2E The evaluator shall confirm, to the level of rigor of appears to be true, that the identified implementation standards have been applied, as appropriate, to all parts of the TOE.

ALC\_TAT-L.3.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-L.3.4E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

#### **A.1.69 ALC\_TAT-R.3 Compliance with implementation standards - all parts**

Dependencies: ADV\_IMP-R.1 Subset of the implementation of the TSF

Developer action elements:

~~ALC\_TAT-L.3.1D~~ The developer shall identify the use well-defined development tools ~~being used~~ for the TOE.

~~ALC\_TAT-L.3.2D~~ The developer shall document identify the selected implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

~~ALC\_TAT-L.3.3D~~ The developer shall ~~describe~~ apply identified implementation standards ~~to be applied~~ for all parts of the TOE as appropriate.

ALC\_TAT-L.3.4D(+) The developer shall use development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

Content and presentation of evidence elements:

~~ALC\_TAT-R.3.1C~~ All development tools used for implementation shall be well-defined.

~~ALC\_TAT-R.3.2C~~ The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

~~ALC\_TAT-R.3.3C~~ The documentation of the development tools shall unambiguously define the meaning of all implementation dependent options.

None

Evaluator action elements:

~~ALC\_TAT-R.3.1E~~ The evaluator shall confirm that the ~~information provided meets all requirements for content and presentation of evidence~~ developer is using well-defined development tools for the TOE.

ALC\_TAT-R.3.2E The evaluator shall confirm that the identified implementation standards have been applied, as appropriate, to all parts of the TOE.

ALC\_TAT-R.3.3E(+) The evaluator shall confirm that the developer has identified the implementation-dependent options of the development tools such that the meaning of each implementation-dependent option to be used is unambiguous.

ALC\_TAT-R.3.4E(+) The evaluator shall confirm that the developer is using development tools for the TOE in a manner that unambiguously defines the meaning of each statement used in the implementation.

## A.6 TESTS

### A.1.70 ATE\_COV-L.2 Analysis of coverage

Objectives: In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved ~~through an examination of by the developer conducting testing on the basis of an analysis of correspondence.~~

Application notes: The developer is required to ~~demonstrate that the tests which have been identified include to have conducted~~ testing of all of the security functions as described in the functional specification. ~~The analysis should not only show on the basis of an analysis that shows the correspondence between tests and security functions, but should provide also sufficient information for the evaluator to determine how the functions have been exercised. This information can be used in planning for additional evaluator tests. Although at this level the developer has to demonstrate that each of the functions within the functional specification has been tested, the amount of testing of each function need not be exhaustive.~~

Dependencies:

ADV\_FSP-L.1 Informal functional specification

ATE\_FUN-L.1 Functional testing

Developer action elements:

ATE\_COV-L.2.1D The developer shall ~~provide an~~ conduct testing on the basis of an analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

Content and presentation of evidence elements:

~~ATE\_COV-L.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.~~

~~ATE\_COV-L.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.~~

None

Evaluator action elements:

~~ATE\_COV-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer testing was performed on the basis of an analysis of test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification. that the information provided meets all requirements for content and presentation of evidence.~~

#### **A.1.71 ATE\_COV-R.2 Analysis of coverage**

**Objectives:** In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved through an examination of analysis of correspondence.

**Application notes:** The developer is required to ~~demonstrate that the tests which have been identified include~~ to have conducted testing of all of the security functions as described in the functional specification. ~~The analysis should not only show on the basis of an analysis that shows the correspondence between tests and security functions, but should provide also~~ and also provides sufficient information for the evaluator to determine how the functions have been exercised. This information can be used in planning for additional evaluator tests. Although at this level the developer has to demonstrate that each of the functions within the functional specification has been tested, the amount of testing of each function need not be exhaustive.

**Dependencies:**

ADV\_FSP-R.1 Informal functional specification

ATE\_FUN-R.1 Functional testing

**Developer action elements:**

~~ATE\_COV-R.2.1D The developer shall provide an~~ conduct testing on the basis of an analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

## Content and presentation of evidence elements:

~~ATE\_COV-R.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.~~

~~ATE\_COV-R.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.~~

None

## Evaluator action elements:

~~ATE\_COV-R.2.1E The evaluator shall confirm that the developer testing was performed on the basis of an analysis of test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification. information provided meets all requirements for content and presentation of evidence.~~

**A.1.72 ATE\_COV-L.3 Analysis of coverage**

Objectives: In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved ~~through an examination of~~ by the developer conducting testing on the basis of a rigorous analysis of correspondence.

Application notes: ~~The developer is required to provide a convincing argument that the tests which have been identified cover all security functions, and that the testing of each security function is complete. There will remain little scope for the evaluator to devise additional functional tests of the TSF interfaces based on the functional specification, as they will have been exhaustively tested. Nevertheless, the evaluator should strive to devise such tests.~~

## Dependencies:

ADV\_FSP-L.1 Informal functional specification

ATE\_FUN-L.1 Functional testing

## Developer action elements:

~~ATE\_COV-L.3.1D The developer shall provide an~~ conduct testing on the basis of an analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

ATE\_COV-L.3.2D(+) The developer shall conduct testing on the basis of a rigorous analysis of the test coverage that was used to ensure that the tests conducted completely tested all internal interfaces of the TSF identified in the functional specification.

Content and presentation of evidence elements:

~~ATE\_COV-L.3.1C~~ The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

~~ATE\_COV-L.3.2C~~ The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

~~ATE\_COV-L.3.3C~~ The analysis of the test coverage shall rigorously demonstrate that all external interfaces of the TSF identified in the functional specification have been completely tested.

None

Evaluator action elements:

ATE\_COV-L.3.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer testing was performed on the basis of an analysis of test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification. ~~that the information provided meets all requirements for content and presentation of evidence.~~

ATE\_COV-L.3.2E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer testing was performed on the basis of a rigorous analysis of test coverage that was used to ensure that all external interfaces of the TSF identified in the functional specification have been completely tested.

### **A.1.73 ATE\_COV-R.3 Analysis of coverage**

**Objectives:** In this component, the objective is to establish that the TSF has been tested against its functional specification in a systematic manner. This is to be achieved through an evaluator examination of and by the developer conducting testing on the basis of a rigorous analysis of correspondence.

**Application notes:** The developer is required to provide a convincing argument that the tests which have been identified cover all security functions, and that the testing of each security function is complete. There will remain little scope for the evaluator to devise additional functional tests of the TSF interfaces based on the functional specification, as they will have been exhaustively tested. Nevertheless, the evaluator should strive to devise such tests.



## Dependencies:

ADV\_FSP-R.1 Informal functional specification

ATE\_FUN-R.1 Functional testing

## Developer action elements:

ATE\_COV-R.3.1D The developer shall ~~provide an~~ conduct testing on the basis of a documented analysis of the test coverage that was used to ensure that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification.

ATE\_COV-R.3.2D(+) The developer shall conduct testing on the basis of a rigorous analysis of the test coverage that was used to ensure that the tests conducted completely tested all internal interfaces of the TSF identified in the functional specification.

## Content and presentation of evidence elements:

~~ATE\_COV-R.3.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.~~

~~ATE\_COV-R.3.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.~~

~~ATE\_COV-R.3.3C The analysis of the test coverage shall rigorously demonstrate that all external interfaces of the TSF identified in the functional specification have been completely tested.~~

None

## Evaluator action elements:

ATE\_COV-R.3.1E The evaluator shall confirm that the developer testing was performed on the basis of an analysis of test coverage that ensures that the tests performed provide complete coverage of the TSF and includes identifying the correspondence between the tests conducted in the test documentation and the TSF as described in the functional specification. ~~information provided meets all requirements for content and presentation of evidence.~~

ATE\_COV-R.3.2E(+) The evaluator shall confirm that the developer testing was performed on the basis of a rigorous analysis of test coverage that ensures that all external interfaces of the TSF identified in the functional specification have been completely tested.

#### A.1.74 ATE\_DPT-L.1 Testing: high-level design

**Objectives:** The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized.

**Application notes:** The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts.

**Dependencies:**

ADV\_HLD-L.1 Descriptive high-level design

ATE\_FUN-L.1 Functional testing

**Developer action elements:**

ATE\_DPT-L.1.1D The developer shall ~~provide the~~ conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

**Content and presentation of evidence elements:**

~~ATE\_DPT-L.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.~~

None

**Evaluator action elements:**

~~ATE\_DPT-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that developer testing was conducted~~ conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design ~~the information provided meets all requirements for content and presentation of evidence.~~

#### A.1.75 ATE\_DPT-L.2 Testing: low-level design

**Objectives:** The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized. The modules of a TSF provide a description of the internal workings of the TSF. Testing at the level of the modules, in order to demonstrate the presence of any flaws, provides assurance that the TSF modules have been correctly realized.

Application notes: The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts. The developer is expected to describe the testing of the low-level design of the TSF in terms of "modules". The term "modules" is used to express the notion of decomposing each of the "subsystems" of the TSF into a relatively small number of parts.

Dependencies:

ADV\_HLD-L.1 Descriptive high-level design

ADV\_LLD-L.1 Descriptive low-level design

ATE\_FUN-L.1 Functional testing

Developer action elements:

ATE\_DPT-L.2.1D The developer shall ~~provide the~~ conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and its low-level design.

Content and presentation of evidence elements:

~~ATE\_DPT-L.2.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and low-level design.~~

None

Evaluator action elements:

ATE\_DPT-L.2.1E The evaluator shall confirm that developer testing was conducted conduct testing on the basis of an analysis of the depth of testing that demonstrates, to the level of rigor of appears reasonable, that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and its low-level design ~~the information provided meets all requirements for content and presentation of evidence.~~

#### **A.1.76 ATE\_DPT-R.2 Testing: low-level design**

Objectives: The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized. The modules of a TSF provide a description of the internal workings of the TSF. Testing at the level of the modules, in order to demonstrate the presence of any flaws, provides assurance that the TSF modules have been correctly realized.

Application notes: The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts. The developer is expected to describe the testing of the low-level design of the TSF in terms of "modules". The term "modules" is used to express the notion of decomposing each of the "subsystems" of the TSF into a relatively small number of parts.

Dependencies:

ADV\_HLD-R.1 Descriptive high-level design  
 ADV\_LLD-R.1 Descriptive low-level design  
 ATE\_FUN-R.1 Functional testing

Developer action elements:

ATE\_DPT-R.2.1D The developer shall ~~provide the~~ conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and its low-level design.

Content and presentation of evidence elements:

~~ATE\_DPT-R.2.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and low-level design.~~

None

Evaluator action elements:

ATE\_DPT-R.2.1E The evaluator shall confirm that developer testing was conducted conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and its low-level design ~~the information provided meets all requirements for content and presentation of evidence.~~

### **A.1.77 ATE\_DPT-L.3 Testing: implementation representation**

Objectives: The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence of any flaws, provides assurance that the TSF subsystems have been correctly realized. The modules of a TSF provide a description of the internal workings of the TSF. Testing at the level of the modules, in order to demonstrate the presence of any flaws, provides assurance that the TSF modules have been correctly realized. The implementation representation of a TSF provides a detailed description of the internal workings of the TSF. Testing at the level

of the implementation, in order to demonstrate the presence of any flaws, provides assurance that the TSF implementation has been correctly realized.

Application notes: The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts. The developer is expected to describe the testing of the low-level design of the TSF in terms of "modules". The term "modules" is used to express the notion of decomposing each of the "subsystems" of the TSF into a relatively small number of parts. The implementation representation is the one which is used to generate the TSF itself (e.g. source code which is then compiled).

#### Dependencies:

ADV\_HLD-L.2 Security enforcing high-level design  
 ADV\_IMP-L.2 Implementation of the TSF  
 ADV\_LLD-L.1 Descriptive low-level design  
 ATE\_FUN-L.1 Functional testing

#### Developer action elements:

ATE\_DPT-L.3.1D The developer shall provide the conduct testing on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation.

#### Content and presentation of evidence elements:

~~ATE\_DPT-L.3.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and low-level design.~~

None

#### Evaluator action elements:

ATE\_DPT-L.3.1E The evaluator shall confirm that developer testing was conducted on the basis of an analysis of the depth of testing that demonstrates, to the level of rigor of appears reasonable, that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation ~~the information provided meets all requirements for content and presentation of evidence.~~

### A.1.78 ATE\_DPT-R.3 Testing: implementation representation

Objectives: The subsystems of a TSF provide a high-level description of the internal workings of the TSF. Testing at the level of the subsystems, in order to demonstrate the presence

of any flaws, provides assurance that the TSF subsystems have been correctly realized. The modules of a TSF provide a description of the internal workings of the TSF. Testing at the level of the modules, in order to demonstrate the presence of any flaws, provides assurance that the TSF modules have been correctly realized. The implementation representation of a TSF provides a detailed description of the internal workings of the TSF. Testing at the level of the implementation, in order to demonstrate the presence of any flaws, provides assurance that the TSF implementation has been correctly realized.

Application notes: The developer is expected to describe the testing of the high-level design of the TSF in terms of "subsystems". The term "subsystem" is used to express the notion of decomposing the TSF into a relatively small number of parts. The developer is expected to describe the testing of the low-level design of the TSF in terms of "modules". The term "modules" is used to express the notion of decomposing each of the "subsystems" of the TSF into a relatively small number of parts. The implementation representation is the one which is used to generate the TSF itself (e.g. source code which is then compiled).

Dependencies:

ADV\_HLD-R.2 Security enforcing high-level design  
ADV\_IMP-R.2 Implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design  
ATE\_FUN-R.1 Functional testing

Developer action elements:

ATE\_DPT-R.3.1D The developer shall provide the conduct testing on the basis of a documented analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation.

Content and presentation of evidence elements:

~~ATE\_DPT-R.3.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design and low-level design.~~

None

Evaluator action elements:

ATE\_DPT-R.3.1E The evaluator shall confirm that developer testing was conducted on the basis of an analysis of the depth of testing that demonstrates that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design, its low-level design, and its implementation representation ~~the information provided meets all requirements for content and presentation of evidence.~~

### A.1.79 ATE\_FUN-L.1 Functional testing

**Objectives:** The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation.

**Dependencies:** No dependencies.

**Developer action elements:**

ATE\_FUN-L.1.1D The developer shall test the TSF ~~and document the results.~~

ATE\_FUN-L.1.2D The developer shall ~~provide~~ produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

**Content and presentation of evidence elements:**

ATE\_FUN-L.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-L.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-L.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE\_FUN-L.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-L.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

**Evaluator action elements:**

ATE\_FUN-L.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the ~~information provided meets all requirements for content and presentation of evidence that the developer tested the TSF.~~

ATE\_FUN-L.1.2E(+) *placeholder for element appearing at higher component*

ATE\_FUN-L.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.

**A.1.80 ATE\_FUN-R.1 Functional testing**

**Objectives:** The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation.

**Dependencies:** No dependencies.

**Developer action elements:**

ATE\_FUN-R.1.1D The developer shall test the TSF and document the results.

ATE\_FUN-R.1.2D The developer shall ~~provide~~ produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

**Content and presentation of evidence elements:**

ATE\_FUN-R.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-R.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-R.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE\_FUN-R.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-R.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

**Evaluator action elements:**

ATE\_FUN-R.1.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the information provided meets all requirements for content and presentation of evidence that the developer tested the TSF.

ATE\_FUN-R.1.2E(+) *placeholder for element appearing at higher component*

ATE\_FUN-R.1.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.



### A.1.81 ATE\_FUN-L.2 Ordered functional testing

**Objectives:** The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation. In this component, an additional objective is to ensure that testing is structured such as to avoid circular arguments about the correctness of the portions of the TSF being tested.

**Application notes:** Although the test procedures may state pre-requisite initial test conditions in terms of ordering of tests, they may not provide a rationale for the ordering. An analysis of test ordering is an important factor in determining the adequacy of testing, as there is a possibility of faults being concealed by the ordering of tests.

**Dependencies:** No dependencies.

**Developer action elements:**

ATE\_FUN-L.2.1D The developer shall test the TSF and document the results.

ATE\_FUN-L.2.2D The developer shall provide produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

ATE\_FUN-L.2.3D(+) The develop shall conduct testing on the basis of an analysis of the test procedure ordering dependencies.

**Content and presentation of evidence elements:**

ATE\_FUN-L.2.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-L.2.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-L.2.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE\_FUN-L.2.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-L.2.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE\_FUN-L.2.6C The test documentation shall include an analysis of the test procedure ordering dependencies.

**Evaluator action elements:**

ATE\_FUN-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the information provided meets all requirements for content and presentation of evidence that the developer tested the TSF.

ATE\_FUN-L.2.2E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer conducted testing on the basis of an analysis of the test procedure ordering dependencies.

ATE\_FUN-L.2.3E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.

### **A.1.82 ATE\_FUN-R.2 Ordered functional testing**

**Objectives:** The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation. In this component, an additional objective is to ensure that testing is structured such as to avoid circular arguments about the correctness of the portions of the TSF being tested.

**Application notes:** Although the test procedures may state pre-requisite initial test conditions in terms of ordering of tests, they may not provide a rationale for the ordering. An analysis of test ordering is an important factor in determining the adequacy of testing, as there is a possibility of faults being concealed by the ordering of tests.

**Dependencies:** No dependencies.

**Developer action elements:**

ATE\_FUN-R.2.1D The developer shall test the TSF and document the results.

ATE\_FUN-R.2.2D The developer shall provide produce test documentation developed as an integral part of the testing process and containing the information described in the content and presentation section below.

ATE\_FUN-R.2.3D(+) The develop shall conduct testing on the basis of an analysis of the test procedure ordering dependencies.

**Content and presentation of evidence elements:**

ATE\_FUN-R.2.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE\_FUN-R.2.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE\_FUN-R.2.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE\_FUN-R.2.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE\_FUN-R.2.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

ATE\_FUN-R.2.6C The test documentation shall include an analysis of the test procedure ordering dependencies.

Evaluator action elements:

ATE\_FUN-R.2.1E The evaluator shall confirm that the ~~information provided meets all requirements for content and presentation of evidence~~ that the developer tested the TSF.

ATE\_FUN-R.2.2E(+) The evaluator shall confirm that the developer conducted testing on the basis of an analysis of the test procedure ordering dependencies.

ATE\_FUN-R.2.3E(+) The evaluator shall confirm that test documentation was produced as an integral part of the developer's testing process and contains the information identified in the content and presentation section above.

### A.1.83 ATE\_IND-R.1 Independent testing - conformance

Objectives: In this component, the objective is to demonstrate that the security functions perform as specified.

Application notes: This component does not address the use of developer test results. It is applicable where such results are not available, and also in cases where the developer's testing is accepted without validation. The evaluator is required to devise and conduct tests with the objective of confirming that the TOE security functional requirements are met. ~~The approach is to gain confidence in correct operation through representative testing, rather than to conduct every possible test. The extent of testing to be planned for this purpose is a methodology issue, and needs to be considered in the context of a particular TOE and the balance of other evaluation activities.~~

Dependencies:

ADV\_FSP-L.1 Informal functional specification

AGD\_ADM-L.1 Administrator guidance

AGD\_USR-L.1 User guidance

Developer action elements:

ATE\_IND-R.1.1D The developer shall provide the TOE suitable for testing.

Content and presentation of evidence elements:

~~ATE\_IND-R.1.1C The TOE shall be suitable for testing. None~~

Evaluator action elements:

ATE\_IND-R.1.1E The evaluator shall confirm that the developer has provides the TOE suitable for testing ~~information provided meets all requirements for content and presentation of evidence.~~

ATE\_IND-R.1.2E The evaluator shall test ~~a subset of the TSF as appropriate to confirm that the TOE~~ meets all functional requirements in the associated security target ~~operates as specified.~~

#### **A.1.84 ATE\_IND-L.2 Independent testing - sample**

Objectives: The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes selecting and repeating a sample of the developer tests.

Application notes: The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. This component contains a requirement that the evaluator has available test results from the developer to supplement the program of testing. The evaluator will repeat a sample of the developer's tests to gain confidence in the results obtained. Having established such confidence the evaluator will build upon the developer's testing, as necessary to confirm the developer testing, by conducting additional tests that exercise the TOE in a different manner. By using a platform of validated developer test results the evaluator is able to gain confidence that the TOE operates correctly in a wider range of conditions than would be possible purely using the developer's own efforts, given a fixed level of resource. Having gained confidence that the developer has tested the TOE, the evaluator will also have more freedom, where appropriate, to concentrate testing in areas where examination of documentation or specialist knowledge has raised particular concerns.

Dependencies:

ADV\_FSP-L.1 Informal functional specification  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance  
ATE\_FUN-L.1 Functional testing

Developer action elements:

ATE\_IND-L.2.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Content and presentation of evidence elements:

~~ATE\_IND-L.2.1C The TOE shall be suitable for testing.~~

~~ATE\_IND-L.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.~~

None

Evaluator action elements:

ATE\_IND-L.2.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF ~~information provided meets all requirements for content and presentation of evidence.~~

ATE\_IND-L.2.2E The evaluator shall test ~~a subset of the TSF~~ only as necessary as appropriate to confirm that the developer testing shows that the TOE meets all functional requirements in the associated security target. ~~operates as specified.~~

ATE\_IND-L.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

#### **A.1.85 ATE\_IND-R.2 Independent testing - sample**

**Objectives:** The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes selecting and repeating a sample of the developer tests.

**Application notes:** The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. This component contains a requirement that the evaluator has available test results from the developer to supplement the program of testing. The evaluator will repeat a sample of the developer's tests to gain confidence in the results obtained. Having established such confidence the evaluator will build upon the developer's testing by conducting additional tests that exercise the TOE in a different manner. By using a platform of validated developer test results the evaluator is able to gain confidence that the TOE operates correctly in a wider range of conditions than would be possible purely using the developer's own efforts, given a fixed level of resource. Having gained confidence that the developer has tested the TOE, the evaluator will also have more freedom, where appropriate, to concentrate testing in areas where examination of documentation or specialist knowledge has raised particular concerns.

**Dependencies:**

ADV\_FSP-R.1 Informal functional specification  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance  
ATE\_FUN-R.1 Functional testing

Developer action elements:

ATE\_IND-R.2.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Content and presentation of evidence elements:

~~ATE\_IND-R.2.1C The TOE shall be suitable for testing.~~  
~~ATE\_IND-R.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.~~

None

Evaluator action elements:

ATE\_IND-R.2.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF ~~information provided meets all requirements for content and presentation of evidence.~~

ATE\_IND-R.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE meets all functional requirements in the associated security target ~~operates as specified.~~

ATE\_IND-R.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

### **A.1.86 ATE\_IND-L.3 Independent testing - complete**

**Objectives:** The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes repeating all of the developer tests.

**Application notes:** The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. In this component the evaluator must repeat all of the developer's tests as part of the program of testing. As in the previous component the evaluator will also conduct tests that aim to exercise the TOE in a different manner from that achieved by the developer. In cases where developer testing has been exhaustive, there may remain little scope for this.

**Dependencies:**

ADV\_FSP-L.1 Informal functional specification  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance  
ATE\_FUN-L.1 Functional testing

**Developer action elements:**

ATE\_IND-L.3.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

**Content and presentation of evidence elements:**

~~ATE\_IND-L.3.1C The TOE shall be suitable for testing.~~  
~~ATE\_IND-L.3.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.~~

None

**Evaluator action elements:**

ATE\_IND-L.3.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF ~~information provided meets all requirements for content and presentation of evidence.~~

ATE\_IND-L.3.2E The evaluator shall test ~~a subset of the TSF~~ only as necessary as appropriate to confirm that the developer testing shows that the TOE meets all functional requirements in the associated security target. ~~operates as specified.~~

ATE\_IND-L.3.3E The evaluator shall execute all of tests in the test documentation to verify the developer test results.

**A.1.87 ATE\_IND-R.3 Independent testing - complete**

**Objectives:** The objective is to demonstrate that the security functions perform as specified. Evaluator testing includes repeating all of the developer tests.

**Application notes:** The intent is that the developer should provide the evaluator with materials necessary for the efficient reproduction of developer tests. This may include such things as machine-readable test documentation, test programs, etc. In this component the evaluator must repeat all of the developer's tests as part of the program of testing. As in the previous component the evaluator will also conduct tests that aim to exercise the TOE in a different

manner from that achieved by the developer. In cases where developer testing has been exhaustive, there may remain little scope for this.

Dependencies:

ADV\_FSP-R.1 Informal functional specification  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance  
ATE\_FUN-R.1 Functional testing

Developer action elements:

ATE\_IND-R.3.1D The developer shall provide the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Content and presentation of evidence elements:

~~ATE\_IND-R.3.1C The TOE shall be suitable for testing.~~  
~~ATE\_IND-R.3.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.~~

None

Evaluator action elements:

ATE\_IND-R.3.1E The evaluator shall confirm that the developer has provided the TOE suitable for testing with an equivalent set of resources to those that were used in the developer's functional testing of the TSF ~~information provided meets all requirements for content and presentation of evidence.~~

ATE\_IND-R.3.2E The evaluator shall test a ~~subset of the TSF~~ as appropriate to confirm that the developer testing shows that the TOE meets all functional requirements in the associated security target. ~~operates as specified.~~

ATE\_IND-R.3.3E The evaluator shall execute all of tests in the test documentation to verify the developer test results.

## A.7 VULNERABILITY ASSESSMENT

### A.1.88 AVA\_CCA-L.1 Covert channel analysis

Objectives: The objective is to identify covert channels that are identifiable, through an informal search for covert channels.



### Dependencies:

ADV\_FSP-L.2 Fully defined external interfaces  
ADV\_IMP-L.2 Implementation of the TSF  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

### Developer action elements:

AVA\_CCA-L.1.1D The developer shall conduct a search for covert channels for each information flow control policy; identifying each covert channel, estimating its capacity, and determining the worst-case exploitation scenario.

~~AVA\_CCA-L.1.2D The developer shall provide covert channel analysis documentation.~~

### Content and presentation of evidence elements:

~~AVA\_CCA-L.1.1C The analysis documentation shall identify covert channels and estimate their capacity.~~

~~AVA\_CCA-L.1.2C The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.~~

~~AVA\_CCA-L.1.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.~~

~~AVA\_CCA-L.1.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.~~

~~AVA\_CCA-L.1.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.~~

None

### Evaluator action elements:

AVA\_CCA-L.1.1E The evaluator shall confirm check that the developer has identified covert channels and for each channel estimated its capacity, and determined the worst-case exploitation scenario information provided meets all requirements for content and presentation of evidence.

~~AVA\_CCA-L.1.2E The evaluator shall confirm that the results of the covert channel analysis show that the TOE meets its functional requirements.~~

~~AVA\_CCA-L.1.3E The evaluator shall selectively validate the covert channel analysis through testing.~~

### A.1.89 AVA\_CCA-R.1 Covert channel analysis

**Objectives:** The objective is to identify covert channels that are identifiable, through an informal search for covert channels.

**Dependencies:**

ADV\_FSP-R.2 Fully defined external interfaces  
ADV\_IMP-R.2 Implementation of the TSF  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance

**Developer action elements:**

AVA\_CCA-R.1.1D The developer shall conduct a search for covert channels for each information flow control policy; identifying each covert channel, estimating its capacity, and determining the worst-case exploitation scenario.

AVA\_CCA-R.1.2D The developer shall provide covert channel analysis documentation containing the information identified in the content and presentation section below.

**Content and presentation of evidence elements:**

AVA\_CCA-R.1.1C The analysis documentation shall identify covert channels and estimate their capacity.

AVA\_CCA-R.1.2C The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.

AVA\_CCA-R.1.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.

AVA\_CCA-R.1.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.

AVA\_CCA-R.1.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

**Evaluator action elements:**

AVA\_CCA-R.1.1E The evaluator shall confirm that the developer has identified covert channels and for each channel estimated its capacity, and determined the worst-case exploitation scenario ~~information provided meets all requirements for content and presentation of evidence.~~

AVA\_CCA-R.1.2E The evaluator shall confirm that the results of the covert channel analysis show that the TOE meets its functional requirements.

AVA\_CCA-R.1.3E The evaluator shall selectively validate the covert channel analysis through testing.

#### **A.1.90 AVA\_CCA-L.2 Systematic covert channel analysis**

Objectives: The objective is to identify covert channels that are identifiable, through an informal search for covert channels.

Application notes: Performing a covert channel analysis in a systematic way requires that the developer identify covert channels in a structured and repeatable way, as opposed to identifying covert channels in an ad-hoc fashion.

Dependencies:

ADV\_FSP-L.2 Fully defined external interfaces  
ADV\_IMP-L.2 Implementation of the TSF  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

Developer action elements:

AVA\_CCA-L.2.1D The developer shall conduct a systematic search for covert channels for each information flow control policy; identifying each covert channel, estimating its capacity, and determining the worst-case exploitation scenario.

~~AVA\_CCA-L.2.2D The developer shall provide covert channel analysis documentation.~~

Content and presentation of evidence elements:

~~AVA\_CCA-L.2.1C The analysis documentation shall identify covert channels and estimate their capacity.~~

~~AVA\_CCA-L.2.2C The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.~~

~~AVA\_CCA-L.2.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.~~

~~AVA\_CCA-L.2.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.~~

~~AVA\_CCA-L.2.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.~~

~~AVA\_CCA-L.2.6C The analysis documentation shall provide evidence that the method used to identify covert channels is systematic.~~

None

Evaluator action elements:

~~AVA\_CCA-L.2.1E The evaluator shall confirm, to the level of rigor of appears to be true, that the developer has systematically identified covert channels and for each channel estimated its capacity, and determined the worst-case exploitation scenario information provided meets all requirements for content and presentation of evidence.~~

~~AVA\_CCA-L.2.2E The evaluator shall confirm that the results of the covert channel analysis show that the TOE meets its functional requirements.~~

~~AVA\_CCA-L.2.3E The evaluator shall selectively validate the covert channel analysis through testing.~~

### **A.1.91 AVA\_CCA-R.2 Systematic covert channel analysis**

**Objectives:** The objective is to identify covert channels that are identifiable, through an informal search for covert channels.

**Application notes:** Performing a covert channel analysis in a systematic way requires that the developer identify covert channels in a structured and repeatable way, as opposed to identifying covert channels in an ad-hoc fashion.

**Dependencies:**

ADV\_FSP-R.2 Fully defined external interfaces  
ADV\_IMP-R.2 Implementation of the TSF  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance

**Developer action elements:**

AVA\_CCA-R.2.1D The developer shall conduct a systematic search for covert channels for each information flow control policy; identifying each covert channel, estimating its capacity, and determining the worst-case exploitation scenario.

AVA\_CCA-R.2.2D The developer shall provide covert channel analysis documentation containing the information identified in the content and presentation section below.

**Content and presentation of evidence elements:**

AVA\_CCA-R.2.1C The analysis documentation shall identify covert channels and estimate their capacity.

AVA\_CCA-R.2.2C The analysis documentation shall describe the procedures used for determining the existence of covert channels, and the information needed to carry out the covert channel analysis.

AVA\_CCA-R.2.3C The analysis documentation shall describe all assumptions made during the covert channel analysis.

AVA\_CCA-R.2.4C The analysis documentation shall describe the method used for estimating channel capacity, based on worst case scenarios.

AVA\_CCA-R.2.5C The analysis documentation shall describe the worst case exploitation scenario for each identified covert channel.

AVA\_CCA-R.2.6C The analysis documentation shall provide evidence that the method used to identify covert channels is systematic.

Evaluator action elements:

AVA\_CCA-R.2.1E The evaluator shall confirm that the developer has systematically identified covert channels and for each channel estimated its capacity, and determined the worst-case exploitation scenario ~~information provided meets all requirements for content and presentation of evidence.~~

AVA\_CCA-R.2.2E The evaluator shall confirm that the results of the covert channel analysis show that the TOE meets its functional requirements.

AVA\_CCA-R.2.3E The evaluator shall selectively validate the covert channel analysis through testing.

#### A.1.92 AVA\_MSU-L.2 Validation of analysis

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met.

Dependencies:

ADO\_IGS-L.1 Installation, generation, and start-up procedures

ADV\_FSP-L.1 Informal functional specification

AGD\_ADM-L.1 Administrator guidance

AGD\_USR-L.1 User guidance

Developer action elements:

AVA\_MSU-L.2.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-L.2.2D The developer shall ~~document an analysis of~~ analyze the guidance documentation to determine that the guidance documentation is complete.

Content and presentation of evidence elements:

AVA\_MSU-L.2.1C The guidance documentation shall identify ~~all~~ possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-L.2.2C *CC element incorporated into developer elements* ~~The guidance documentation shall be complete, clear, consistent and reasonable.~~

AVA\_MSU-L.2.3C The guidance documentation shall list ~~all~~ assumptions about the intended environment.

AVA\_MSU-L.2.4C The guidance documentation shall list ~~all~~ requirements for external security measures (including external procedural, physical and personnel controls).

~~AVA\_MSU-L.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.~~

#### Evaluator action elements:

AVA\_MSU-L.2.1E The evaluator shall ~~confirm~~ check that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-L.2.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-L.2.3E The evaluator shall determine, to the level of rigor of appears to be true, that the use of the guidance documentation allows ~~all~~ insecure states to be detected.

AVA\_MSU-L.2.4E The evaluator shall confirm, to the level of rigor of appears reasonable, that the analysis documentation shows that guidance is provided for secure operation in ~~all~~ the modes of operation of the TOE.

### A.1.93 AVA\_MSU-R.2 Validation of analysis

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met.

#### Dependencies:

ADO\_IGS-R.1 Installation, generation, and start-up procedures

ADV\_FSP-R.1 Informal functional specification  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance

Developer action elements:

AVA\_MSU-R.2.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-R.2.2D The developer shall ~~document an analysis of~~ analyze the guidance documentation to determine that the guidance documentation is complete.

Content and presentation of evidence elements:

AVA\_MSU-R.2.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-R.2.2C *CC element incorporated into developer elements* ~~The guidance documentation shall be complete, clear, consistent and reasonable.~~

AVA\_MSU-R.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA\_MSU-R.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

~~AVA\_MSU-R.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.~~

Evaluator action elements:

AVA\_MSU-R.2.1E The evaluator shall confirm that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-R.2.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-R.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA\_MSU-R.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

### A.1.94 AVA\_MSU-L.3 Analysis and testing for insecure states

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met, and this analysis is validated and confirmed through testing by the developer evaluator.

**Application notes:** In this component the developer evaluator is required to undertake testing to ensure that if and when the TOE enters an insecure state this may easily be detected. This testing may be considered as a specific aspect of penetration testing.

**Dependencies:**

ADO\_IGS-L.1 Installation, generation, and start-up procedures  
ADV\_FSP-L.1 Informal functional specification  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

**Developer action elements:**

AVA\_MSU-L.3.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-L.3.2D The developer shall document an analysis of analyze the guidance documentation to determine that the guidance documentation is complete.

AVA\_MSU-L.3.3D(+) The developer shall perform testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

**Content and presentation of evidence elements:**

AVA\_MSU-L.3.1C The guidance documentation shall identify ~~all~~ possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-L.3.2C *CC element incorporated into developer elements* ~~The guidance documentation shall be complete, clear, consistent and reasonable.~~

AVA\_MSU-L.3.3C The guidance documentation shall list ~~all~~ assumptions about the intended environment.

AVA\_MSU-L.3.4C The guidance documentation shall list ~~all~~ requirements for external security measures (including external procedural, physical and personnel controls).



~~AVA\_MSU-L.3.5C The analysis documentation shall demonstrate that the guidance documentation is complete.~~

Evaluator action elements:

AVA\_MSU-L.3.1E The evaluator shall ~~confirm~~ check that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-L.3.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-L.3.3E The evaluator shall determine, to the level of rigor of appears to be true, that the use of the guidance documentation allows ~~all~~ insecure states to be detected.

AVA\_MSU-L.3.4E The evaluator shall confirm, to the level of rigor of appears reasonable, that the analysis documentation shows that guidance is provided for secure operation in ~~all~~ the modes of operation of the TOE.

AVA\_MSU-L.3.5E ~~CC element deleted~~ The evaluator shall perform independent testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

AVA\_MSU-L.3.6E(+) The evaluator shall confirm, to the level of rigor of appears to be true, that the developer performed testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

#### **A.1.95 AVA\_MSU-R.3 Analysis and testing for insecure states**

**Objectives:** The objective is to ensure that misleading, unreasonable and conflicting guidance is absent from the guidance documentation, and that secure procedures for all modes of operation have been addressed. Insecure states should be easy to detect. In this component, an analysis of the guidance documentation by the developer is required to provide additional assurance that the objective has been met, and this analysis is validated and confirmed through testing by the developer and the evaluator.

**Application notes:** In this component the developer and the evaluator are required to undertake testing to ensure that if and when the TOE enters an insecure state this may easily be detected. This testing may be considered as a specific aspect of penetration testing.

**Dependencies:**

ADO\_IGS-R.1 Installation, generation, and start-up procedures

ADV\_FSP-R.1 Informal functional specification

AGD\_ADM-R.1 Administrator guidance

AGD\_USR-R.1 User guidance

Developer action elements:

AVA\_MSU-R.3.1D The developer shall provide guidance documentation, that is complete, clear, consistent, and reasonable and includes the information identified in the content and presentation section below.

AVA\_MSU-R.3.2D The developer shall ~~document an analysis of~~ analyze the guidance documentation to determine that the guidance documentation is complete.

AVA\_MSU-R.3.3D(+) The developer shall perform testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

Content and presentation of evidence elements:

AVA\_MSU-R.3.1C The guidance documentation shall identify ~~all~~ possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA\_MSU-R.3.2C *CC element incorporated into developer elements* ~~The guidance documentation shall be complete, clear, consistent and reasonable.~~

AVA\_MSU-R.3.3C The guidance documentation shall list ~~all~~ assumptions about the intended environment.

AVA\_MSU-R.3.4C The guidance documentation shall list ~~all~~ requirements for external security measures (including external procedural, physical and personnel controls).

~~AVA\_MSU-R.3.5C The analysis documentation shall demonstrate that the guidance documentation is complete.~~

Evaluator action elements:

AVA\_MSU-R.3.1E The evaluator shall confirm that the guidance documentation contains the information identified in the content and presentation of evidence section above.

AVA\_MSU-R.3.2E If the developer has not already taken measures deemed adequate to show this, the evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA\_MSU-R.3.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA\_MSU-R.3.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

AVA\_MSU-R.3.5E The evaluator shall perform independent testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

AVA\_MSU-R.3.6E(+) The evaluator shall confirm that the developer performed testing to determine that an administrator or user, with an understanding of the guidance documentation, would reasonably be able to determine if the TOE is configured and operating in a manner that is insecure.

#### **A.1.96 AVA\_SOF-L.1 Strength of TOE security function evaluation**

Dependencies:

ADV\_FSP-L.1 Informal functional specification

~~ADV\_HLD.1 Descriptive high-level design~~

Developer action elements:

AVA\_SOF-L.1.1D The developer shall perform a strength of TOE security function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.

Content and presentation of evidence elements:

~~AVA\_SOF-L.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.~~

~~AVA\_SOF-L.1.2C For each mechanism with a specific strength of TOE security function claim and a specific strength of function metric, the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.~~

None

Evaluator action elements:

~~AVA\_SOF-L.1.1E The evaluator shall confirm, to the level of rigor of appears reasonable, that the developer's strength of function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric meets all requirements for content and presentation of evidence.~~

~~AVA\_SOF-L.1.2E The evaluator shall confirm that the strength claims are correct.~~

#### **A.1.97 AVA\_SOF-R.1 Strength of TOE security function evaluation**

Dependencies:

ADV\_FSP-R.1 Informal functional specification  
ADV\_HLD-R.1 Descriptive high-level design

Developer action elements:

~~AVA\_SOF-R.1.1D The developer shall perform a strength of TOE security function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric.~~

Content and presentation of evidence elements:

~~AVA\_SOF-R.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.~~

~~AVA\_SOF-R.1.2C For each mechanism with a specific strength of TOE security function claim and a specific strength of function metric, the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.~~

None

Evaluator action elements:

~~AVA\_SOF-R.1.1E The evaluator shall confirm that the developer's strength of function analysis that shows for each mechanism identified in the ST as having a strength of TOE security function claim that the mechanism meets or exceeds the minimum strength level defined in the PP/ST and, when the PP/ST defines a specific strength of function metric, meets or exceeds that metric meets all requirements for content and presentation of evidence.~~

AVA\_SOF-R.1.2E The evaluator shall ~~confirm~~ determine that the strength claims are correct.

#### **A.1.98 AVA\_VLA-L.2 Independent vulnerability analysis**

Objectives. ~~A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks performed using publicly known attacks and otherwise obvious attack methods by attackers possessing a low attack potential.~~

Dependencies:

~~ADV\_FSP.1 Informal functional specification  
ADV\_HLD.2 Security enforcing high level design  
ADV\_IMP.1 Subset of the implementation of the TSP  
ADV\_LLD.1 Descriptive low level design  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance~~

Developer action elements:

~~AVA\_VLA-L.2.1D CC element deleted The developer shall perform and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.~~

~~AVA\_VLA-L.2.2D CC element deleted The developer shall document the disposition of identified vulnerabilities.~~

AVA\_VLA-L.2.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

Content and presentation of evidence elements:

~~AVA\_VLA-L.2.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.~~

~~AVA\_VLA-L.2.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.~~

None

Evaluator action elements:

AVA\_VLA-L.2.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing information provided meets all requirements for content and presentation of evidence.

AVA\_VLA-L.2.2E *CC element deleted* ~~The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.~~

AVA\_VLA-L.2.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.2.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods ~~of additional identified vulnerabilities in the intended environment.~~

AVA\_VLA-L.2.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods possessing a low attack potential.

#### **A.1.99 AVA\_VLA-L.3 Moderately resistant**

Objectives. A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks using publicly known attacks and otherwise obvious attack methods. The developer performs penetration testing, supported by a vulnerability analysis, to determine that the TOE is resistant to penetration attacks by attackers possessing a moderate attack potential.

#### **Dependencies:**

ADV\_FSP-L.1 Informal functional specification  
 ADV\_HLD-L.2 Security enforcing high-level design  
 ADV\_IMP-L.1 Subset of the implementation of the TSF  
 ADV\_INT-L.1 Modularity  
 ADV\_LLD-L.1 Descriptive low-level design  
 AGD\_ADM-L.1 Administrator guidance  
 AGD\_USR-L.1 User guidance  
 FPT\_RVM.1 Non-Bypassability of the TSP  
 FPT\_SEP.2 SFP domain separation

#### **Developer action elements:**

AVA\_VLA-L.3.1D The developer shall perform ~~and document~~ a systematic analysis of the TOE deliverables searching for ways in which a user possessing a moderate attack potential can violate the TSP.

AVA\_VLA-L.3.2D The developer shall conduct penetration testing based upon this analysis to determine that the TOE is resistant to penetration attacks by attackers possessing a moderate attack potential ~~document the disposition of identified vulnerabilities.~~

AVA\_VLA-L.3.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

Content and presentation of evidence elements:

~~AVA\_VLA-L.3.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.~~

~~AVA\_VLA-L.3.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.~~

~~AVA\_VLA-L.3.3C The evidence shall show that the search for vulnerabilities is systematic.~~

None

Evaluator action elements:

AVA\_VLA-L.3.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing ~~information provided meets all requirements for content and presentation of evidence.~~

~~AVA\_VLA-L.3.2E CC element deleted The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.~~

AVA\_VLA-L.3.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.3.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods ~~of additional identified vulnerabilities in the intended environment.~~

AVA\_VLA-L.3.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods ~~possessing a moderate attack potential.~~

AVA\_VLA-L.3.6E(+) The evaluator shall determine, to the level of rigor of appears reasonable, that the developer has performed penetration testing on the basis of a systematic

vulnerability analysis to determine that the TOE is resistant to penetration attacks by attackers possessing moderate attack capability.

### **A.1.100 AVA\_VLA-R.3 Moderately resistant**

Objectives. A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks using publicly known attacks and otherwise obvious attack methods and by attackers possessing a moderate attack potential.

Dependencies:

ADV\_FSP-R.1 Informal functional specification  
 ADV\_HLD-R.2 Security enforcing high-level design  
 ADV\_IMP-R.1 Subset of the implementation of the TSF  
 ADV\_INT-R.1 Modularity  
 ADV\_LLD-R.1 Descriptive low-level design  
 AGD\_ADM-R.1 Administrator guidance  
 AGD\_USR-R.1 User guidance  
 FPT\_RVM.1 Non-Bypassability of the TSP  
 FPT\_SEP.2 SFP domain separation

Developer action elements:

AVA\_VLA-R.3.1D The developer shall perform and document a systematic analysis of the TOE deliverables searching for ways in which a user possessing a moderate attack potential can violate the TSP.

AVA\_VLA-R.3.2D The developer shall conduct penetration testing based upon this analysis to determine that the TOE is resistant to penetration attacks by attackers possessing a moderate attack potential ~~document the disposition of identified vulnerabilities.~~

AVA\_VLA-R.3.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

Content and presentation of evidence elements:

AVA\_VLA-R.3.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA\_VLA-R.3.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AL5- AVA\_VLA-R.3.3C The evidence shall show that the search for vulnerabilities is systematic.



#### Evaluator action elements:

AVA\_VLA-R.3.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing information provided meets all requirements for content and presentation of evidence.

AVA\_VLA-R.3.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA\_VLA-R.3.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods and with respect to attackers with moderate attack capability.

AVA\_VLA-R.3.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods and to attackers with moderate attack capability of additional identified vulnerabilities in the intended environment.

AVA\_VLA-R.3.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods and by an attacker possessing a moderate attack potential.

AVA\_VLA-R.3.6E(+) The evaluator shall determine that the developer has performed penetration testing on the basis of a systematic vulnerability analysis to determine that the TOE is resistant to penetration attacks by attackers possessing moderate attack capability.

#### A.1.101 AVA\_VLA-L.4 Highly resistant

Objectives. A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks using publicly known attacks and otherwise obvious attack methods. The developer performs penetration testing, supported by a vulnerability analysis, to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential.

#### Dependencies:

ADV\_FSP-L.1 Informal functional specification  
ADV\_HLD-L.2 Security enforcing high-level design  
ADV\_IMP-L.1 Subset of the implementation of the TSF  
ADV\_LLD-L.1 Descriptive low-level design  
ADV\_INT-L.3 Minimization of complexity  
AGD\_ADM-L.1 Administrator guidance  
AGD\_USR-L.1 User guidance

FPT\_RVM.1 Non-Bypassability of the TSP  
FPT\_SEP.3 Complete reference monitor

Developer action elements:

AVA\_VLA-L.4.1D The developer shall perform ~~and document~~ a systematic analysis of the TOE that completely addresses TOE deliverables searching for ways in which a user possessing a high attack potential can violate the TSP.

AVA\_VLA-L.4.2D The developer shall conduct penetration testing based upon this analysis to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential ~~document the disposition of identified vulnerabilities.~~

AVA\_VLA-L.4.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

Content and presentation of evidence elements:

~~AVA\_VLA-L.4.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.~~

~~AVA\_VLA-L.4.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.~~

~~AVA\_VLA-L.4.3C The evidence shall show that the search for vulnerabilities is systematic.~~

~~AVA\_VLA-L.4.4C The analysis documentation shall provide a justification that the analysis completely addresses the TOE deliverables.~~

None

Evaluator action elements:

AVA\_VLA-L.4.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing ~~information provided meets all requirements for content and presentation of evidence.~~

~~AVA\_VLA-L.4.2E CC element deleted The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.~~

AVA\_VLA-L.4.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods.

AVA\_VLA-L.4.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods ~~of additional identified vulnerabilities in the intended environment.~~

AVA\_VLA-L.4.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods ~~possessing a moderate attack potential~~.

AVA\_VLA-L.4.6E(+) The evaluator shall determine, to the level of rigor of appears reasonable, that the developer has performed penetration testing on the basis of a systematic vulnerability analysis, addressing all TOE deliverables, to determine that the TOE is resistant to penetration attacks by attackers possessing high attack capability.

#### A.1.102 AVA\_VLA-R.4 Highly resistant

Objectives. A vulnerability analysis is performed by the developer to ascertain the presence of security vulnerabilities, and to confirm that they cannot be exploited in the intended environment for the TOE. The evaluator performs independent penetration testing, supported by the evaluator's independent vulnerability analysis, to determine that the TOE is resistant to penetration attacks using publicly known attacks and otherwise obvious attack methods. The developer and the evaluator perform penetration testing, supported by a vulnerability analysis, to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential.

##### Dependencies:

ADV\_FSP-R.1 Informal functional specification  
ADV\_HLD-R.2 Security enforcing high-level design  
ADV\_IMP-R.1 Subset of the implementation of the TSF  
ADV\_LLD-R.1 Descriptive low-level design  
ADV\_INT-R.3 Minimization of complexity  
AGD\_ADM-R.1 Administrator guidance  
AGD\_USR-R.1 User guidance  
FPT\_RVM.1 Non-Bypassability of the TSP  
FPT\_SEP.3 Complete reference monitor

##### Developer action elements:

AVA\_VLA-R.4.1D The developer shall perform and document a systematic analysis of the TOE that completely addresses TOE deliverables searching for ways in which a user possessing a high attack potential can violate the TSP.

AVA\_VLA-R.4.2D The developer shall conduct penetration testing based upon this analysis to determine that the TOE is resistant to penetration attacks by attackers possessing a high attack potential document the disposition of identified vulnerabilities.

AVA\_VLA-R.4.3D(implied) The developer shall provide the TOE suitable for use by the evaluator in the conduct of penetration testing.

##### Content and presentation of evidence elements:

AVA\_VLA-R.4.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

AVA\_VLA-R.4.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA\_VLA-R.4.3C The evidence shall show that the search for vulnerabilities is systematic.

AVA\_VLA-R.4.4C The analysis documentation shall provide a justification that the analysis completely addresses the TOE deliverables.

Evaluator action elements:

AVA\_VLA-R.4.1E The evaluator shall confirm that the TOE supplied is suitable for use in penetration testing ~~information provided meets all requirements for content and presentation of evidence.~~

AVA\_VLA-R.4.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA\_VLA-R.4.3E The evaluator shall perform an independent vulnerability analysis with respect to publicly known attacks and otherwise obvious attack methods and with respect to attackers with high attack capability.

AVA\_VLA-R.4.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of the TOE to publicly known attacks and otherwise obvious attack methods and to attackers with high attack capability ~~of additional identified vulnerabilities in the intended environment.~~

AVA\_VLA-R.4.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker using publicly known attacks or otherwise obvious attack methods and by an attacker possessing a high attack potential.

AVA\_VLA-R.4.6E(+) The evaluator shall determine that the developer has performed penetration testing on the basis of a systematic vulnerability analysis to determine that the TOE is resistant to penetration attacks by attackers possessing high attack capability.

## APPENDIX B

## B

**B. REFERENCES**

## LAWS, POLICIES, REGULATIONS, STANDARDS, AND SPECIAL PUBLICATIONS

1. International Standard, ISO/IEC 15408, *Common Criteria for Information Technology Security Evaluation* (CC), October 1999.
2. Merriam-Webster, online dictionary,  
<http://www.m-w.com/cgi-bin/dictionary?book=Dictionary&va=assurance>
3. National Security Agency, Information Assurance Directorate, *Protection Profile for Single-level Operating Systems in Environments Requiring Medium Robustness*, Version 1.22, May 23, 2001,  
[http://www.iatf.net/protection\\_profiles/pdffile.cfm?chapter=SLMROSPPVer1\\_22](http://www.iatf.net/protection_profiles/pdffile.cfm?chapter=SLMROSPPVer1_22)
4. NIST, NISTIR 6462, *CSPP - Guidance for COTS Security Protection Profiles*, Version 1.0, December 1999, <http://csrc.nist.gov/publications/nistir/ir6462.pdf>
5. *Common Evaluation Methodology for Information Technology Security Evaluation* (CEM), Part 2, version 1.0, August 1999.

## APPENDIX C

## C

**C. GLOSSARY**

## COMMON TERMS ASSOCIATED WITH SECURITY REQUIREMENTS

Acceptable Risk	A concern that is acceptable to responsible management, due to the cost and magnitude of implementing security controls.
Accountability [SP 800-33]	The security objective that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports non-repudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action.
Adequate Security	Security commensurate with risk, including the magnitude of harm resulting from the loss, misuse, or unauthorized access to or modification of information.
Assurance [Common Criteria]	Grounds for confidence that an entity meets its security objectives.
Assurance [SP 800-33]	Grounds for confidence that the other four security objectives (integrity, availability, confidentiality, and accountability) have been adequately met by a specific implementation. “Adequately met” includes (1) functionality that performs correctly, (2) sufficient protection against unintentional errors (by users or software), and (3) sufficient resistance to intentional penetration or by-pass.
Authenticity	The property of being genuine and able to be verified and be trusted. See authentication; assurance of the validity of a transmission, message, or originator within an information system.
Authentication	Verifying the identity of a user, process, or device, often as a prerequisite to allowing access to resources in an information system.
Availability [44 U.S.C., SEC. 3542]	Ensuring timely and reliable access to and use of information.
Availability [SP 800-33]	The security objective that generates the requirement for protection against intentional or accidental attempts to (1) perform unauthorized deletion of data or (2) otherwise cause a denial of service or data.
Check [CEM]	To generate a verdict by a simple comparison.

Confidentiality [44 U.S.C., SEC. 3542]	Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.
Confidentiality [SP 800-33]	The security objective that generates the requirement for protection from intentional or accidental attempts to perform unauthorized data reads. Confidentiality covers data in storage, during processing, and while in transit.
Confirm	To generate a verdict by applying evaluator expertise to verify the actions of others.
Countermeasures	Synonymous with security controls and safeguards.
Data Integrity [SP 800-33]	The property that data has not been altered in an unauthorized manner. Data integrity covers data in storage, during processing, and while in transit.
Determine	To generate a verdict by applying evaluator expertise to perform actions and may also include verifying the actions of others.
Domain	See Security Domain
Entity [SP 800-33]	Either a subject (an active element that operates on information or the system state) or an object (a passive element that contains or receives information).
Federal Information System [40 U.S.C., SEC. 11331]	An information system used or operated by an executive agency, by a contractor of an executive agency, or by another organization on behalf of an executive agency.
Information Resources [44 U.S.C., SEC. 3502]	Information and related resources, such as personnel, equipment, funds, and information technology.
Information Security [44 U.S.C., SEC. 3542]	The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability.
Information System [44 U.S.C., SEC. 3502]	A discrete set of information resources organized for the collection, processing, maintenance, use, sharing, dissemination, or disposition of information.

Information Technology [40 U.S.C., SEC. 1401]	Any equipment or interconnected system or subsystem of equipment that is used in the automatic acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information by the executive agency. For purposes of the preceding sentence, equipment is used by an executive agency if the equipment is used by the executive agency directly or is used by a contractor under a contract with the executive agency which: (i) requires the use of such equipment, or (ii) requires the use, to a significant extent, of such equipment in the performance of a service or the furnishing of a product. The term information technology includes computers, ancillary equipment, software, firmware and similar procedures, services (including support services), and related resources.
Integrity [44 U.S.C., Sec. 3542]	Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity.
Integrity [SP 800-33]	The security objective that generates the requirement for protection against either intentional or accidental attempts to violate data integrity (the property that data has not been altered in an unauthorized manner) or system integrity (the quality that a system has when it performs its intended function in an unimpaired manner, free from unauthorized manipulation).
Identity [SP 800-33]	Information that is unique within a security domain and which is recognized as denoting a particular entity within that domain.
IT-Related Risk [SP 800-30A]	<p>The level of impact on agency operations (including mission, functions, image, or reputation), agency assets, or individuals considering (1) the probability that a particular threat-source will exercise (accidentally trigger or intentionally exploit) a particular information system vulnerability and (2) the resulting impact if this should occur. IT-related risks arise from legal liability or loss due to:</p> <ol style="list-style-type: none"> <li>1. Unauthorized (malicious, non-malicious, or accidental) disclosure, modification, or destruction of information.</li> <li>2. Non-malicious errors and omissions.</li> <li>3. IT disruptions due to natural or man-made disasters.</li> <li>4. Failure to exercise due care and diligence in the implementation and operation of the IT.</li> </ol>
IT Security Architecture [SP 800-33]	A description of security principles and an overall approach for complying with the principles that drive the system design; i.e., guidelines on the placement and implementation of specific security services within various distributed computing environments.
IT Security Objective	See Security Objective



National Security Information	Information that has been determined pursuant to Executive Order 12958 or any predecessor order, or by the Atomic Energy Act of 1954, as amended, to require protection against unauthorized disclosure and is marked to indicate its classified status.
National Security System [44 U.S.C., SEC. 3542]	Any information system (including any telecommunications system) used or operated by an agency or by a contractor of an agency, or other organization on behalf of an agency— (i) the function, operation, or use of which: involves intelligence activities; involves cryptologic activities related to national security; involves command and control of military forces; involves equipment that is an integral part of a weapon or weapons system; or is critical to the direct fulfillment of military or intelligence missions (excluding a system that is to be used for routine administrative and business applications, for example, payroll, finance, logistics, and personnel management applications); or, (ii) Is protected at all times by procedures established for information that have been specifically authorized under criteria established by an Executive order or an Act of Congress to be kept classified in the interest of national defense or foreign policy.
Non-repudiation	Assurance that the sender of information is provided with proof of delivery and the recipient is provided with proof of the sender's identity, so neither can later legitimately deny having processed, stored, or transmitted the information.
Object [SP 800-33]	A passive entity that contains or receives information. Note that access to an object potentially implies access to the information it contains
Reference Monitor [SP 800-33]	The security engineering term for IT functionality that (1) controls all access, (2) cannot be by-passed, (3) is tamper-resistant, and (4) provides confidence that the other three items are true.
Residual Risk	The portion of risk remaining after the application of appropriate security controls in the information system.
Risk [SP 800-30A]	The level of impact on agency operations (including mission, functions, image, or reputation), agency assets, or individuals resulting from the operation of an information system given the potential-impact of a threat and the probability of that threat occurring. (Conceptually, risk = potential-impact x probability).
Risk Assessment [SP 800-30A]	The process of identifying risks to agency operations (including mission, functions, image, or reputation), agency assets, or individuals by determining the probability of occurrence, the resulting impact, and additional security controls that would mitigate this impact. Part of risk management, synonymous with risk analysis, and incorporates threat and vulnerability analyses.

Risk Management [SP 800-30A]	The process of identifying, controlling, and mitigating risks to agency operations (including mission, functions, image, or reputation), agency assets, or individuals resulting from the operation of an information system or multiple information systems. It includes: risk assessment, cost benefit analysis, and the selection, implementation, testing and evaluation of security controls.
Rule-based security policy [SP 800-33]	A security policy based on global rules imposed for all subjects. These rules usually rely on a comparison of the sensitivity of the objects being accessed and the possession of corresponding attributes by the subjects requesting access.
Safeguards	Synonymous with security controls and countermeasures.
Security [SP 800-33]	Security is a system property. Security is much more than a set of functions and mechanisms. Information technology security is a system characteristic as well as a set of mechanisms which span the system both logically and physically.
Security Controls	The management, operational, and technical controls (safeguards or countermeasures) prescribed for an information system which, taken together, satisfy the specified security requirements and adequately protect the confidentiality, integrity, and availability of the system and its information.
Security Domain [SP 800-33]	A set of subjects, their information objects, and a common security policy.
Security Goal [SP 800-33]	The IT security goal is to enable an organization to meet all mission/business objectives by implementing systems with due care consideration of IT-related risks to the organization, its partners, and its customers.
Security Objectives [SP 800-33]	The five security objectives are integrity, availability, confidentiality, accountability, and assurance.
Security Plan	Formal document that provides an overview of the security requirements of the information system and describes the security controls in place or planned for meeting those requirements.
Security Policy [SP 800-33]	The statement of required protection of the information objects.
Subject [SP 800-33]	An active entity, generally in the form of a person, process, or device, that causes information to flow among objects or changes the system state.
Subsystem	A major subdivision or component of an information system consisting of hardware, software, or firmware that performs a specific function.

System Integrity [SP 800-33]	The quality that a system has when it performs its intended function in an unimpaired manner, free from unauthorized manipulation of the system, whether intentional or accidental.
Threat [SP 800-30A]	<p>Any circumstance or event with the potential to adversely impact agency operations (including mission, functions, image, or reputation), agency assets, or individuals through an information system via unauthorized access, destruction, disclosure, modification of information and/or denial of service.</p> <p>Alternate: The potential for a threat-source to exercise (accidentally trigger or intentionally exploit) a specific vulnerability.</p>
Vulnerability	A flaw or weakness in the design or implementation of an information system (including the security procedures and security controls associated with the system) that could be intentionally or unintentionally exploited to adversely effect an agency's operations (including mission, functions, image, or reputation), an agency's assets, or individuals (including privacy) through a loss of confidentiality, integrity, or availability.

DRAFT

## APPENDIX D

## D

**D. ACRONYMS**

## SHORTHAND NOTATIONS FOR ASSURANCE-REQUIREMENT-RELATED TERMS

CC	Common Criteria
CM	Configuration Management
COTS	Commercial Off The Shelf
EAL	Evaluation Assurance Level (CC terminology)
FISMA	Federal Information Security Management Act
IT	Information Technology
NIST	National Institute of Standards and Technology
OMB	Office of Management and Budget
PP	Protection Profile, a CC defined, implementation independent requirements document format
ST	Security Target, a CC defined implementation dependent requirements document format
SPM	TOE Security Policy Model (CC terminology)
TOE	Target of Evaluation, the IT for which requirements are being specified (CC terminology)
TSF	TOE Security Functions (CC terminology)
TSP	TOE Security Policy (CC terminology)
U.S.C.	United States Code