Common Criteria

# Common Criteria
# for Information Technology
# Security Evaluation

---

Part 2: Security functional requirements

August 2005

Version 2.3

CCMB-2005-08-002

# Foreword

This version of the Common Criteria for Information Technology Security Evaluation (CC v2.3) contains all final interpretations, editorial changes, and agreed new material since the publication of CC v2.1, in alignment with International Standard ISO/IEC 15408:2005.

CC version 2.3 consists of the following parts:

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

# Table of Contents

## Table of contents

## Table of contents

## Table of contents

# List of figures

# List of tables

# 1 Introduction

1 Security functional components, as defined in this CC Part 2, are the basis for the security functional requirements expressed in a Protection Profile (PP) or a Security Target (ST). These requirements describe the desired security behaviour expected of a Target of Evaluation (TOE) or the IT environment of the TOE and are intended to meet the security objectives as stated in a PP or an ST. These requirements describe security properties that users can detect by direct interaction (i.e. inputs, outputs) with the IT or by the IT response to stimulus.

2 Security functional components express security requirements intended to counter threats in the assumed operating environment of the TOE and/or cover any identified organisational security policies and assumptions.

3 The audience for this CC Part 2 includes consumers, developers, and evaluators of secure IT systems and products. CC Part 1 Chapter 6 provides additional information on the target audience of the CC, and on the use of the CC by the groups that comprise the target audience. These groups may use this part of the CC as follows:

a) Consumers, who use this CC Part 2 when selecting components to express functional requirements to satisfy the security objectives expressed in a PP or ST. CC Part 1 Section 6.3 provides more detailed information on the relationship between security objectives and security requirements.

b) Developers, who respond to actual or perceived consumer security requirements in constructing a TOE, may find a standardised method to understand those requirements in this part of the CC. They can also use the contents of this part of the CC as a basis for further defining the TOE security functions and mechanisms that comply with those requirements.

c) Evaluators, who use the functional requirements defined in this part of the CC in verifying that the TOE functional requirements expressed in the PP or ST satisfy the IT security objectives and that all dependencies are accounted for and shown to be satisfied. Evaluators also should use this part of the CC to assist in determining whether a given TOE satisfies stated requirements.

# 2 Scope

4 This part of the CC defines the required structure and content of security functional components for the purpose of security evaluation. It includes a catalogue of functional components that will meet the common security functionality requirements of many IT products and systems.

# 3 Normative references

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

CC Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005. Part 1: Introduction and general model.

# 4 Terms and definitions, symbols and abbreviated terms

6 For the purposes of this document, the terms, definitions, symbols and abbreviated terms given in CC Part 1 apply.

# 5 Overview

7 The CC and the associated security functional requirements described herein are not meant to be a definitive answer to all the problems of IT security. Rather, the CC offers a set of well understood security functional requirements that can be used to create trusted products or systems reflecting the needs of the market. These security functional requirements are presented as the current state of the art in requirements specification and evaluation.

8 This part of the CC does not presume to include all possible security functional requirements but rather contains those that are known and agreed to be of value by the CC Part 2 authors at the time of release.

9 Since the understanding and needs of consumers may change, the functional requirements in this part of the CC will need to be maintained. It is envisioned that some PP/ST authors may have security needs not (yet) covered by the functional requirement components in CC Part 2. In those cases the PP/ST author may choose to consider using functional requirements not taken from the CC (referred to as extensibility), as explained in annexes A and B of CC Part 1.

## 5.1 Organisation of CC Part 2

10 Chapter 6 describes the paradigm used in the security functional requirements of CC Part 2.

11 Chapter 7 introduces the catalogue of CC Part 2 functional components while chapters 8 through 18 describe the functional classes.

12 Annex A provides explanatory information for potential users of the functional components including a complete cross reference table of the functional component dependencies.

13 Annex B through M provide the explanatory information for the functional classes. This material must be seen as normative instructions on how to apply relevant operations and select appropriate audit or documentation information; the use of the auxiliary verb should means that the instruction is strongly preferred, but others may be justifiable. Where different options are given, the choice is left to the PP/ST author.

14 Those who author PPs or STs should refer to chapter 2 of CC Part 1 for relevant structures, rules, and guidance:

a) CC Part 1, chapter 3 defines the terms used in the CC.

b) CC Part 1, annex A defines the structure for PPs.

c) CC Part 1, annex B defines the structure for STs.

d) CC Part 1, annex C contains a bibliography of relevant reference documents.

# 6 Functional requirements paradigm

15 This chapter describes the paradigm used in the security functional requirements of this part of the CC. Figures 1 and 2 depict some of the key concepts of the paradigm. This section provides descriptive text for those figures and for other key concepts not depicted. Key concepts discussed are highlighted in bold/italics. This section is not intended to replace or supersede any of the terms found in CC Part 1, chapter 3.

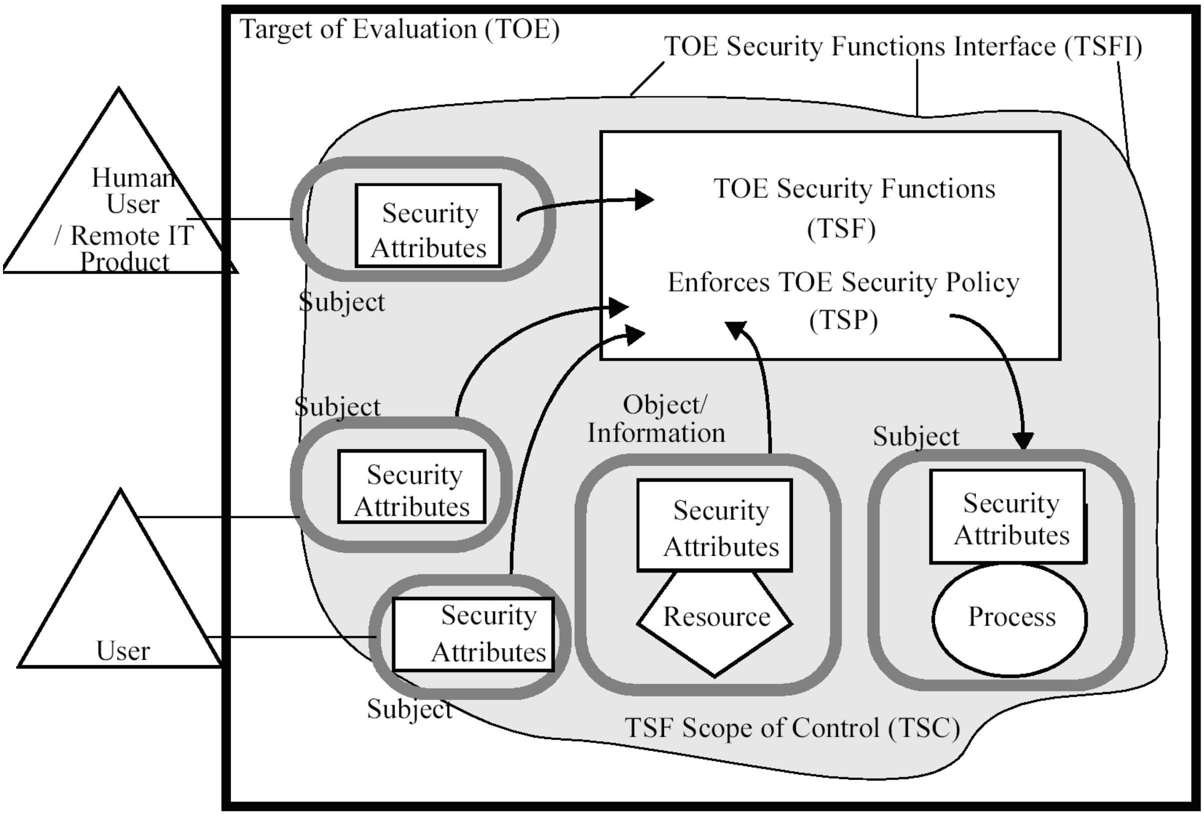



**Figure 1 - Security functional requirements paradigm (Monolithic TOE)**

16 This part of the CC is a catalogue of security functional requirements that can be specified for a **Target of Evaluation (TOE)**. A TOE is an IT product or system (along with user and administrator guidance documentation) containing resources such as electronic storage media (e.g. disks), peripheral devices (e.g. printers), and computing capacity (e.g. CPU time) that can be used for processing and storing information and is the subject of an evaluation.

17 TOE evaluation is concerned primarily with ensuring that a defined **TOE Security Policy (TSP)** is enforced over the TOE resources. The TSP defines the rules by which the TOE governs access to its resources, and thus all information and services controlled by the TOE.

18 The TSP is, in turn, made up of multiple **Security Function Policies (SFPs)**. Each SFP has a scope of control, that defines the subjects, objects, and operations controlled under the SFP. The SFP is implemented by a **Security Function (SF)**, whose mechanisms enforce the policy and provide necessary capabilities.

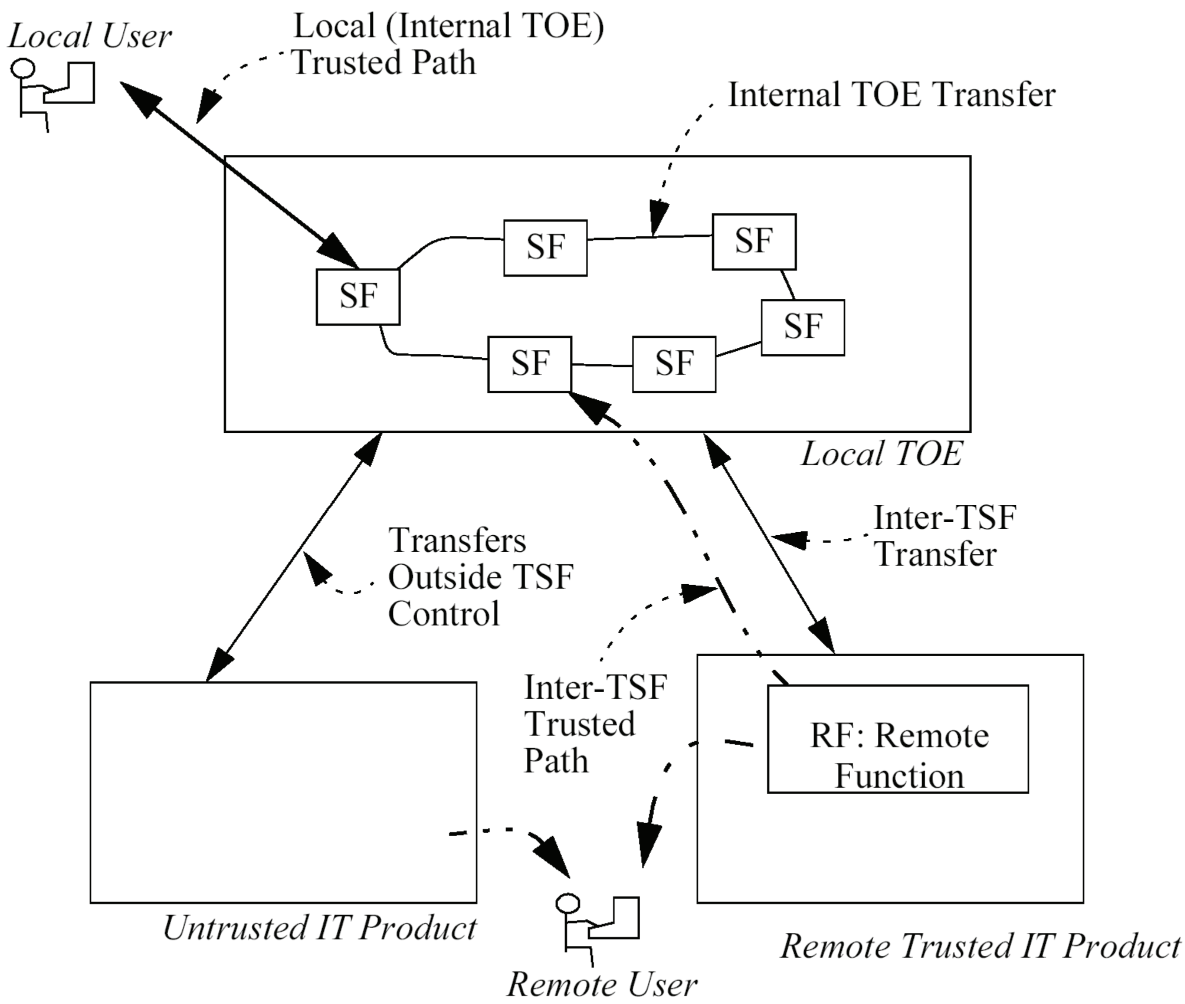



**Figure 2 - Diagram of security functions in a distributed TOE**

19 Those portions of a TOE that must be relied on for the correct enforcement of the TSP are collectively referred to as the **TOE Security Functions (TSF)**. The TSF consists of all hardware, software, and firmware of a TOE that is either directly or indirectly relied upon for security enforcement.

20 A **reference monitor** is an abstract machine that enforces the access control policies of a TOE. A **reference validation mechanism** is an implementation of the reference monitor concept that possesses the following properties: tamperproof, always invoked, and simple enough to be subjected to thorough analysis and testing. The **TSF** may consist of a reference validation mechanism and/or other security functions necessary for the operation of the TOE.

21 The TOE may be a monolithic product containing hardware, firmware, and software.

22 Alternatively a TOE may be a distributed product that consists internally of multiple separated parts. Each of these parts of the TOE provides a particular service for the TOE, and is connected to the other parts of the TOE through an **internal communication channel**. This channel can be as small as a processor bus, or may encompass a network internal to the TOE.

23 When the TOE consists of multiple parts, each part of the TOE may have its own part of the TSF which exchanges user and TSF data over internal communication channels with other parts of the TSF. This interaction is called **internal TOE transfer**. In this case the separate parts of the TSF abstractly form the composite TSF, which enforces the TSP.

24 TOE interfaces may be localised to the particular TOE, or they may allow interaction with other IT products over **external communication channels**. These external interactions with other IT products may take two forms:

a) The security policy of the “remote trusted IT product” and the TSP of the local TOEs have been administratively coordinated and evaluated. Exchanges of information in this situation are called **inter-TSF transfers**, as they are between the TSFs of distinct trusted products.

b) The remote IT product may not be evaluated, indicated in Figure 2 as “untrusted IT product”, therefore its security policy is unknown. Exchanges of information in this situation are called **transfers outside TSF control**, as there is no TSF (or its policy characteristics are unknown) on the remote IT product.

25 The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP is called the **TSF Scope of Control (TSC)**. The TSC encompasses a defined set of interactions based on subjects, objects, and operations within the TOE, but it need not encompass all resources of a TOE.

26 The set of interfaces, whether interactive (man-machine interface) or programmatic (application programming interface), through which resources are accessed that are mediated by the TSF, or information is obtained from the TSF, is referred to as the **TSF Interface (TSFI)**. The TSFI defines the boundaries of the TOE functions that provide for the enforcement of the TSP.

27 Users are outside of the TOE, and therefore outside of the TSC. However, in order to request that services be performed by the TOE, users interact with the TOE through the TSFI. There are two types of users of interest to the CC Part 2 security functional requirements: **human users** and **external IT entities**. Human users are further differentiated as **local human users**, meaning they interact directly with the TOE via TOE devices (e.g. workstations), or **remote human users**, meaning they interact indirectly with the TOE through another IT product.

28 A period of interaction between users and the TSF is referred to as a user **session**. Establishment of user sessions can be controlled based on a variety of considerations, for example: user authentication, time of day, method of accessing the TOE, and number of allowed concurrent sessions per user.

29 This part of the CC uses the term **authorised** to signify a user who possesses the rights and/or privileges necessary to perform an operation. The term **authorised user**, therefore, indicates that it is allowable for a user to perform an operation as defined by the TSP.

30 To express requirements that call for the separation of administrator duties, the relevant CC Part 2 security functional components (from family FMT_SMR) explicitly state that administrative **roles** are required. A role is a pre-defined set of rules establishing the allowed interactions between a user and the TOE. A TOE may support the definition of any number of roles. For example, roles related to the secure operation of a TOE may include "Audit Administrator" and "User Accounts Administrator".

31 TOEs contain resources that may be used for the processing and storing of information. The primary goal of the TSF is the complete and correct enforcement of the TSP over the resources and information that the TOE controls.

32 TOE resources can be structured and utilised in many different ways. However, CC Part 2 makes a specific distinction that allows for the specification of desired security properties. All entities that can be created from resources can be characterised in one of two ways. The entities may be active, meaning that they are the cause of actions that occur internal to the TOE and cause operations to be performed on information. Alternatively, the entities may be passive, meaning that they are either the container from which information originates or to which information is stored.

33 Active entities are referred to as **subjects**. Several types of subjects may exist within a TOE:

a) those acting on behalf of an authorised user and which are subject to all the rules of the TSP (e.g. UNIX processes);

b) those acting as a specific functional process that may in turn act on behalf of multiple users (e.g. functions as might be found in client/server architectures); or

c) those acting as part of the TOE itself (e.g. trusted processes).

34 CC Part 2 addresses the enforcement of the TSP over types of subjects as those listed above.

35 Passive entities (i.e. information containers) are referred to in the CC Part 2 security functional requirements as **objects**. Objects are the targets of operations that may be performed by subjects. In the case where a subject (an

active entity) is the target of an operation (e.g. interprocess communication), a subject may also be acted on as an object.

36 Objects can contain **information**. This concept is required to specify information flow control policies as addressed in the FDP class.

37 Users, subjects, information and objects possess certain **attributes** that contain information that allows the TOE to behave correctly. Some attributes, such as file names, may be intended to be informational (i.e. to increase the user-friendliness of the TOE) while others, such as access control information, may exist specifically for the enforcement of the TSP. These latter attributes are generally referred to as "**security attributes**". The word attribute will be used as a shorthand in this part of the CC for the word "security attribute", unless otherwise indicated. However, no matter what the intended purpose of the attribute information, it may be necessary to have controls on attributes as dictated by the TSP.

38 Data in a TOE is categorised as either user data or TSF data. Figure 3 depicts this relationship. **User Data** is information stored in TOE resources that can be operated upon by users in accordance with the TSP and upon which the TSF places no special meaning. For example, the contents of an electronic mail message is user data. **TSF Data** is information used by the TSF in making TSP decisions. TSF Data may be influenced by users if allowed by the TSP. Security attributes, authentication data and access control list entries are examples of TSF data.

39 There are several SFPs that apply to data protection such as **access control SFPs** and **information flow control SFPs**. The mechanisms that implement access control SFPs base their policy decisions on attributes of the subjects, objects and operations within the scope of control. These attributes are used in the set of rules that govern operations that subjects may perform on objects.

40 The mechanisms that implement information flow control SFPs base their policy decisions on the attributes of the subjects and information within the scope of control and the set of rules that govern the operations by subjects on information. The attributes of the information, which may be associated with the attributes of the container (or may not, as in the case of a multi-level database) stay with the information as it moves.

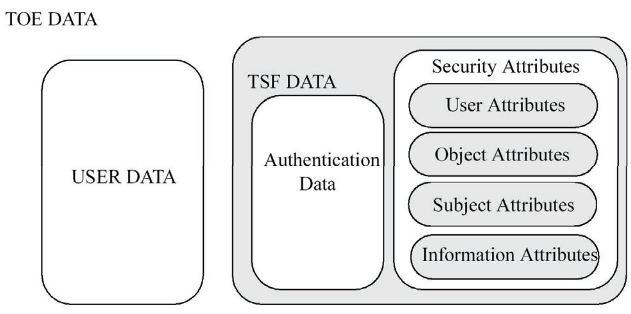



**Figure 3 - Relationship between user data and TSF data**

41 Two specific types of TSF data addressed by CC Part 2 can be, but are not necessarily, the same. These are **authentication data and secrets**.

42 Authentication data is used to verify the claimed identity of a user requesting services from a TOE. The most common form of authentication data is the password, which depends on being kept secret in order to be an effective security mechanism. However, not all forms of authentication data need to be kept secret. Biometric authentication devices (e.g. fingerprint readers, retinal scanners) do not rely on the fact that the data is kept secret, but rather that the data is something that only one user possesses and that cannot be forged.

43 The term secrets, as used in CC Part 2 functional requirements, while applicable to authentication data, is intended to also be applicable to other types of data that must be kept secret in order to enforce a specific SFP. For example, a trusted channel mechanism that relies on cryptography to preserve the confidentiality of information being transmitted via the channel can only be as strong as the method used to keep the cryptographic keys secret from unauthorised disclosure.

44 Therefore, some, but not all, authentication data needs to be kept secret and some, but not all, secrets are used as authentication data. Figure 4 shows this relationship between secrets and authentication data. In the Figure the types of data typically encountered in the authentication data and the secrets sections are indicated.

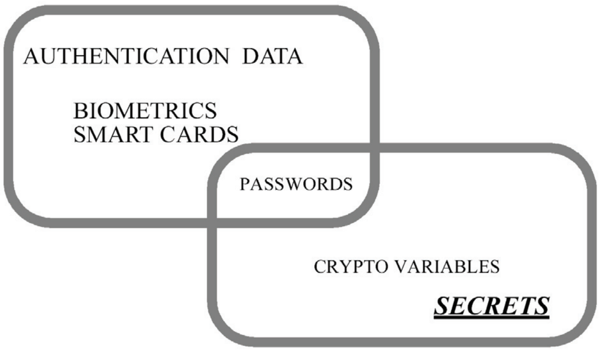



**Figure 4 - Relationship between “authentication data” and “secrets”**

# 7 Security functional components

## 7.1 Overview

45 This chapter defines the content and presentation of the functional requirements of the CC, and provides guidance on the organisation of the requirements for new components to be included in an ST. The functional requirements are expressed in classes, families, and components.

### 7.1.1 Class structure

46 Figure 5 illustrates the functional class structure in diagrammatic form. Each functional class includes a class name, class introduction, and one or more functional families.

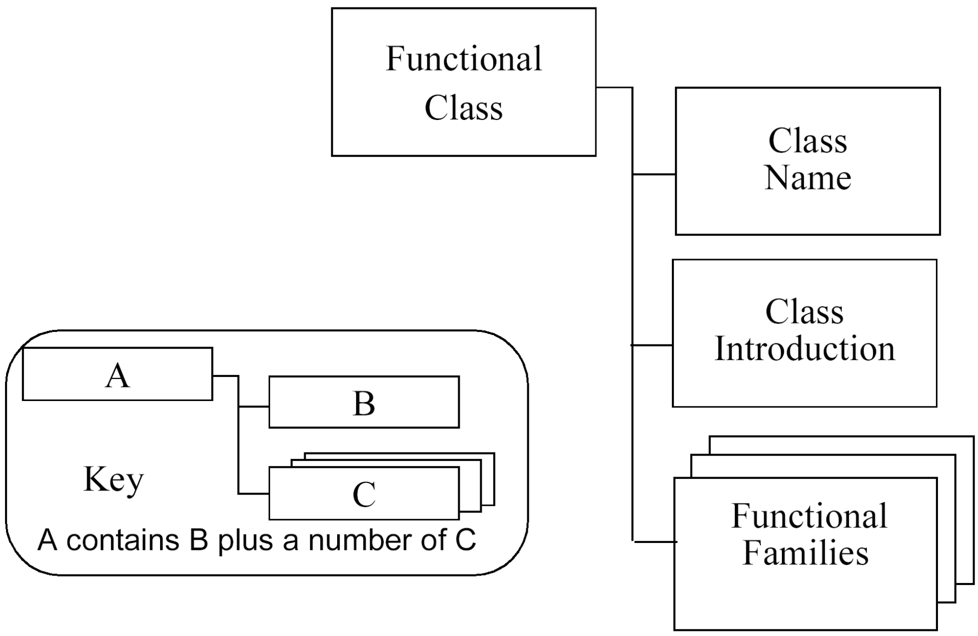



**Figure 5 - Functional class structure**

#### 7.1.1.1 Class name

47 The class name section provides information necessary to identify and categorise a functional class. Every functional class has a unique name. The categorical information consists of a short name of three characters. The short name of the class is used in the specification of the short names of the families of that class.

#### 7.1.1.2 Class introduction

48 The class introduction expresses the common intent or approach of those families to satisfy security objectives. The definition of functional classes does not reflect any formal taxonomy in the specification of the requirements.

49 The class introduction provides a figure describing the families in this class and the hierarchy of the components in each family, as explained in section 7.2.

### 7.1.2 Family structure

50 Figure 6 illustrates the functional family structure in diagrammatic form.

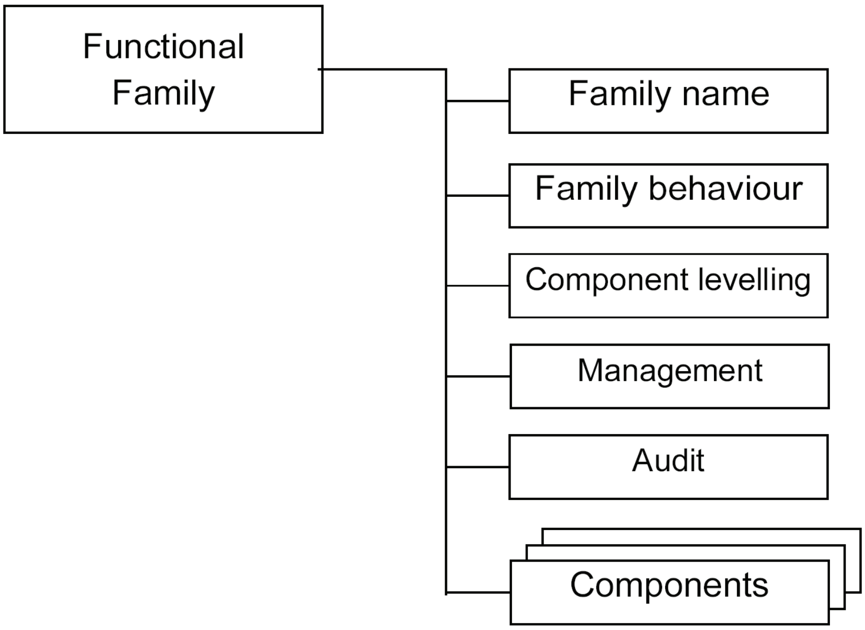



**Figure 6 - Functional family structure**

#### 7.1.2.1 Family name

51 The family name section provides categorical and descriptive information necessary to identify and categorise a functional family. Every functional family has a unique name. The categorical information consists of a short name of seven characters, with the first three identical to the short name of the class followed by an underscore and the short name of the family as follows XXX_YYY. The unique short form of the family name provides the principal reference name for the components.

#### 7.1.2.2 Family behaviour

52 The family behaviour is the narrative description of the functional family stating its security objective and a general description of the functional requirements. These are described in greater detail below:

a) The *security objectives* of the family address a security problem that may be solved with the help of a TOE that incorporates a component of this family;

b) The description of the *functional requirements* summarises all the requirements that are included in the component(s). The description is aimed at authors of PPs, STs and functional packages who wish to assess whether the family is relevant to their specific requirements.

#### 7.1.2.3 Component levelling

53 Functional families contain one or more components, any one of which can be selected for inclusion in PPs, STs and functional packages. The goal of this section is to provide information to users in selecting an appropriate functional component once the family has been identified as being a necessary or useful part of their security requirements.

54 This section of the functional family description describes the components available, and their rationale. The exact details of the components are contained within each component.

55 The relationships between components within a functional family may or may not be hierarchical. A component is hierarchical to another if it offers more security.

56 As explained in 7.2 the descriptions of the families provide a graphical overview of the hierarchy of the components in a family.

#### 7.1.2.4 Management

57 The *management* requirements contain information for the PP/ST authors to consider as management activities for a given component. The management requirements are detailed in components of the management class (FMT).

58 A PP/ST author may select the indicated management requirements or may include other management requirements not listed. As such the information should be considered informative.

#### 7.1.2.5 Audit

59 The *audit* requirements contain auditable events for the PP/ST authors to select, if requirements from the class FAU: Security audit, are included in the PP/ST. These requirements include security relevant events in terms of the various levels of detail supported by the components of the Security audit data generation (FAU_GEN) family. For example, an audit note might include actions that are in terms of: Minimal - successful use of the security mechanism; Basic - any use of the security mechanism as well as relevant information regarding the security attributes involved; Detailed - any configuration changes made to the mechanism, including the actual configuration values before and after the change.

60 It should be observed that the categorisation of auditable events is hierarchical. For example, when Basic Audit Generation is desired, all auditable events identified as being both Minimal and Basic should be included in the PP/ST through the use of the appropriate assignment operation, except when the higher level event simply provides more detail than the lower level event. When Detailed Audit Generation is desired, all

identified auditable events (Minimal, Basic and Detailed) should be included in the PP/ST.

61 In the class FAU: Security audit the rules governing the audit are explained in more detail.

### 7.1.3 Component structure

62 Figure 7 illustrates the functional component structure.

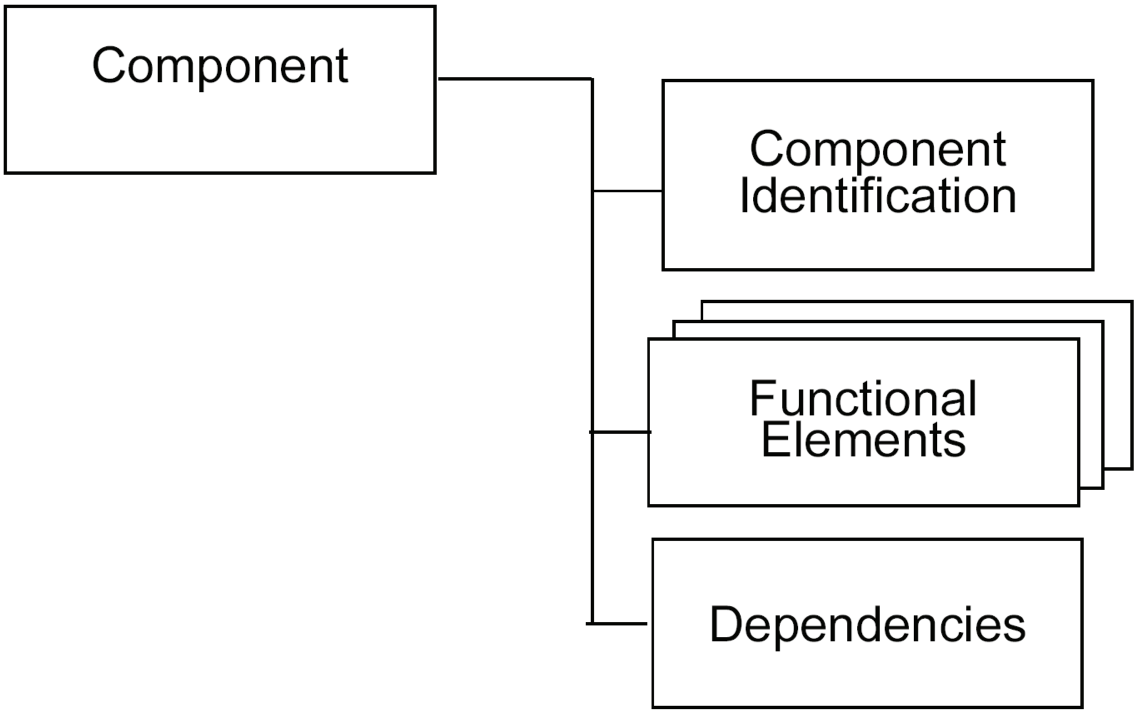



**Figure 7 - Functional component structure**

#### 7.1.3.1 Component identification

63 The component identification section provides descriptive information necessary to identify, categorise, register and cross-reference a component. The following is provided as part of every functional component:

64 A *unique name*. The name reflects the purpose of the component.

65 A *short name*. A unique short form of the functional component name. This short name serves as the principal reference name for the categorisation, registration and cross-referencing of the component. This short name reflects the class and family to which the component belongs and the component number within the family.

66 A *hierarchical-to list*. A list of other components that this component is hierarchical to and for which this component can be used to satisfy dependencies to the listed components.

### 7.1.3.2 Functional elements

67 A set of elements is provided for each component. Each element is individually defined and is self-contained.

68 A functional element is a security functional requirement that if further divided would not yield a meaningful evaluation result. It is the smallest security functional requirement identified and recognised in the CC.

69 When building packages, PPs and/or STs, it is not permitted to select only one or more elements from a component. The complete set of elements of a component must be selected for inclusion in a PP, ST or package.

70 A unique short form of the functional element name is provided. For example the requirement name FDP_IFF.4.2 reads as follows: F - functional requirement, DP - class "User data protection", _IFF - family "Information flow control functions", .4 - 4th component named "Partial elimination of illicit information flows", .2 - 2nd element of the component.

### 7.1.3.3 Dependencies

71 Dependencies among functional components arise when a component is not self sufficient and relies upon the functionality of, or interaction with, another component for its own proper functioning.

72 Each functional component provides a complete list of dependencies to other functional and assurance components. Some components may list "No dependencies". The components depended upon may in turn have dependencies on other components. The list provided in the components will be the direct dependencies. That is only references to the functional requirements that are required for this requirement to perform its job properly. The indirect dependencies, that is the dependencies that result from the depended upon components can be found in Annex A of this part of the CC. It is noted that in some cases the dependency is optional in that a number of functional requirements are provided, where each one of them would be sufficient to satisfy the dependency (see for example FDP_UIT.1 Data exchange integrity).

73 The dependency list identifies the minimum functional or assurance components needed to satisfy the security requirements associated with an identified component. Components that are hierarchical to the identified component may also be used to satisfy the dependency.

74 The dependencies indicated in CC Part 2 are normative. They must be satisfied within a PP/ST. In specific situations the indicated dependencies might not be applicable. The PP/ST author, by providing the rationale why it is not applicable, may leave the depended upon component out of the package, PP or ST.

## 7.2 Component catalogue

75 The grouping of the components in this part of the CC does not reflect any formal taxonomy.

76 This part of the CC contains classes of families and components, which are rough groupings on the basis of related function or purpose, presented in alphabetic order. At the start of each class is an informative diagram that indicates the taxonomy of each class, indicating the families in each class and the components in each family. The diagram is a useful indicator of the hierarchical relationship that may exist between components.

77 In the description of the functional components, a section identifies the dependencies between the component and any other components.

78 In each class a figure describing the family hierarchy similar to Figure 8, is provided. In Figure 8 the first family, Family 1, contains three hierarchical components, where component 2 and component 3 can both be used to satisfy dependencies on component 1. Component 3 is hierarchical to component 2 and can also be used to satisfy dependencies on component 2.

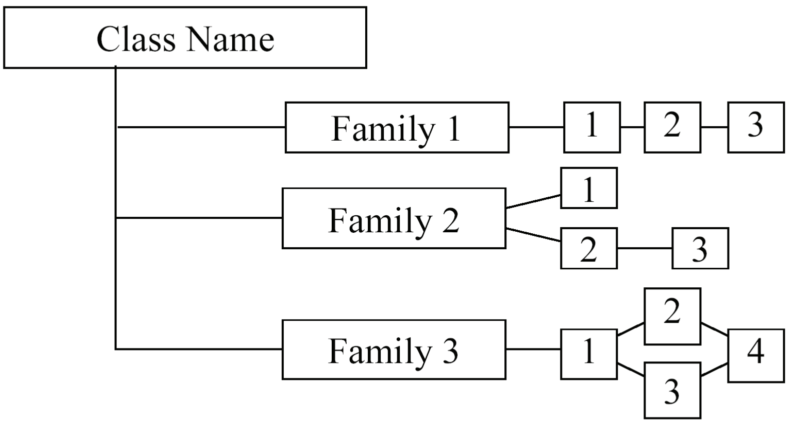



**Figure 8 - Sample class decomposition diagram**

79 In Family 2 there are three components not all of which are hierarchical. Components 1 and 2 are hierarchical to no other components. Component 3 is hierarchical to component 2, and can be used to satisfy dependencies on component 2, but not to satisfy dependencies on component 1.

80 In Family 3, components 2, 3, and 4 are hierarchical to component 1. Components 2 and 3 are both hierarchical to component 1, but non-comparable. Component 4 is hierarchical to both component 2 and component 3.

81 These diagrams are meant to complement the text of the families and make identification of the relationships easier. They do not replace the "Hierarchical to:" note in each component that is the mandatory claim of hierarchy for each component.

### 7.2.1 Component changes highlighting

82 The relationship between components within a family is highlighted using a **bolding** convention. This bolding convention calls for the bolding of all new requirements. For hierarchical components, requirements are bolded when they are enhanced or modified beyond the requirements of the previous component. In addition, any new or enhanced permitted operations beyond the previous component are also highlighted using **bold** type.

# 8 Class FAU: Security audit

83 Security auditing involves recognising, recording, storing, and analysing information related to security relevant activities (i.e. activities controlled by the TSP). The resulting audit records can be examined to determine which security relevant activities took place and whom (which user) is responsible for them.

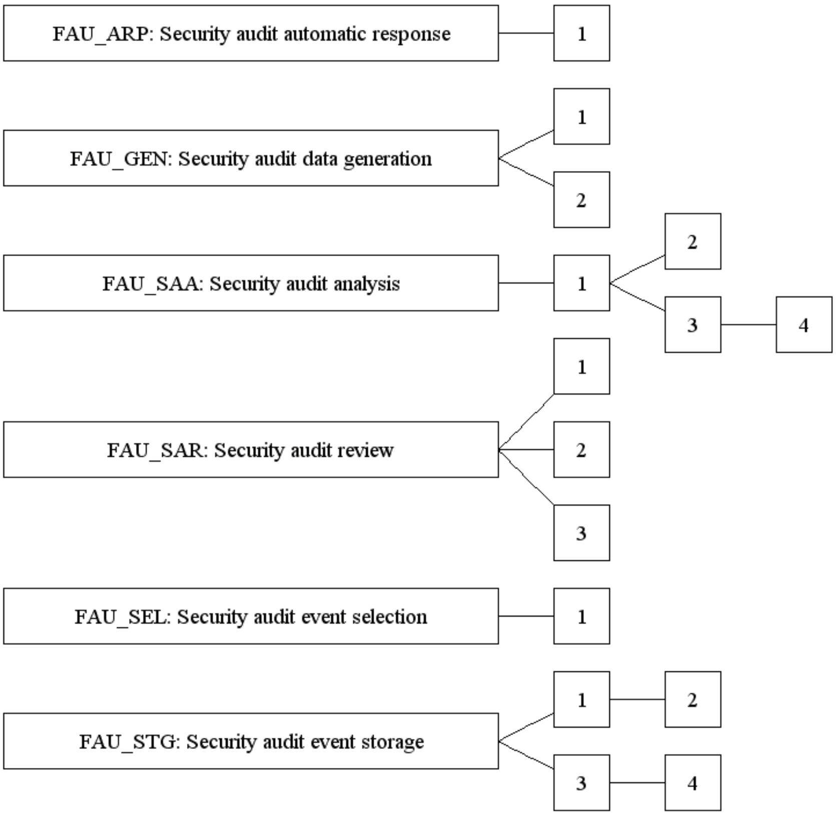



**Figure 9 - FAU: Security audit class decomposition**

## 8.1 Security audit automatic response (FAU_ARP)

### Family Behaviour

84 This family defines the response to be taken in case of detected events indicative of a potential security violation.

### Component levelling


85 At FAU_ARP.1 Security alarms, the TSF shall take actions in case a potential security violation is detected.

### Management: FAU_ARP.1

86 The following actions could be considered for the management functions in FMT:

a) the management (addition, removal, or modification) of actions.

### Audit: FAU_ARP.1

87 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Actions taken due to imminent security violations.

## FAU_ARP.1 Security alarms

Hierarchical to: No other components.

Dependencies: FAU_SAA.1 Potential violation analysis

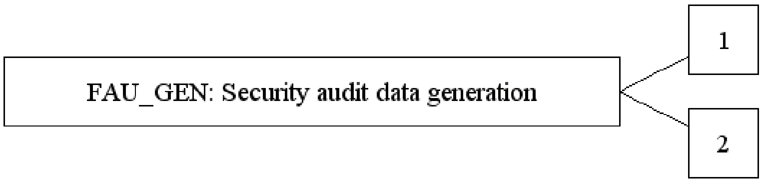
FAU_ARP.1.1 **The TSF shall take [assignment: *list of the least disruptive actions*] upon detection of a potential security violation.**

## 8.2 Security audit data generation (FAU_GEN)

### Family Behaviour

88 This family defines requirements for recording the occurrence of security relevant events that take place under TSF control. This family identifies the level of auditing, enumerates the types of events that shall be auditable by the TSF, and identifies the minimum set of audit-related information that should be provided within various audit record types.

### Component levelling




89 FAU_GEN.1 Audit data generation defines the level of auditable events, and specifies the list of data that shall be recorded in each record.

90 At FAU_GEN.2 User identity association, the TSF shall associate auditable events to individual user identities.

### Management:FAU_GEN.1, FAU_GEN.2

91 There are no management activities foreseen.

### Audit: FAU_GEN.1, FAU_GEN.2

92 There are no auditable events foreseen.

### FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

FAU_GEN.1.1 **The TSF shall be able to generate an audit record of the following auditable events:**

**a) Start-up and shutdown of the audit functions;**

**b) All auditable events for the [selection, choose one of: *minimum,basic,detailed,not specified*] level of audit; and**

**c) [assignment: *other specifically defined auditable events*].**

FAU_GEN.1.2 **The TSF shall record within each audit record at least the following information:**

a) **Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and**

b) **For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: *other audit relevant information*].**

### FAU_GEN.2 User identity association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

FAU_GEN.2.1 **The TSF shall be able to associate each auditable event with the identity of the user that caused the event.**

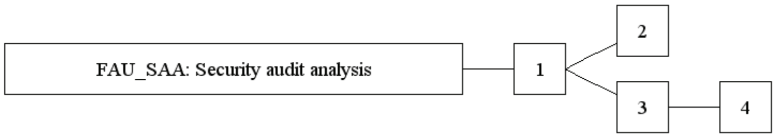
## 8.3 Security audit analysis (FAU_SAA)

### Family Behaviour

93 This family defines requirements for automated means that analyse system activity and audit data looking for possible or real security violations. This analysis may work in support of intrusion detection, or automatic response to an imminent security violation.

94 The actions to be taken based on the detection can be specified using the Security audit automatic response (FAU_ARP) family as desired.

### Component levelling




95 In FAU_SAA.1 Potential violation analysis, basic threshold detection on the basis of a fixed rule set is required.

96 In FAU_SAA.2 Profile based anomaly detection, the TSF maintains individual profiles of system usage, where a profile represents the historical patterns of usage performed by members of the profile target group. A profile target group refers to a group of one or more individuals (e.g. a single user, users who share a group ID or group account, users who operate under an assigned role, users of an entire system or network node) who interact with the TSF. Each member of a profile target group is assigned an individual suspicion rating that represents how well that member's current activity corresponds to the established patterns of usage represented in the profile. This analysis can be performed at runtime or during a post-collection batch-mode analysis.

97 In FAU_SAA.3 Simple attack heuristics, the TSF shall be able to detect the occurrence of signature events that represent a significant threat to TSP enforcement. This search for signature events may occur in real-time or during a post-collection batch-mode analysis.

98 In FAU_SAA.4 Complex attack heuristics, the TSF shall be able to represent and detect multi-step intrusion scenarios. The TSF is able to compare system events (possibly performed by multiple individuals) against event sequences known to represent entire intrusion scenarios. The TSF shall be able to indicate when a signature event or event sequence is found that indicates a potential violation of the TSP.

### Management: FAU_SAA.1

99 The following actions could be considered for the management functions in FMT:

a) maintenance of the rules by (adding, modifying, deletion) of rules from the set of rules.

### Management: FAU_SAA.2

100 The following actions could be considered for the management functions in FMT:

a) maintenance (deletion, modification, addition) of the group of users in the profile target group.

### Management: FAU_SAA.3

101 The following actions could be considered for the management functions in FMT:

a) maintenance (deletion, modification, addition) of the subset of system events.

### Management: FAU_SAA.4

102 The following actions could be considered for the management functions in FMT:

a) maintenance (deletion, modification, addition) of the subset of system events;

b) maintenance (deletion, modification, addition) of the set of sequence of system events.

### Audit: FAU_SAA.1, FAU_SAA.2, FAU_SAA.3, FAU_SAA.4

103 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Enabling and disabling of any of the analysis mechanisms;

b) Minimal: Automated responses performed by the tool.

## FAU_SAA.1 Potential violation analysis

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_SAA.1.1 **The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.**

FAU_SAA.1.2 **The TSF shall enforce the following rules for monitoring audited events:**

**a) Accumulation or combination of [assignment: *subset of defined auditable events*] known to indicate a potential security violation;**

**b) [assignment: *any other rules*].**

### FAU_SAA.2 Profile based anomaly detection

Hierarchical to: FAU_SAA.1 Potential violation analysis

Dependencies: FIA_UID.1 Timing of identification

FAU_SAA.2.1 The TSF shall be able to **maintain profiles** of **system usage, where an individual profile represents** the **historical patterns** of **usage performed by** the **member(s) of [assignment:** ***the profile target group*****].**

FAU_SAA.2.2 The TSF shall **be able to maintain a suspicion rating associated with each user whose activity is recorded in a profile, where** the **suspicion rating represents the degree to which the user's current activity is found inconsistent with the established patterns** of **usage represented in the profile.**

FAU_SAA.2.3 **The TSF shall be able to indicate an imminent violation of the TSP when a user's suspicion rating exceeds the following threshold conditions [assignment:** ***conditions under which anomalous activity is reported by the TSF*****].**

### FAU_SAA.3 Simple attack heuristics

Hierarchical to: FAU_SAA.1 Potential violation analysis

Dependencies: No dependencies.

FAU_SAA.3.1 The TSF shall be able to **maintain an internal representation** of the **following signature** events **[assignment:** ***a subset of system events*****] that may** indicate a violation of the TSP.

FAU_SAA.3.2 The TSF shall **be able to compare** the **signature** events **against the record** of **system activity discernible from an examination of** [assignment: ***the information*** *to* ***be used to determine system activity*****].**

FAU_SAA.3.3 **The TSF shall be able to indicate an imminent violation of the TSP when a system event is found to match a signature event that indicates a potential violation of the TSP.**

### FAU_SAA.4 Complex attack heuristics

Hierarchical to: FAU_SAA.3 Simple attack heuristics

Dependencies: No dependencies.

FAU_SAA.4.1 The TSF shall be able to maintain an internal representation of the following **event sequences of known intrusion scenarios [assignment:** ***list of sequences of system events whose occurrence are representative of known penetration scenarios*****] and the following** signature events [assignment: *a subset of system events*] that may indicate a **potential** violation of the TSP.

**FAU_SAA.4.2** The TSF shall be able to compare the signature events **and event sequences** against the record of system activity discernible from an examination of [assignment: *the information to be used to determine system activity*].

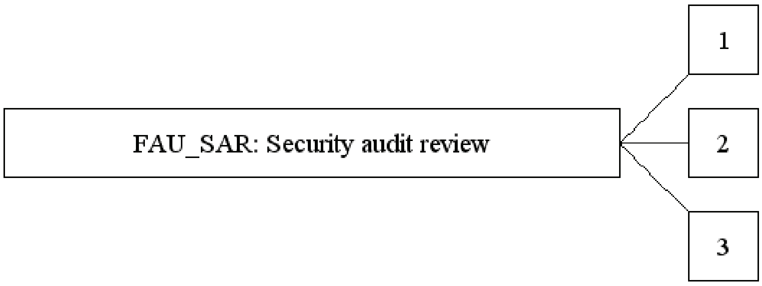
**FAU_SAA.4.3** The TSF shall be able to indicate an imminent violation of the TSP when system **activity** is found to match a signature event **or event sequence** that indicates a potential violation of the TSP.

## 8.4 Security audit review (FAU_SAR)

### Family Behaviour

104 This family defines the requirements for audit tools that should be available to authorised users to assist in the review of audit data.

### Component levelling




105 FAU_SAR.1 Audit review, provides the capability to read information from the audit records.

106 FAU_SAR.2 Restricted audit review, requires that there are no other users except those that have been identified in FAU_SAR.1 Audit review that can read the information.

107 FAU_SAR.3 Selectable audit review, requires audit review tools to select the audit data to be reviewed based on criteria.

### Management: FAU_SAR.1

108 The following actions could be considered for the management functions in FMT:

a) maintenance (deletion, modification, addition) of the group of users with read access right to the audit records.

### Management:FAU_SAR.2, FAU_SAR.3

109 There are no management activities foreseen.

### Audit: FAU_SAR.1

110 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Reading of information from the audit records.

### Audit: FAU_SAR.2

111 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Unsuccessful attempts to read information from the audit records.

Audit: FAU_SAR.3

112 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Detailed: the parameters used for the viewing.

### FAU_SAR.1 Audit review

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

113 This component will provide authorised users the capability to obtain and interpret the information. In case of human users this information needs to be in a human understandable presentation. In case of external IT entities the information needs to be unambiguously represented in an electronic fashion.

FAU_SAR.1.1 **The TSF shall provide [assignment: *authorised users*] with the capability to read [assignment: *list of audit information*] from the audit records.**

FAU_SAR.1.2 **The TSF shall provide the audit records in a manner suitable for the user to interpret the information.**

### FAU_SAR.2 Restricted audit review

Hierarchical to: No other components.

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.2.1 **The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.**

### FAU_SAR.3 Selectable audit review

Hierarchical to: No other components.

Dependencies: FAU_SAR.1 Audit review

FAU_SAR.3.1 **The TSF shall provide the ability to perform [selection: *searches,sorting,ordering*] of audit data based on [assignment: *criteria with logical relations*].**

## 8.5 Security audit event selection (FAU_SEL)

### Family Behaviour

114 This family defines requirements to select the events to be audited during TOE operation. It defines requirements to include or exclude events from the set of auditable events.

### Component levelling


115 FAU_SEL.1 Selective audit, requires the ability to include or exclude events from the set of audited events based upon attributes to be specified by the PP/ST author.

### Management: FAU_SEL.1

116 The following actions could be considered for the management functions in FMT:

a) maintenance of the rights to view/modify the audit events.

### Audit: FAU_SEL.1

117 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All modifications to the audit configuration that occur while the audit collection functions are operating.

## FAU_SEL.1 Selective audit

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FMT_MTD.1 Management of TSF data

FAU_SEL.1.1 **The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:**

**a) [selection: *object identity,user identity,subject identity,host identity,event type*]**

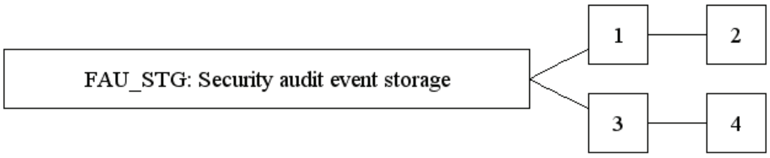
**b) [assignment: *list of additional attributes that audit selectivity is based upon*]**

## 8.6 Security audit event storage (FAU_STG)

### Family Behaviour

118 This family defines the requirements for the TSF to be able to create and maintain a secure audit trail.

### Component levelling




119 At FAU_STG.1 Protected audit trail storage, requirements are placed on the audit trail. It will be protected from unauthorised deletion and/or modification.

120 FAU_STG.2 Guarantees of audit data availability, specifies the guarantees that the TSF maintains over the audit data given the occurrence of an undesired condition.

121 FAU_STG.3 Action in case of possible audit data loss, specifies actions to be taken if a threshold on the audit trail is exceeded.

122 FAU_STG.4 Prevention of audit data loss, specifies actions in case the audit trail is full.

### Management:FAU_STG.1

123 There are no management activities foreseen.

### Management: FAU_STG.2

124 The following actions could be considered for the management functions in FMT:

a) maintenance of the parameters that control the audit storage capability.

### Management: FAU_STG.3

125 The following actions could be considered for the management functions in FMT:

a) maintenance of the threshold;

b) maintenance (deletion, modification, addition) of actions to be taken in case of imminent audit storage failure.

### Management: FAU_STG.4

126 The following actions could be considered for the management functions in FMT:

a) maintenance (deletion, modification, addition) of actions to be taken in case of audit storage failure.

### Audit: FAU_STG.1, FAU_STG.2

127 There are no auditable events foreseen.

### Audit: FAU_STG.3

128 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Actions taken due to exceeding of a threshold.

### Audit: FAU_STG.4

129 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Actions taken due to the audit storage failure.

## FAU_STG.1 Protected audit trail storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.1.1 **The TSF shall protect the stored audit records from unauthorised deletion.**

FAU_STG.1.2 **The TSF shall be able to [selection, choose one of: *prevent,detect*] unauthorised modifications to the stored audit records in the audit trail.**

## FAU_STG.2 Guarantees of audit data availability

Hierarchical to: FAU_STG.1 Protected audit trail storage

Dependencies: FAU_GEN.1 Audit data generation

FAU_STG.2.1 The TSF shall protect the stored audit records from unauthorised deletion.

FAU_STG.2.2 The TSF shall be able to [selection, choose one of: *prevent,detect*] unauthorised modifications to the stored audit records in the audit trail.

FAU_STG.2.3 **The TSF shall ensure that [assignment: *metric for saving audit records*] audit records will be maintained when the following conditions occur: [selection: *audit storage exhaustion,failure,attack*]**

## FAU_STG.3 Action in case of possible audit data loss

Hierarchical to: No other components.

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.3.1 **The TSF shall [assignment: *actions to be taken in case of possible audit storage failure*] if the audit trail exceeds [assignment: *pre-defined limit*].**

## FAU_STG.4 Prevention of audit data loss

Hierarchical to: FAU_STG.3 Action in case of possible audit data loss

Dependencies: FAU_STG.1 Protected audit trail storage

FAU_STG.4.1 The TSF shall **[selection, choose one of: *"ignore auditable events","prevent auditable events, except those taken by the authorised user with special rights", "overwrite the oldest stored audit records"*] and** [assignment: ***other*** *actions to be taken in case of audit storage failure*] if the audit trail **is full.**

# 9 Class FCO: Communication

130 This class provides two families specifically concerned with assuring the identity of a party participating in a data exchange. These families are related to assuring the identity of the originator of transmitted information (proof of origin) and assuring the identity of the recipient of transmitted information (proof of receipt). These families ensure that an originator cannot deny having sent the message, nor can the recipient deny having received it.

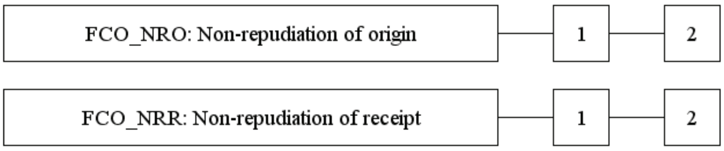


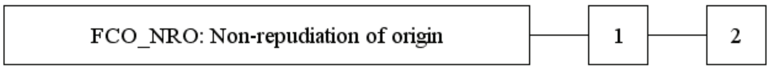

**Figure 10 - FCO: Communication class decomposition**

## 9.1 Non-repudiation of origin (FCO_NRO)

### Family Behaviour

131 Non-repudiation of origin ensures that the originator of information cannot successfully deny having sent the information. This family requires that the TSF provide a method to ensure that a subject that receives information during a data exchange is provided with evidence of the origin of the information. This evidence can then be verified by either this subject or other subjects.

### Component levelling




132 FCO_NRO.1 Selective proof of origin, requires the TSF to provide subjects with the capability to request evidence of the origin of information.

133 FCO_NRO.2 Enforced proof of origin, requires that the TSF always generate evidence of origin for transmitted information.

### Management: FCO_NRO.1, FCO_NRO.2

134 The following actions could be considered for the management functions in FMT:

a) The management of changes to information types, fields, originator attributes and recipients of evidence.

### Audit: FCO_NRO.1

135 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The identity of the user who requested that evidence of origin would be generated.

b) Minimal: The invocation of the non-repudiation service.

c) Basic: Identification of the information, the destination, and a copy of the evidence provided.

d) Detailed: The identity of the user who requested a verification of the evidence.

### Audit: FCO_NRO.2

136 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The invocation of the non-repudiation service.

b) Basic: Identification of the information, the destination, and a copy of the evidence provided.

c) Detailed: The identity of the user who requested a verification of the evidence.

## FCO_NRO.1 Selective proof of origin

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FCO_NRO.1.1 **The TSF shall be able to generate evidence of origin for transmitted [assignment: *list of information types*] at the request of the [selection: *originator, recipient,[assignment: list of third parties]*].**

FCO_NRO.1.2 **The TSF shall be able to relate the [assignment: *list of attributes*] of the originator of the information, and the [assignment: *list of information fields*] of the information to which the evidence applies.**

FCO_NRO.1.3 **The TSF shall provide a capability to verify the evidence of origin of information to [selection: *originator, recipient,[assignment: list of third parties]*] given [assignment: *limitations on the evidence of origin*].**

## FCO_NRO.2 Enforced proof of origin

Hierarchical to: FCO_NRO.1 Selective proof of origin

Dependencies: FIA_UID.1 Timing of identification

FCO_NRO.2.1 The TSF shall **enforce the generation of** evidence of origin for transmitted [assignment: *list of information types*] at **all times.**

FCO_NRO.2.2 The TSF shall be able to relate the [assignment: *list of attributes*] of the originator of the information, and the [assignment: ***list*** *of information fields*] of the information to which the evidence applies.

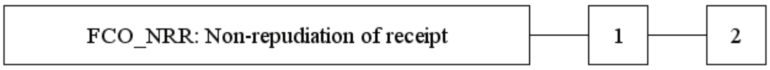
FCO_NRO.2.3 The TSF shall provide a capability to verify the evidence of origin of information to [selection: *originator, recipient,[assignment: list of third parties]*] given [assignment: *limitations on the evidence of origin*].

## 9.2 Non-repudiation of receipt (FCO_NRR)

### Family Behaviour

137 Non-repudiation of receipt ensures that the recipient of information cannot successfully deny receiving the information. This family requires that the TSF provide a method to ensure that a subject that transmits information during a data exchange is provided with evidence of receipt of the information. This evidence can then be verified by either this subject or other subjects.

### Component levelling




138 FCO_NRR.1 Selective proof of receipt, requires the TSF to provide subjects with a capability to request evidence of the receipt of information.

139 FCO_NRR.2 Enforced proof of receipt, requires that the TSF always generate evidence of receipt for received information.

### Management: FCO_NRR.1, FCO_NRR.2

140 The following actions could be considered for the management functions in FMT:

a) The management of changes to information types, fields, originator attributes and third parties recipients of evidence.

### Audit: FCO_NRR.1

141 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The identity of the user who requested that evidence of receipt would be generated.

b) Minimal: The invocation of the non-repudiation service.

c) Basic: Identification of the information, the destination, and a copy of the evidence provided.

d) Detailed: The identity of the user who requested a verification of the evidence.

### Audit: FCO_NRR.2

142 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The invocation of the non-repudiation service.

b) Basic: Identification of the information, the destination, and a copy of the evidence provided.

c) Detailed: The identity of the user who requested a verification of the evidence.

### FCO_NRR.1 Selective proof of receipt

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FCO_NRR.1.1 **The TSF shall be able to generate evidence of receipt for received [assignment: *list of information types*] at the request of the [selection: *originator, recipient,[assignment: list of third parties]*].**

FCO_NRR.1.2 **The TSF shall be able to relate the [assignment: *list of attributes*] of the recipient of the information, and the [assignment: *list of information fields*] of the information to which the evidence applies.**

FCO_NRR.1.3 **The TSF shall provide a capability to verify the evidence of receipt of information to [selection: *originator, recipient,[assignment: list of third parties]*] given [assignment: *limitations on the evidence of receipt*].**

### FCO_NRR.2 Enforced proof of receipt

Hierarchical to: FCO_NRR.1 Selective proof of receipt

Dependencies: FIA_UID.1 Timing of identification

FCO_NRR.2.1 The TSF shall **enforce the generation of** evidence of receipt for received [assignment: *list of information types*].

FCO_NRR.2.2 The TSF shall be able to relate the [assignment: *list of attributes*] of the recipient of the information, and the [assignment: *list of information fields*] of the information to which the evidence applies.

FCO_NRR.2.3 The TSF shall provide a capability to verify the evidence of receipt of information to [selection: *originator, recipient,[assignment: list of third parties]*] given [assignment: *limitations on the evidence of receipt*].

# 10 Class FCS: Cryptographic support

143 The TSF may employ cryptographic functionality to help satisfy several high-level security objectives. These include (but are not limited to): identification and authentication, non-repudiation, trusted path, trusted channel and data separation. This class is used when the TOE implements cryptographic functions, the implementation of which could be in hardware, firmware and/or software.

144 The FCS: Cryptographic support class is composed of two families: Cryptographic key management (FCS_CKM) and Cryptographic operation (FCS_COP). The Cryptographic key management (FCS_CKM) family addresses the management aspects of cryptographic keys, while the Cryptographic operation (FCS_COP) family is concerned with the operational use of those cryptographic keys.

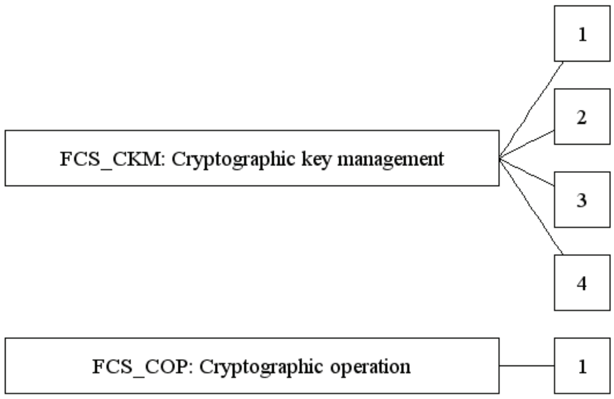


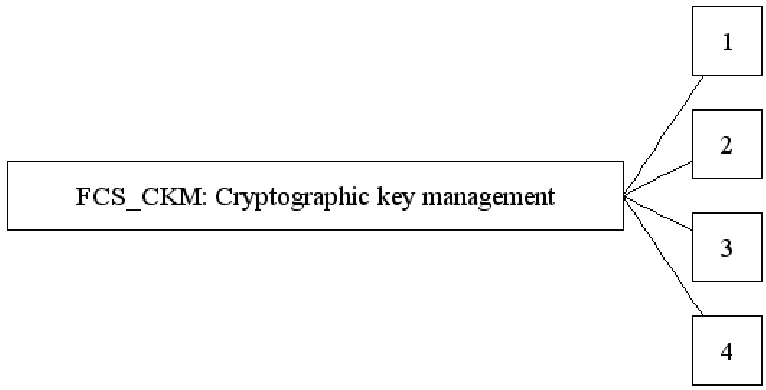

**Figure 11 - FCS: Cryptographic support class decomposition**

## 10.1 Cryptographic key management (FCS_CKM)

### Family Behaviour

145 Cryptographic keys must be managed throughout their life cycle. This family is intended to support that lifecycle and consequently defines requirements for the following activities: cryptographic key generation, cryptographic key distribution, cryptographic key access and cryptographic key destruction. This family should be included whenever there are functional requirements for the management of cryptographic keys.

### Component levelling




146 FCS_CKM.1 Cryptographic key generation, requires cryptographic keys to be generated in accordance with a specified algorithm and key sizes which can be based on an assigned standard.

147 FCS_CKM.2 Cryptographic key distribution, requires cryptographic keys to be distributed in accordance with a specified distribution method which can be based on an assigned standard.

148 FCS_CKM.3 Cryptographic key access, requires access to cryptographic keys to be performed in accordance with a specified access method which can be based on an assigned standard.

149 FCS_CKM.4 Cryptographic key destruction, requires cryptographic keys to be destroyed in accordance with a specified destruction method which can be based on an assigned standard.

### Management: FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4

150 The following actions could be considered for the management functions in FMT:

a) the management of changes to cryptographic key attributes. Examples of key attributes include user, key type (e.g. public, private, secret), validity period, and use (e.g. digital signature, key encryption, key agreement, data encryption).

Audit: FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.4

151 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Success and failure of the activity.

b) Basic: The object attribute(s), and object value(s) excluding any sensitive information (e.g. secret or private keys).

### FCS_CKM.1 Cryptographic key generation

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

FCS_CKM.1.1 **The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: *cryptographic key generation algorithm*] and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].**

### FCS_CKM.2 Cryptographic key distribution

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

FCS_CKM.2.1 **The TSF shall distribute cryptographic keys in accordance with a specified cryptographic key distribution method [assignment: *cryptographic key distribution method*] that meets the following: [assignment: *list of standards*].**

### FCS_CKM.3 Cryptographic key access

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FMT_MSA.2 Secure security attributes

FCS_CKM.3.1 **The TSF shall perform [assignment: *type of cryptographic key access*] in accordance with a specified cryptographic key access method [assignment: *cryptographic key access method*] that meets the following: [assignment: *list of standards*].**

## FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FMT_MSA.2 Secure security attributes

FCS_CKM.4.1 **The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].**

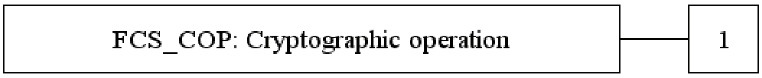
## 10.2 Cryptographic operation (FCS_COP)

### Family Behaviour

152 In order for a cryptographic operation to function correctly, the operation must be performed in accordance with a specified algorithm and with a cryptographic key of a specified size. This family should be included whenever there are requirements for cryptographic operations to be performed.

153 Typical cryptographic operations include data encryption and/or decryption, digital signature generation and/or verification, cryptographic checksum generation for integrity and/or verification of checksum, secure hash (message digest), cryptographic key encryption and/or decryption, and cryptographic key agreement.

### Component levelling




154 FCS_COP.1 Cryptographic operation, requires a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes. The specified algorithm and cryptographic key sizes can be based on an assigned standard.

### Management:FCS_COP.1

155 There are no management activities foreseen.

### Audit: FCS_COP.1

156 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Success and failure, and the type of cryptographic operation.

b) Basic: Any applicable cryptographic mode(s) of operation, subject attributes and object attributes.

### FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction
FMT_MSA.2 Secure security attributes

FCS_COP.1.1 **The TSF shall perform [assignment: *list of cryptographic operations*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].**

# 11 Class FDP: User data protection

157 This class contains families specifying requirements for TOE security functions and TOE security function policies related to protecting user data. FDP: User data protection is split into four groups of families (listed below) that address user data within a TOE, during import, export, and storage as well as security attributes directly related to user data.

158 The families in this class are organised into four groups:

a) User data protection security function policies:

- Access control policy (FDP_ACC); and
- Information flow control policy (FDP_IFC).

Components in these families permit the PP/ST author to name the user data protection security function policies and define the scope of control of the policy, necessary to address the security objectives. The names of these policies are meant to be used throughout the remainder of the functional components that have an operation that calls for an assignment or selection of an "access control SFP" or an "information flow control SFP". The rules that define the functionality of the named access control and information flow control SFPs will be defined in the Access control functions (FDP_ACF) and Information flow control functions (FDP_IFF) families (respectively).

b) Forms of user data protection:

- Access control functions (FDP_ACF);
- Information flow control functions (FDP_IFF);
- Internal TOE transfer (FDP_ITT);
- Residual information protection (FDP_RIP);
- Rollback (FDP_ROL); and
- Stored data integrity (FDP_SDI).

c) Off-line storage, import and export:

- Data authentication (FDP_DAU);
- Export to outside TSF control (FDP_ETC);
- Import from outside TSF control (FDP_ITC).

Components in these families address the trustworthy transfer into or out of the TSC.

d) Inter-TSF communication:

- Inter-TSF user data confidentiality transfer protection (FDP_UCT); and
- Inter-TSF user data integrity transfer protection (FDP_UIT).

Components in these families address communication between the TSF of the TOE and another trusted IT product.

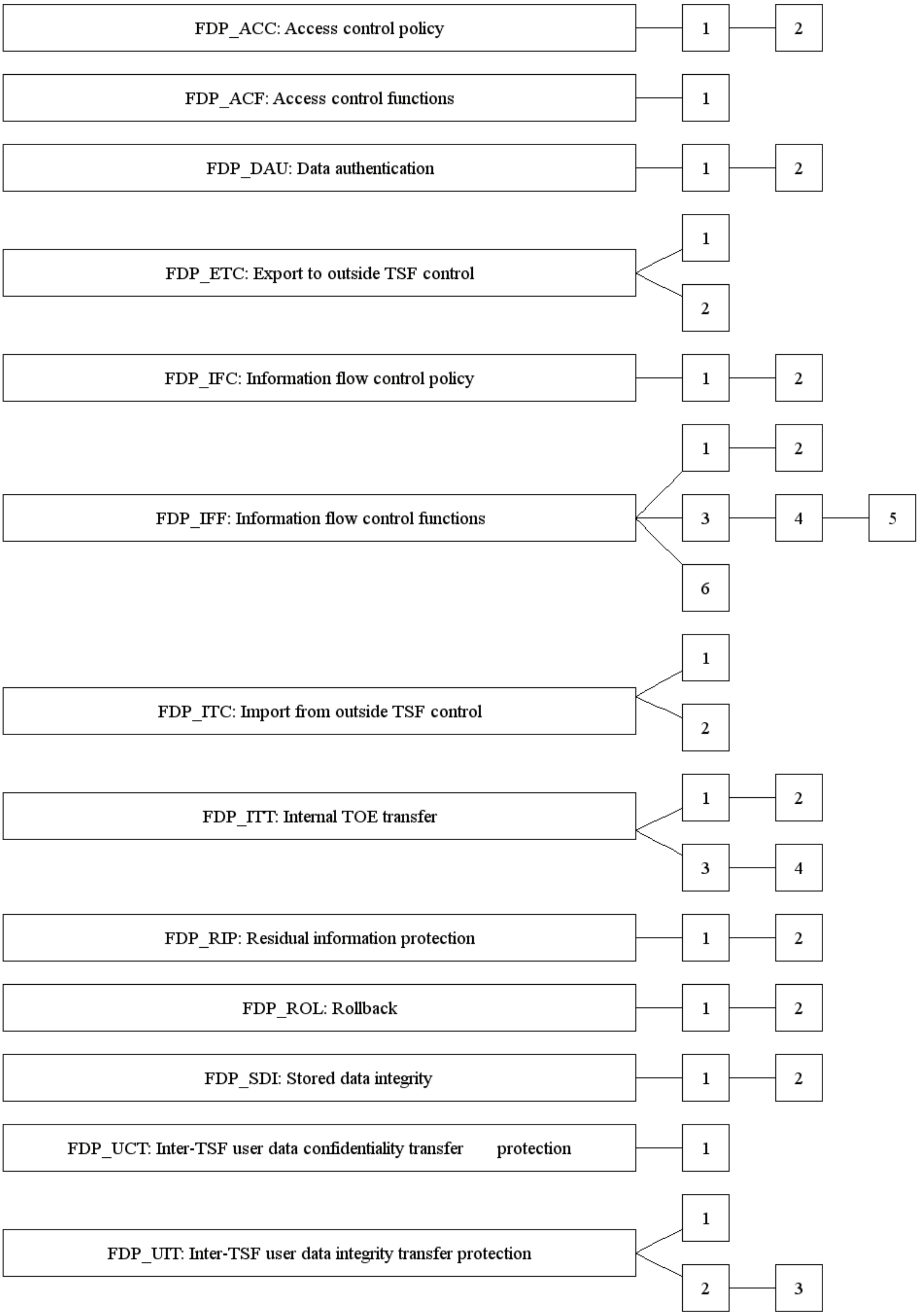


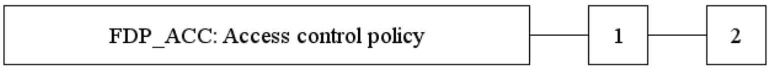

**Figure 12 - FDP: User data protection class decomposition**

## 11.1 Access control policy (FDP_ACC)

### Family Behaviour

159 This family identifies the access control SFPs (by name) and defines the scope of control of the policies that form the identified access control portion of the TSP. This scope of control is characterised by three sets: the subjects under control of the policy, the objects under control of the policy, and the operations among controlled subjects and controlled objects that are covered by the policy. The criteria allows multiple policies to exist, each having a unique name. This is accomplished by iterating components from this family once for each named access control policy. The rules that define the functionality of an access control SFP will be defined by other families such as Access control functions (FDP_ACF) and Stored data integrity (FDP_SDI). The names of the access control SFPs identified here in Access control policy (FDP_ACC) are meant to be used throughout the remainder of the functional components that have an operation that calls for an assignment or selection of an "access control SFP."

### Component levelling




160 FDP_ACC.1 Subset access control, requires that each identified access control SFP be in place for a subset of the possible operations on a subset of the objects in the TOE.

161 FDP_ACC.2 Complete access control, requires that each identified access control SFP cover all operations on subjects and objects covered by that SFP. It further requires that all objects and operations with the TSC are covered by at least one identified access control SFP.

### Management:FDP_ACC.1, FDP_ACC.2

162 There are no management activities foreseen.

### Audit: FDP_ACC.1, FDP_ACC.2

163 There are no auditable events foreseen.

## FDP_ACC.1 Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 **The TSF shall enforce the [assignment: *access control SFP*] on [assignment: *list of subjects, objects, and operations among subjects and objects covered by the SFP*].**

## FDP_ACC.2 Complete access control

Hierarchical to: FDP_ACC.1 Subset access control

Dependencies: FDP_ACF.1 Security attribute based access control

**FDP_ACC.2.1** The TSF shall enforce the [assignment: *access control SFP*] on [assignment: *list of subjects and* ***objects***] **and all** operations among subjects and objects covered by the **SFP.**

**FDP_ACC.2.2** **The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.**

## 11.2 Access control functions (FDP_ACF)

### Family Behaviour

164 This family describes the rules for the specific functions that can implement an access control policy named in Access control policy (FDP_ACC). Access control policy (FDP_ACC) specifies the scope of control of the policy.

### Component levelling


165 This family addresses security attribute usage and characteristics of policies. The component within this family is meant to be used to describe the rules for the function that implements the SFP as identified in Access control policy (FDP_ACC). The PP/ST author may also iterate this component to address multiple policies in the TOE.

166 FDP_ACF.1 Security attribute based access control Security attribute based access control allows the TSF to enforce access based upon security attributes and named groups of attributes. Furthermore, the TSF may have the ability to explicitly authorise or deny access to an object based upon security attributes.

### Management: FDP_ACF.1

167 The following actions could be considered for the management functions in FMT:

a) Managing the attributes used to make explicit access or denial based decisions.

### Audit: FDP_ACF.1

168 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Successful requests to perform an operation on an object covered by the SFP.

b) Basic: All requests to perform an operation on an object covered by the SFP.

c) Detailed: The specific security attributes used in making an access check.

### FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1 **The TSF shall enforce the [assignment: *access control SFP*] to objects based on the following: [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*].**

FDP_ACF.1.2 **The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*].**

FDP_ACF.1.3 **The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*].**

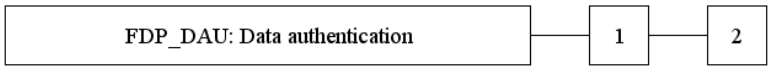
FDP_ACF.1.4 **The TSF shall explicitly deny access of subjects to objects based on the [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].**

## 11.3 Data authentication (FDP_DAU)

### Family Behaviour

169 Data authentication permits an entity to accept responsibility for the authenticity of information (e.g., by digitally signing it). This family provides a method of providing a guarantee of the validity of a specific unit of data that can be subsequently used to verify that the information content has not been forged or fraudulently modified. In contrast to FAU: Security audit, this family is intended to be applied to "static" data rather than data that is being transferred.

### Component levelling




170 FDP_DAU.1 Basic Data Authentication, requires that the TSF is capable of generating a guarantee of authenticity of the information content of objects (e.g. documents).

171 FDP_DAU.2 Data Authentication with Identity of Guarantor additionally requires that the TSF is capable of establishing the identity of the subject who provided the guarantee of authenticity.

### Management: FDP_DAU.1, FDP_DAU.2

172 The following actions could be considered for the management functions in FMT:

a) The assignment or modification of the objects for which data authentication may apply could be configurable in the system.

### Audit: FDP_DAU.1

173 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Successful generation of validity evidence.

b) Basic: Unsuccessful generation of validity evidence.

c) Detailed: The identity of the subject that requested the evidence.

### Audit: FDP_DAU.2

174 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Successful generation of validity evidence.

b) Basic: Unsuccessful generation of validity evidence.

c) Detailed: The identity of the subject that requested the evidence.

d) Detailed: The identity of the subject that generated the evidence.

## FDP_DAU.1 Basic Data Authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_DAU.1.1 **The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of [assignment: *list of objects or information types*].**

FDP_DAU.1.2 **The TSF shall provide [assignment: *list of subjects*] with the ability to verify evidence of the validity of the indicated information.**

## FDP_DAU.2 Data Authentication with Identity of Guarantor

Hierarchical to: FDP_DAU.1 Basic Data Authentication

Dependencies: FIA_UID.1 Timing of identification

FDP_DAU.2.1 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of [assignment: *list of objects or information types*].

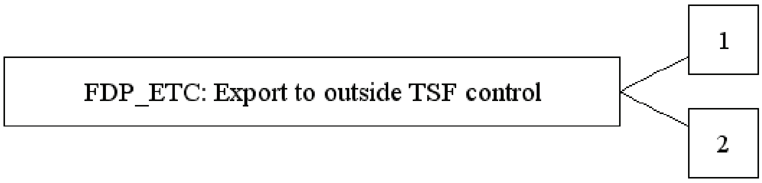
FDP_DAU.2.2 The TSF shall provide [assignment: *list of subjects*] with the ability to verify evidence of the validity of the indicated information **and the identity of the user that generated the evidence.**

## 11.4 Export to outside TSF control (FDP_ETC)

### Family Behaviour

175 This family defines functions for exporting user data from the TOE such that its security attributes and protection either can be explicitly preserved or can be ignored once it has been exported. It is concerned with limitations on export and with the association of security attributes with the exported user data.

### Component levelling




176 FDP_ETC.1 Export of user data without security attributes, requires that the TSF enforce the appropriate SFPs when exporting user data outside the TSF. User data that is exported by this function is exported without its associated security attributes.

177 FDP_ETC.2 Export of user data with security attributes, requires that the TSF enforce the appropriate SFPs using a function that accurately and unambiguously associates security attributes with the user data that is exported.

### Management:FDP_ETC.1

178 There are no management activities foreseen.

### Management: FDP_ETC.2

179 The following actions could be considered for the management functions in FMT:

a) The additional exportation control rules could be configurable by a user in a defined role.

### Audit: FDP_ETC.1, FDP_ETC.2

180 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Successful export of information.

b) Basic: All attempts to export information.

## FDP_ETC.1 Export of user data without security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ETC.1.1 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] when exporting user data, controlled under the SFP(s), outside of the TSC.**

FDP_ETC.1.2 **The TSF shall export the user data without the user data's associated security attributes**

## FDP_ETC.2 Export of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ETC.2.1 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] when exporting user data, controlled under the SFP(s), outside of the TSC.**

FDP_ETC.2.2 **The TSF shall export the user data with the user data's associated security attributes.**

FDP_ETC.2.3 **The TSF shall ensure that the security attributes, when exported outside the TSC, are unambiguously associated with the exported user data.**

FDP_ETC.2.4 **The TSF shall enforce the following rules when user data is exported from the TSC: [assignment: *additional exportation control rules*].**

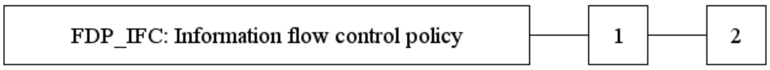
## 11.5 Information flow control policy (FDP_IFC)

### Family Behaviour

181 This family identifies the information flow control SFPs (by name) and defines the scope of control of the policies that form the identified information flow control portion of the TSP. This scope of control is characterised by three sets: the subjects under control of the policy, the information under control of the policy, and operations which cause controlled information to flow to and from controlled subjects covered by the policy. The criteria allows multiple policies to exist, each having a unique name. This is accomplished by iterating components from this family once for each named information flow control policy. The rules that define the functionality of an information flow control SFP will be defined by other families such as Information flow control functions (FDP_IFF) and Stored data integrity (FDP_SDI). The names of the information flow control SFPs identified here in Information flow control policy (FDP_IFC) are meant to be used throughout the remainder of the functional components that have an operation that calls for an assignment or selection of an “information flow control SFP.”

182 The TSF mechanism controls the flow of information in accordance with the information flow control SFP. Operations that would change the security attributes of information are not generally permitted as this would be in violation of an information flow control SFP. However, such operations may be permitted as exceptions to the information flow control SFP if explicitly specified.

### Component levelling




183 FDP_IFC.1 Subset information flow control, requires that each identified information flow control SFPs be in place for a subset of the possible operations on a subset of information flows in the TOE.

184 FDP_IFC.2 Complete information flow control, requires that each identified information flow control SFP cover all operations on subjects and information covered by that SFP. It further requires that all information flows and operations with the TSC are covered by at least one identified information flow control SFP. In conjunction with the FPT_RVM.1 component, this gives the “always invoked” aspect of a reference monitor.

### Management:FDP_IFC.1, FDP_IFC.2

185 There are no management activities foreseen.

### Audit: FDP_IFC.1, FDP_IFC.2

186 There are no auditable events foreseen.

### FDP_IFC.1 Subset information flow control

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 **The TSF shall enforce the [assignment: *information flow control SFP*] on [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*].**

### FDP_IFC.2 Complete information flow control

Hierarchical to: FDP_IFC.1 Subset information flow control

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.2.1 The TSF shall enforce the [assignment: *information flow control SFP*] on [assignment: *list of subjects and* ***information*****] and all** operations that cause **that** information to flow to and from subjects covered by the **SFP.**

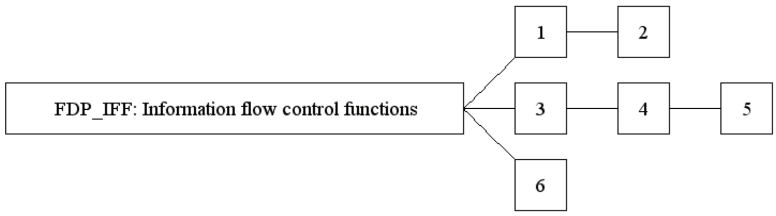
FDP_IFC.2.2 **The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.**

## 11.6 Information flow control functions (FDP_IFF)

### Family Behaviour

187 This family describes the rules for the specific functions that can implement the information flow control SFPs named in Information flow control policy (FDP_IFC), which also specifies the scope of control of the policy. It consists of two kinds of requirements: one addressing the common information flow function issues, and a second addressing illicit information flows (i.e. covert channels). This division arises because the issues concerning illicit information flows are, in some sense, orthogonal to the rest of an information flow control SFP. By their nature they circumvent the information flow control SFP resulting in a violation of the policy. As such, they require special functions to either limit or prevent their occurrence.

### Component levelling




188 FDP_IFF.1 Simple security attributes, requires security attributes on information, and on subjects that cause that information to flow and on subjects that act as recipients of that information. It specifies the rules that must be enforced by the function, and describes how security attributes are derived by the function.

189 FDP_IFF.2 Hierarchical security attributes expands on the requirements of FDP_IFF.1 Simple security attributes by requiring that all information flow control SFPs in the TSP use hierarchical security attributes that form a lattice.

190 FDP_IFF.3 Limited illicit information flows, requires the SFP to cover illicit information flows, but not necessarily eliminate them.

191 FDP_IFF.4 Partial elimination of illicit information flows, requires the SFP to cover the elimination of some (but not necessarily all) illicit information flows.

192 FDP_IFF.5 No illicit information flows, requires SFP to cover the elimination of all illicit information flows.

193 FDP_IFF.6 Illicit information flow monitoring, requires the SFP to monitor illicit information flows for specified and maximum capacities.

### Management: FDP_IFF.1, FDP_IFF.2

194 The following actions could be considered for the management functions in FMT:

a) Managing the attributes used to make explicit access based decisions.

### Management:FDP_IFF.3, FDP_IFF.4, FDP_IFF.5

195 There are no management activities foreseen.

### Management: FDP_IFF.6

196 The following actions could be considered for the management functions in FMT:

a) The enabling or disabling of the monitoring function.

b) Modification of the maximum capacity at which the monitoring occurs.

### Audit: FDP_IFF.1, FDP_IFF.2, FDP_IFF.5

197 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Decisions to permit requested information flows.

b) Basic: All decisions on requests for information flow.

c) Detailed: The specific security attributes used in making an information flow enforcement decision.

d) Detailed: Some specific subsets of the information that has flowed based upon policy goals (e.g. auditing of downgraded material).

### Audit: FDP_IFF.3, FDP_IFF.4, FDP_IFF.6

198 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Decisions to permit requested information flows.

b) Basic: All decisions on requests for information flow.

c) Basic: The use of identified illicit information flow channels.

d) Detailed: The specific security attributes used in making an information flow enforcement decision.

e) Detailed: Some specific subsets of the information that has flowed based upon policy goals (e.g. auditing of downgraded material).

f) Detailed: The use of identified illicit information flow channels with estimated maximum capacity exceeding a specified value.

### FDP_IFF.1 Simple security attributes

Hierarchical to: No other components.

Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

FDP_IFF.1.1 **The TSF shall enforce the [assignment: *information flow control SFP*] based on the following types of subject and information security attributes: [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*].**

FDP_IFF.1.2 **The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*].**

FDP_IFF.1.3 **The TSF shall enforce the [assignment: *additional information flow control SFP rules*].**

FDP_IFF.1.4 **The TSF shall provide the following [assignment: *list of additional SFP capabilities*].**

FDP_IFF.1.5 **The TSF shall explicitly authorise an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorise information flows*].**

FDP_IFF.1.6 **The TSF shall explicitly deny an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly deny information flows*].**

### FDP_IFF.2 Hierarchical security attributes

Hierarchical to: FDP_IFF.1 Simple security attributes

Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialisation

FDP_IFF.2.1 The TSF shall enforce the [assignment: *information flow control SFP*] based on the following types of subject and information security attributes: [assignment: *list of subjects and information controlled under the indicated SFP, and for each, the security attributes*].

FDP_IFF.2.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules, **based on the ordering relationships between security attributes** hold: [assignment: *for each operation, the security attribute-based relationship that must hold between subject and information security attributes*].

FDP_IFF.2.3 The TSF shall enforce the [assignment: *additional information flow control SFP rules*].

**FDP_IFF.2.4** The TSF shall provide the following [assignment: *list of additional SFP capabilities*].

**FDP_IFF.2.5** The TSF shall explicitly authorise an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorise information flows*].

**FDP_IFF.2.6** The TSF shall explicitly deny an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly deny information flows*].

**FDP_IFF.2.7** **The TSF shall enforce the following relationships for any two valid information flow control security attributes:**

**a) There exists an ordering function that, given two valid security attributes, determines if the security attributes are equal, if one security attribute is greater than the other, or if the security attributes are incomparable; and**

**b) There exists a "least upper bound" in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is greater than or equal to the two valid security attributes; and**

**c) There exists a "greatest lower bound" in the set of security attributes, such that, given any two valid security attributes, there is a valid security attribute that is not greater than the two valid security attributes.**

## FDP_IFF.3 Limited illicit information flows

Hierarchical to: No other components.

Dependencies: AVA_CCA.1 Covert channel analysis
FDP_IFC.1 Subset information flow control

**FDP_IFF.3.1** **The TSF shall enforce the [assignment: *information flow control SFP*] to limit the capacity of [assignment: *types of illicit information flows*] to a [assignment: *maximum capacity*].**

## FDP_IFF.4 Partial elimination of illicit information flows

Hierarchical to: FDP_IFF.3 Limited illicit information flows

Dependencies: AVA_CCA.1 Covert channel analysis
FDP_IFC.1 Subset information flow control

**FDP_IFF.4.1** The TSF shall enforce the [assignment: *information flow control SFP*] to limit the capacity of [assignment: *types of illicit information flows*] to a [assignment: *maximum capacity*].

**FDP_IFF.4.2** **The TSF shall prevent [assignment: *types of illicit information flows*].**

### FDP_IFF.5 No illicit information flows

Hierarchical to: FDP_IFF.4 Partial elimination of illicit information flows

Dependencies: AVA_CCA.3 Exhaustive covert channel analysis
FDP_IFC.1 Subset information flow control

FDP_IFF.5.1 The TSF shall **ensure that no illicit information flows exist to circumvent** [assignment: ***name of*** *information flow control SFP*].

### FDP_IFF.6 Illicit information flow monitoring

Hierarchical to: No other components.

Dependencies: AVA_CCA.1 Covert channel analysis
FDP_IFC.1 Subset information flow control

FDP_IFF.6.1 **The TSF shall enforce the [assignment: *information flow control SFP*] to monitor [assignment: *types of illicit information flows*] when it exceeds the [assignment: *maximum capacity*].**

## 11.7 Import from outside TSF control (FDP_ITC)

### Family Behaviour

199 This family defines the mechanisms for introduction of user data into the TOE such that it has appropriate security attributes and is appropriately protected. It is concerned with limitations on importation, determination of desired security attributes, and interpretation of security attributes associated with the user data.

### Component levelling


200 FDP_ITC.1 Import of user data without security attributes, requires that the security attributes correctly represent the user data and are supplied separately from the object.

201 FDP_ITC.2 Import of user data with security attributes, requires that security attributes correctly represent the user data and are accurately and unambiguously associated with the user data imported from outside the TSC.

### Management: FDP_ITC.1, FDP_ITC.2

202 The following actions could be considered for the management functions in FMT:

a) The modification of the additional control rules used for import.

### Audit: FDP_ITC.1, FDP_ITC.2

203 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Successful import of user data, including any security attributes.

b) Basic: All attempts to import user data, including any security attributes.

c) Detailed: The specification of security attributes for imported user data supplied by an authorised user.

## FDP_ITC.1 Import of user data without security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialisation

FDP_ITC.1.1 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] when importing user data, controlled under the SFP, from outside of the TSC.**

FDP_ITC.1.2 **The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.**

FDP_ITC.1.3 **The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [assignment: *additional importation control rules*].**

## FDP_ITC.2 Import of user data with security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
FPT_TDC.1 Inter-TSF basic TSF data consistency

FDP_ITC.2.1 **The TSF shall enforce the [assignment: *access control SFP and/or information flow control SFP*] when importing user data, controlled under the SFP, from outside of the TSC.**

FDP_ITC.2.2 **The TSF shall use the security attributes associated with the imported user data.**

FDP_ITC.2.3 **The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.**

FDP_ITC.2.4 **The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.**

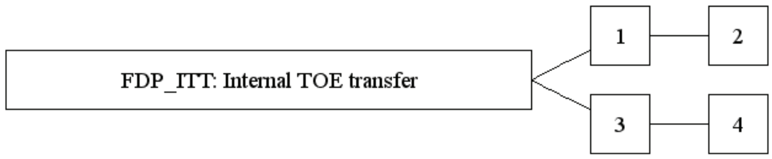
FDP_ITC.2.5 **The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: [assignment: *additional importation control rules*].**

## 11.8 Internal TOE transfer (FDP_ITT)

### Family Behaviour

204 This family provides requirements that address protection of user data when it is transferred between parts of a TOE across an internal channel. This may be contrasted with the Inter-TSF user data confidentiality transfer protection (FDP_UCT) and Inter-TSF user data integrity transfer protection (FDP_UIT) families, which provide protection for user data when it is transferred between distinct TSFs across an external channel, and Export to outside TSF control (FDP_ETC) and Import from outside TSF control (FDP_ITC), which address transfer of data to or from outside the TSF's control.

### Component levelling




205 FDP_ITT.1 Basic internal transfer protection, requires that user data be protected when transmitted between parts of the TOE.

206 FDP_ITT.2 Transmission separation by attribute, requires separation of data based on the value of SFP-relevant attributes in addition to the first component.

207 FDP_ITT.3 Integrity monitoring, requires that the SF monitor user data transmitted between parts of the TOE for identified integrity errors.

208 FDP_ITT.4 Attribute-based integrity monitoring expands on the third component by allowing the form of integrity monitoring to differ by SFP-relevant attribute.

### Management: FDP_ITT.1, FDP_ITT.2

209 The following actions could be considered for the management functions in FMT:

a) If the TSF provides multiple methods to protect user data during transmission between physically separated parts of the TOE, the TSF could provide a pre-defined role with the ability to select the method that will be used.

### Management: FDP_ITT.3, FDP_ITT.4

210 The following actions could be considered for the management functions in FMT:

a) The specification of the actions to be taken upon detection of an integrity error could be configurable.

### Audit: FDP_ITT.1, FDP_ITT.2

211 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Successful transfers of user data, including identification of the protection method used.

b) Basic: All attempts to transfer user data, including the protection method used and any errors that occurred.

### Audit: FDP_ITT.3, FDP_ITT.4

212 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Successful transfers of user data, including identification of the integrity protection method used.

b) Basic: All attempts to transfer user data, including the integrity protection method used and any errors that occurred.

c) Basic: Unauthorised attempts to change the integrity protection method.

d) Detailed: The action taken upon detection of an integrity error.

## FDP_ITT.1 Basic internal transfer protection

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ITT.1.1 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to prevent the [selection: *disclosure, modification, loss of use*] of user data when it is transmitted between physically-separated parts of the TOE.**

## FDP_ITT.2 Transmission separation by attribute

Hierarchical to: FDP_ITT.1 Basic internal transfer protection

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ITT.2.1 The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to prevent the [selection: *disclosure, modification, loss of use*] of user data when it is transmitted between physically-separated parts of the TOE.

FDP_ITT.2.2 **The TSF shall separate data controlled by the SFP(s) when transmitted between physically-separated parts of the TOE, based on the values of the following: [assignment: *security attributes that require separation*].**

## FDP_ITT.3 Integrity monitoring

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FDP_ITT.1 Basic internal transfer protection

FDP_ITT.3.1 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to monitor user data transmitted between physically-separated parts of the TOE for the following errors: [assignment: *integrity errors*].**

FDP_ITT.3.2 **Upon detection of a data integrity error, the TSF shall [assignment: *specify the action to be taken upon integrity error*].**

## FDP_ITT.4 Attribute-based integrity monitoring

Hierarchical to: FDP_ITT.3 Integrity monitoring

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FDP_ITT.2 Transmission separation by attribute

FDP_ITT.4.1 The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to monitor user data transmitted between physically-separated parts of the TOE for the following errors: [assignment: *integrity errors*], **based on the following attributes: [assignment: *security attributes that require separate transmission channels*].**

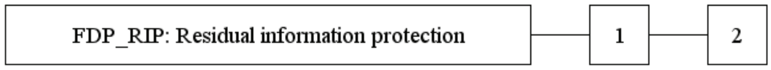
FDP_ITT.4.2 Upon detection of a data integrity error, the TSF shall [assignment: *specify the action to be taken upon integrity error*].

## 11.9 Residual information protection (FDP_RIP)

### Family Behaviour

213 This family addresses the need to ensure that deleted information is no longer accessible, and that newly created objects do not contain information that should not be accessible. This family requires protection for information that has been logically deleted or released, but may still be present within the TOE.

### Component levelling




214 FDP_RIP.1 Subset residual information protection, requires that the TSF ensure that any residual information content of any resources is unavailable to a defined subset of the objects in the TSC upon the resource's allocation or deallocation.

215 FDP_RIP.2 Full residual information protection, requires that the TSF ensure that any residual information content of any resources is unavailable to all objects upon the resource's allocation or deallocation.

### Management: FDP_RIP.1, FDP_RIP.2

216 The following actions could be considered for the management functions in FMT:

a) The choice of when to perform residual information protection (i.e. upon allocation or deallocation) could be made configurable within the TOE.

### Audit: FDP_RIP.1, FDP_RIP.2

217 There are no auditable events foreseen.

### FDP_RIP.1 Subset residual information protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_RIP.1.1 **The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] the following objects: [assignment: *list of objects*].**

## FDP_RIP.2 Full residual information protection

Hierarchical to: FDP_RIP.1 Subset residual information protection

Dependencies: No dependencies.

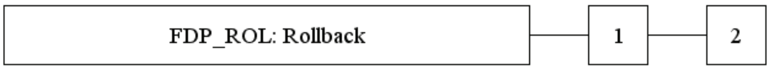
**FDP_RIP.2.1** The TSF shall ensure that any previous information content of a resource is made unavailable upon the [selection: *allocation of the resource to, deallocation of the resource from*] **all** objects.

## 11.10 Rollback (FDP_ROL)

### Family Behaviour

218 The rollback operation involves undoing the last operation or a series of operations, bounded by some limit, such as a period of time, and return to a previous known state. Rollback provides the ability to undo the effects of an operation or series of operations to preserve the integrity of the user data.

### Component levelling




219 FDP_ROL.1 Basic rollback addresses a need to roll back or undo a limited number of operations within the defined bounds.

220 FDP_ROL.2 Advanced rollback addresses the need to roll back or undo all operations within the defined bounds.

### Management: FDP_ROL.1, FDP_ROL.2

221 The following actions could be considered for the management functions in FMT:

a) The boundary limit to which rollback may be performed could be a configurable item within the TOE.

b) Permission to perform a rollback operation could be restricted to a well defined role.

### Audit: FDP_ROL.1, FDP_ROL.2

222 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All successful rollback operations.

b) Basic: All attempts to perform rollback operations.

c) Detailed: All attempts to perform rollback operations, including identification of the types of operations rolled back.

### FDP_ROL.1 Basic rollback

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ROL.1.1 **The TSF shall enforce [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to permit the rollback of the**

**[assignment:** ***list of operations*****] on the [assignment:** ***information and/or list of objects*****].**

FDP_ROL.1.2 **The TSF shall permit operations to be rolled back within the [assignment:** ***boundary limit to which rollback may be performed*****].**

## FDP_ROL.2 Advanced rollback

Hierarchical to: FDP_ROL.1 Basic rollback

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

FDP_ROL.2.1 The TSF shall enforce [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to permit the rollback of **all** the **operations** on the [assignment: *list of objects*].

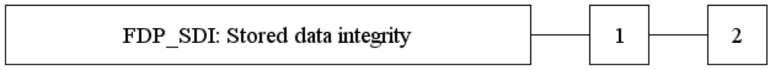
FDP_ROL.2.2 The TSF shall permit operations to be rolled back within the [assignment: *boundary limit to which rollback may be performed*].

## 11.11 Stored data integrity (FDP_SDI)

### Family Behaviour

223 This family provides requirements that address protection of user data while it is stored within the TSC. Integrity errors may affect user data stored in memory, or in a storage device. This family differs from Internal TOE transfer (FDP_ITT) which protects the user data from integrity errors while being transferred within the TOE.

### Component levelling




224 FDP_SDI.1 Stored data integrity monitoring, requires that the SF monitor user data stored within the TSC for identified integrity errors.

225 FDP_SDI.2 Stored data integrity monitoring and action adds the additional capability to the first component by allowing for actions to be taken as a result of an error detection.

### Management:FDP_SDI.1

226 There are no management activities foreseen.

### Management: FDP_SDI.2

227 The following actions could be considered for the management functions in FMT:

a) The actions to be taken upon the detection of an integrity error could be configurable.

### Audit: FDP_SDI.1

228 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Successful attempts to check the integrity of user data, including an indication of the results of the check.

b) Basic: All attempts to check the integrity of user data, including an indication of the results of the check, if performed.

c) Detailed: The type of integrity error that occurred.

### Audit: FDP_SDI.2

229 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Successful attempts to check the integrity of user data, including an indication of the results of the check.

b) Basic: All attempts to check the integrity of user data, including an indication of the results of the check, if performed.

c) Detailed: The type of integrity error that occurred.

d) Detailed: The action taken upon detection of an integrity error.

### FDP_SDI.1 Stored data integrity monitoring

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_SDI.1.1 **The TSF shall monitor user data stored within the TSC for [assignment: *integrity errors*] on all objects, based on the following attributes: [assignment: *user data attributes*].**

### FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to: FDP_SDI.1 Stored data integrity monitoring

Dependencies: No dependencies.

FDP_SDI.2.1 The TSF shall monitor user data stored within the TSC for [assignment: *integrity errors*] on all objects, based on the following attributes: [assignment: *user data attributes*].

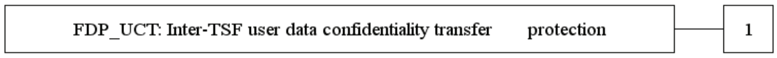
FDP_SDI.2.2 **Upon detection of a data integrity error, the TSF shall [assignment: *action to be taken*].**

## 11.12 Inter-TSF user data confidentiality transfer protection (FDP_UCT)

### Family Behaviour

230 This family defines the requirements for ensuring the confidentiality of user data when it is transferred using an external channel between distinct TOEs or users on distinct TOEs.

### Component levelling




231 In FDP_UCT.1 Basic data exchange confidentiality, the goal is to provide protection from disclosure of user data while in transit.

### Management:FDP_UCT.1

232 There are no management activities foreseen.

### Audit: FDP_UCT.1

233 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The identity of any user or subject using the data exchange mechanisms.

b) Basic: The identity of any unauthorised user or subject attempting to use the data exchange mechanisms.

c) Basic: A reference to the names or other indexing information useful in identifying the user data that was transmitted or received. This could include security attributes associated with the information.

## FDP_UCT.1 Basic data exchange confidentiality

Hierarchical to: No other components.

Dependencies: [FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
[FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]

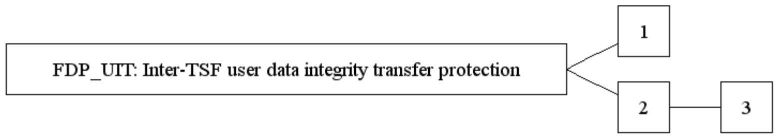
**FDP_UCT.1.1** **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to be able to [selection: *transmit, receive*] objects in a manner protected from unauthorised disclosure.**

## 11.13 Inter-TSF user data integrity transfer protection (FDP_UIT)

### Family Behaviour

234 This family defines the requirements for providing integrity for user data in transit between the TSF and another trusted IT product and recovering from detectable errors. At a minimum, this family monitors the integrity of user data for modifications. Furthermore, this family supports different ways of correcting detected integrity errors.

### Component levelling




235 FDP_UIT.1 Data exchange integrity addresses detection of modifications, deletions, insertions, and replay errors of the user data transmitted.

236 FDP_UIT.2 Source data exchange recovery addresses recovery of the original user data by the receiving TSF with help from the source trusted IT product.

237 FDP_UIT.3 Destination data exchange recovery addresses recovery of the original user data by the receiving TSF on its own without any help from the source trusted IT product.

### Management:FDP_UIT.1, FDP_UIT.2, FDP_UIT.3

238 There are no management activities foreseen.

### Audit: FDP_UIT.1

239 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The identity of any user or subject using the data exchange mechanisms.

b) Basic: The identity of any user or subject attempting to use the user data exchange mechanisms, but who is unauthorised to do so.

c) Basic: A reference to the names or other indexing information useful in identifying the user data that was transmitted or received. This could include security attributes associated with the user data.

d) Basic: Any identified attempts to block transmission of user data.

e) Detailed: The types and/or effects of any detected modifications of transmitted user data.

Audit: FDP_UIT.2, FDP_UIT.3

240 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The identity of any user or subject using the data exchange mechanisms.

b) Minimal: Successful recovery from errors including they type of error that was detected.

c) Basic: The identity of any user or subject attempting to use the user data exchange mechanisms, but who is unauthorised to do so.

d) Basic: A reference to the names or other indexing information useful in identifying the user data that was transmitted or received. This could include security attributes associated with the user data.

e) Basic: Any identified attempts to block transmission of user data.

f) Detailed: The types and/or effects of any detected modifications of transmitted user data.

## FDP_UIT.1 Data exchange integrity

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]

FDP_UIT.1.1 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to be able to [selection: *transmit, receive*] user data in a manner protected from [selection: *modification, deletion, insertion, replay*] errors.**

FDP_UIT.1.2 **The TSF shall be able to determine on receipt of user data, whether [selection: *modification, deletion, insertion, replay*] has occurred.**

## FDP_UIT.2 Source data exchange recovery

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FDP_UIT.1 Data exchange integrity, or
FTP_ITC.1 Inter-TSF trusted channel]

FDP_UIT.2.1 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to be able to recover from [assignment: *list of recoverable errors*] with the help of the source trusted IT product.**

### FDP_UIT.3 Destination data exchange recovery

Hierarchical to: FDP_UIT.2 Source data exchange recovery

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
[FDP_UIT.1 Data exchange integrity, or
FTP_ITC.1 Inter-TSF trusted channel]

**FDP_UIT.3.1** The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to be able to recover from [assignment: *list of recoverable errors*] **without any** help **from** the source trusted IT product.

# 12 Class FIA: Identification and authentication

241 Families in this class address the requirements for functions to establish and verify a claimed user identity.

242 Identification and Authentication is required to ensure that users are associated with the proper security attributes (e.g. identity, groups, roles, security or integrity levels).

243 The unambiguous identification of authorised users and the correct association of security attributes with users and subjects is critical to the enforcement of the intended security policies. The families in this class deal with determining and verifying the identity of users, determining their authority to interact with the TOE, and with the correct association of security attributes for each authorised user. Other classes of requirements (e.g. User Data Protection, Security Audit) are dependent upon correct identification and authentication of users in order to be effective.

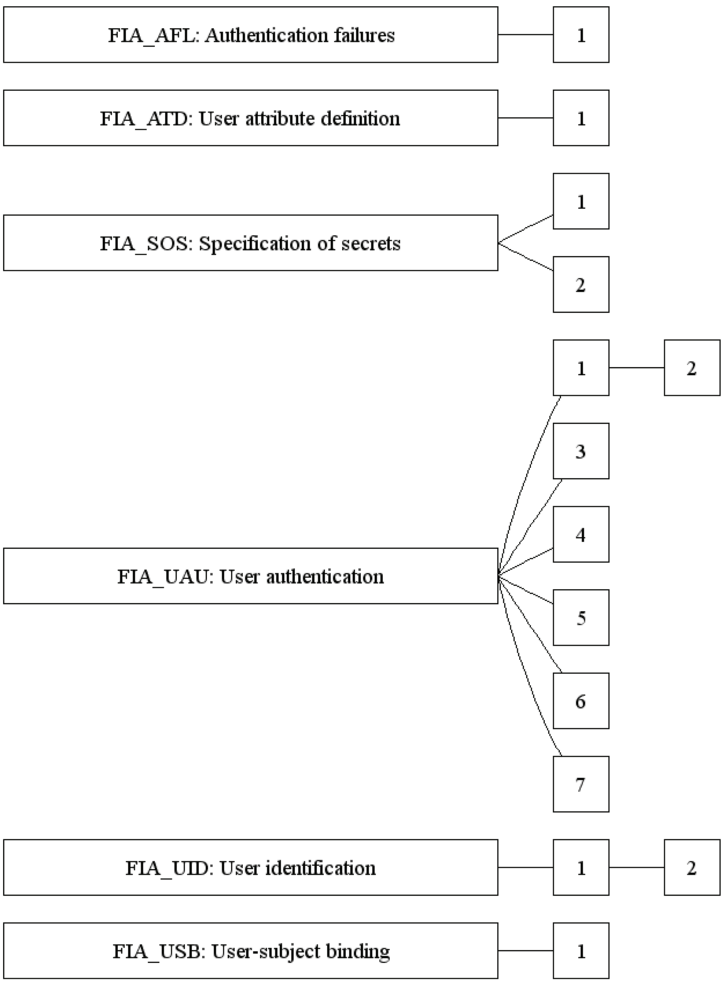



**Figure 13 - FIA: Identification and authentication class decomposition**

## 12.1 Authentication failures (FIA_AFL)

### Family Behaviour

244 This family contains requirements for defining values for some number of unsuccessful authentication attempts and TSF actions in cases of authentication attempt failures. Parameters include, but are not limited to, the number of failed authentication attempts and time thresholds.

### Component levelling


245 FIA_AFL.1 Authentication failure handling, requires that the TSF be able to terminate the session establishment process after a specified number of unsuccessful user authentication attempts. It also requires that, after termination of the session establishment process, the TSF be able to disable the user account or the point of entry (e.g. workstation) from which the attempts were made until an administrator-defined condition occurs.

### Management: FIA_AFL.1

246 The following actions could be considered for the management functions in FMT:

a) management of the threshold for unsuccessful authentication attempts;

b) management of actions to be taken in the event of an authentication failure.

### Audit: FIA_AFL.1

247 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: the reaching of the threshold for the unsuccesful authentication attempts and the actions (e.g. disabling of a terminal) taken and the subsequent, if appropriate, restoration to the normal state (e.g. re-enabling of a terminal).

### FIA_AFL.1 Authentication failure handling

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_AFL.1.1 **The TSF shall detect when [selection: *[assignment: positive integer number], an administrator configurable positive integer within[assignment: range of acceptable values]*] unsuccessful**

**authentication attempts occur related to [assignment: *list of authentication events*].**

FIA_AFL.1.2 **When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [assignment: *list of actions*].**

## 12.2 User attribute definition (FIA_ATD)

### Family Behaviour

248 All authorised users may have a set of security attributes, other than the user's identity, that is used to enforce the TSP. This family defines the requirements for associating user security attributes with users as needed to support the TSP.

### Component levelling


249 FIA_ATD.1 User attribute definition, allows user security attributes for each user to be maintained individually.

### Management: FIA_ATD.1

250 The following actions could be considered for the management functions in FMT:

a) if so indicated in the assignment, the authorised administrator might be able to define additional security attributes for users.

### Audit: FIA_ATD.1

251 There are no auditable events foreseen.

## FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

Dependencies: No dependencies.

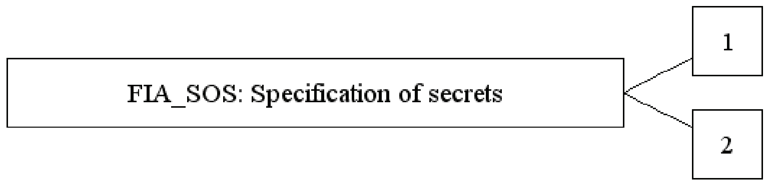
**FIA_ATD.1.1** **The TSF shall maintain the following list of security attributes belonging to individual users: [assignment: *list of security attributes*].**

## 12.3 Specification of secrets (FIA_SOS)

### Family Behaviour

252 This family defines requirements for mechanisms that enforce defined quality metrics on provided secrets and generate secrets to satisfy the defined metric.

### Component levelling




253 FIA_SOS.1 Verification of secrets, requires the TSF to verify that secrets meet defined quality metrics.

254 FIA_SOS.2 TSF Generation of secrets, requires the TSF to be able to generate secrets that meet defined quality metrics.

### Management: FIA_SOS.1

255 The following actions could be considered for the management functions in FMT:

a) the management of the metric used to verify the secrets.

### Management: FIA_SOS.2

256 The following actions could be considered for the management functions in FMT:

a) the management of the metric used to generate the secrets.

### Audit: FIA_SOS.1, FIA_SOS.2

257 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Rejection by the TSF of any tested secret;

b) Basic: Rejection or acceptance by the TSF of any tested secret;

c) Detailed: Identification of any changes to the defined quality metrics.

### FIA_SOS.1 Verification of secrets

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_SOS.1.1 **The TSF shall provide a mechanism to verify that secrets meet [assignment: *a defined quality metric*].**

### FIA_SOS.2 TSF Generation of secrets

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_SOS.2.1 **The TSF shall provide a mechanism to generate secrets that meet [assignment: *a defined quality metric*].**

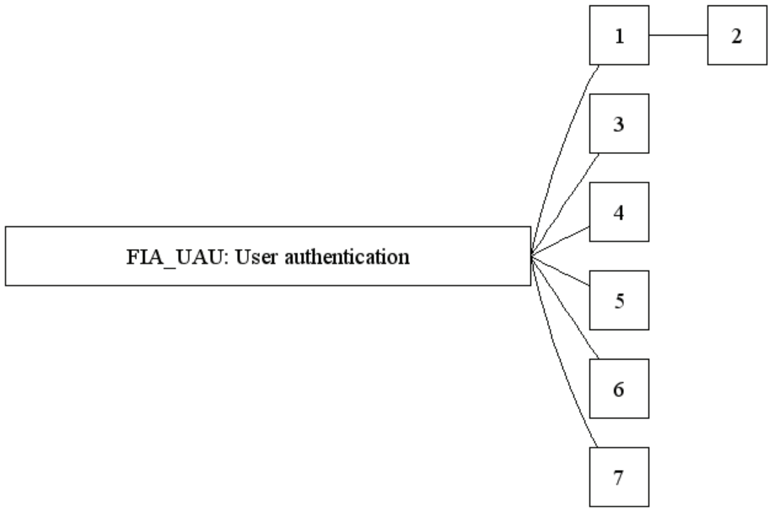
FIA_SOS.2.2 **The TSF shall be able to enforce the use of TSF generated secrets for [assignment: *list of TSF functions*].**

## 12.4 User authentication (FIA_UAU)

### Family Behaviour

258 This family defines the types of user authentication mechanisms supported by the TSF. This family also defines the required attributes on which the user authentication mechanisms must be based.

### Component levelling




259 FIA_UAU.1 Timing of authentication, allows a user to perform certain actions prior to the authentication of the user's identity.

260 FIA_UAU.2 User authentication before any action, requires that users are authenticated before any action will be allowed by the TSF.

261 FIA_UAU.3 Unforgeable authentication Unforgeable authentication, requires the authentication mechanism to be able to detect and prevent the use of authentication data that has been forged or copied.

262 FIA_UAU.4 Single-use authentication mechanisms, requires an authentication mechanism that operates with single-use authentication data.

263 FIA_UAU.5 Multiple authentication mechanisms, requires that different authentication mechanisms be provided and used to authenticate user identities for specific events.

264 FIA_UAU.6 Re-authenticating, requires the ability to specify events for which the user needs to be re-authenticated.

265 FIA_UAU.7 Protected authentication feedback, requires that only limited feedback information is provided to the user during the authentication.

### Management: FIA_UAU.1

266 The following actions could be considered for the management functions in FMT:

a) management of the authentication data by an administrator;

b) management of the authentication data by the associated user;

c) managing the list of actions that can be taken before the user is authenticated.

### Management: FIA_UAU.2

267 The following actions could be considered for the management functions in FMT:

a) management of the authentication data by an administrator;

b) management of the authentication data by the user associated with this data.

### Management:FIA_UAU.3, FIA_UAU.4, FIA_UAU.7

268 There are no management activities foreseen.

### Management: FIA_UAU.5

269 The following actions could be considered for the management functions in FMT:

a) the management of authentication mechanisms;

b) the management of the rules for authentication.

### Management: FIA_UAU.6

270 The following actions could be considered for the management functions in FMT:

a) if an authorised administrator could request re-authentication, the management includes a re-authentication request.

### Audit: FIA_UAU.1

271 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Unsuccessful use of the authentication mechanism;

b) Basic: All use of the authentication mechanism;

c) Detailed: All TSF mediated actions performed before authentication of the user.

### Audit: FIA_UAU.2

272 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Unsuccessful use of the authentication mechanism;

b) Basic: All use of the authentication mechanism.

### Audit: FIA_UAU.3

273 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Detection of fraudulent authentication data;

b) Basic: All immediate measures taken and results of checks on the fraudulent data.

### Audit: FIA_UAU.4

274 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Attempts to reuse authentication data.

### Audit: FIA_UAU.5

275 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The final decision on authentication;

b) Basic: The result of each activated mechanism together with the final decision.

### Audit: FIA_UAU.6

276 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Failure of reauthentication;

b) Basic: All reauthentication attempts.

### Audit: FIA_UAU.7

277 There are no auditable events foreseen.

### FIA_UAU.1 Timing of authentication

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.1.1 **The TSF shall allow [assignment: *list of TSF mediated actions*] on behalf of the user to be performed before the user is authenticated.**

FIA_UAU.1.2 **The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.**

### FIA_UAU.2 User authentication before any action

Hierarchical to: FIA_UAU.1 Timing of authentication

Dependencies: FIA_UID.1 Timing of identification

FIA_UAU.2.1 The TSF shall **require each** user to be **successfully authenticated** before **allowing any other TSF-mediated actions on behalf of that** user.

### FIA_UAU.3 Unforgeable authentication

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.3.1 **The TSF shall [selection: *detect, prevent*] use of authentication data that has been forged by any user of the TSF.**

FIA_UAU.3.2 **The TSF shall [selection: *detect, prevent*] use of authentication data that has been copied from any other user of the TSF.**

### FIA_UAU.4 Single-use authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.4.1 **The TSF shall prevent reuse of authentication data related to [assignment: *identified authentication mechanism(s)*].**

### FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.5.1 **The TSF shall provide [assignment: *list of multiple authentication mechanisms*] to support user authentication.**

FIA_UAU.5.2 **The TSF shall authenticate any user's claimed identity according to the [assignment: *rules describing how the multiple authentication mechanisms provide authentication*].**

### FIA_UAU.6 Re-authenticating

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UAU.6.1 **The TSF shall re-authenticate the user under the conditions [assignment: *list of conditions under which re-authentication is required*].**

### FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

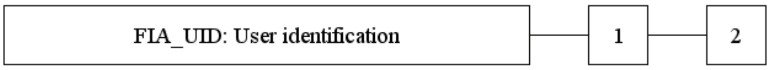
FIA_UAU.7.1 **The TSF shall provide only [assignment: *list of feedback*] to the user while the authentication is in progress.**

## 12.5 User identification (FIA_UID)

### Family Behaviour

278 This family defines the conditions under which users shall be required to identify themselves before performing any other actions that are to be mediated by the TSF and which require user identification.

### Component levelling




279 FIA_UID.1 Timing of identification, allows users to perform certain actions before being identified by the TSF.

280 FIA_UID.2 User identification before any action, requires that users identify themselves before any action will be allowed by the TSF.

### Management: FIA_UID.1

281 The following actions could be considered for the management functions in FMT:

a) the management of the user identities;

b) if an authorised administrator can change the actions allowed before identification, the managing of the action lists.

### Management: FIA_UID.2

282 The following actions could be considered for the management functions in FMT:

a) the management of the user identities.

### Audit: FIA_UID.1, FIA_UID.2

283 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Unsuccessful use of the user identification mechanism, including the user identity provided;

b) Basic: All use of the user identification mechanism, including the user identity provided.

## FIA_UID.1 Timing of identification

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_UID.1.1 **The TSF shall allow [assignment: *list of TSF-mediated actions*] on behalf of the user to be performed before the user is identified.**

FIA_UID.1.2 **The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.**

## FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1 Timing of identification

Dependencies: No dependencies.

FIA_UID.2.1 The TSF shall **require each user to identify itself before allowing any other** TSF-mediated **actions** on behalf of **that** user.

## 12.6 User-subject binding (FIA_USB)

### Family Behaviour

284 An authenticated user, in order to use the TOE, typically activates a subject. The user's security attributes are associated (totally or partially) with this subject. This family defines requirements to create and maintain the association of the user's security attributes to a subject acting on the user's behalf.

### Component levelling


285 FIA_USB.1 User-subject binding, requires the specification of any rules governing the association between user attributes and the subject attributes into which they are mapped.

### Management: FIA_USB.1

286 The following actions could be considered for the management functions in FMT:

a) an authorised administrator can define default subject security attributes.

b) an authorised administrator can change subject security attributes.

### Audit: FIA_USB.1

287 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Unsuccessful binding of user security attributes to a subject (e.g. creation of a subject).

b) Basic: Success and failure of binding of user security attributes to a subject (e.g. success or failure to create a subject).

### FIA_USB.1 User-subject binding

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition

FIA_USB.1.1 **The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: *list of user security attributes*].**

FIA_USB.1.2 **The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: *rules for the initial association of attributes*].**

FIA_USB.1.3 **The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: *rules for the changing of attributes*].**

# 13 Class FMT: Security management

288 This class is intended to specify the management of several aspects of the TSF: security attributes, TSF data and functions. The different management roles and their interaction, such as separation of capability, can be specified.

289 This class has several objectives:

a) management of TSF data, which include, for example, banners;

b) management of security attributes, which include, for example, the Access Control Lists, and Capability Lists;

c) management of functions of the TSF, which includes, for example, the selection of functions, and rules or conditions influencing the behaviour of the TSF;

d) definition of security roles.

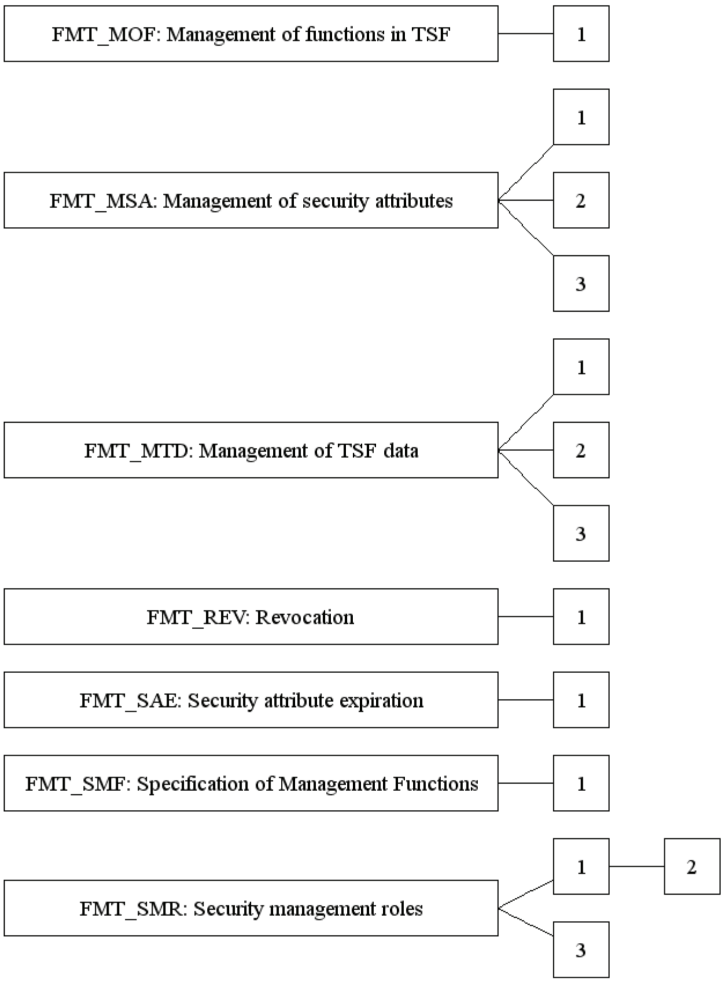



**Figure 14 - FMT: Security management class decomposition**

## 13.1 Management of functions in TSF (FMT_MOF)

### Family Behaviour

290 This family allows authorised users control over the management of functions in the TSF. Examples of functions in the TSF include the audit functions and the multiple authentication functions.

### Component levelling


291 FMT_MOF.1 Management of security functions behaviour allows the authorised users (roles) to manage the behaviour of functions in the TSF that use rules or have specified conditions that may be manageable.

### Management: FMT_MOF.1

292 The following actions could be considered for the management functions in FMT:

a) managing the group of roles that can interact with the functions in the TSF;

### Audit: FMT_MOF.1

293 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: All modifications in the behaviour of the functions in the TSF.

### FMT_MOF.1 Management of security functions behaviour

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

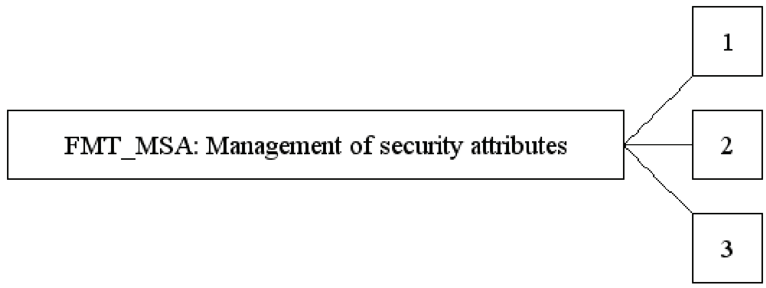
FMT_MOF.1.1 **The TSF shall restrict the ability to [selection: *determine the behaviour of, disable, enable, modify the behaviour of*] the functions [assignment: *list of functions*] to [assignment: *the authorised identified roles*].**

## 13.2 Management of security attributes (FMT_MSA)

### Family Behaviour

294 This family allows authorised users control over the management of security attributes. This management might include capabilities for viewing and modifying of security attributes.

### Component levelling




295 FMT_MSA.1 Management of security attributes allows authorised users (roles) to manage the specified security attributes.

296 FMT_MSA.2 Secure security attributes ensures that values assigned to security attributes are valid with respect to the secure state.

297 FMT_MSA.3 Static attribute initialisation ensures that the default values of security attributes are appropriately either permissive or restrictive in nature.

### Management: FMT_MSA.1

298 The following actions could be considered for the management functions in FMT:

a) managing the group of roles that can interact with the security attributes.

### Management:FMT_MSA.2

299 There are no management activities foreseen.

### Management: FMT_MSA.3

300 The following actions could be considered for the management functions in FMT:

a) managing the group of roles that can specify initial values;

b) managing the permissive or restrictive setting of default values for a given access control SFP.

### Audit: FMT_MSA.1

301 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: All modifications of the values of security attributes.

### Audit: FMT_MSA.2

302 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All offered and rejected values for a security attribute;

b) Detailed: All offered and accepted secure values for a security attribute.

### Audit: FMT_MSA.3

303 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Modifications of the default setting of permissive or restrictive rules.

b) Basic: All modifications of the initial values of security attributes.

## FMT_MSA.1 Management of security attributes

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MSA.1.1 **The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to restrict the ability to [selection: *change_default, query, modify, delete,[assignment: other operations]*] the security attributes [assignment: *list of security attributes*] to [assignment: *the authorised identified roles*].**

## FMT_MSA.2 Secure security attributes

Hierarchical to: No other components.

Dependencies: ADV_SPM.1 Informal TOE security policy model
[FDP_ACC.1 Subset access control, or
FDP_IFC.1 Subset information flow control]
FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.2.1 **The TSF shall ensure that only secure values are accepted for security attributes.**

### FMT_MSA.3 Static attribute initialisation

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3.1 **The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to provide [selection, choose one of: *restrictive, permissive,[assignment: other property]*] default values for security attributes that are used to enforce the SFP.**

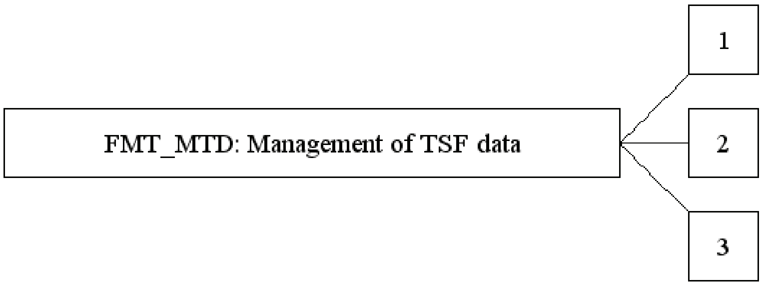
FMT_MSA.3.2 **The TSF shall allow the [assignment: *the authorised identified roles*] to specify alternative initial values to override the default values when an object or information is created.**

## 13.3 Management of TSF data (FMT_MTD)

### Family Behaviour

304 This family allows authorised users (roles) control over the management of TSF data. Examples of TSF data include audit information, clock, system configuration and other TSF configuration parameters.

### Component levelling




305 FMT_MTD.1 Management of TSF data allows authorised users to manage TSF data.

306 FMT_MTD.2 Management of limits on TSF data specifies the action to be taken if limits on TSF data are reached or exceeded.

307 FMT_MTD.3 Secure TSF data ensures that values assigned to TSF data are valid with respect to the secure state.

### Management: FMT_MTD.1

308 The following actions could be considered for the management functions in FMT:

a) managing the group of roles that can interact with the TSF data.

### Management: FMT_MTD.2

309 The following actions could be considered for the management functions in FMT:

a) managing the group of roles that can interact with the limits on the TSF data.

### Management:FMT_MTD.3

310 There are no management activities foreseen.

### Audit: FMT_MTD.1

311 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: All modifications to the values of TSF data.

### Audit: FMT_MTD.2

312 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: All modifications to the limits on TSF data;

b) Basic: All modifications in the actions to be taken in case of violation of the limits.

### Audit: FMT_MTD.3

313 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All rejected values of TSF data.

## FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 **The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear,[assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].**

## FMT_MTD.2 Management of limits on TSF data

Hierarchical to: No other components.

Dependencies: FMT_MTD.1 Management of TSF data
FMT_SMR.1 Security roles

FMT_MTD.2.1 **The TSF shall restrict the specification of the limits for [assignment: *list of TSF data*] to [assignment: *the authorised identified roles*].**

FMT_MTD.2.2 **The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: [assignment: *actions to be taken*].**

## FMT_MTD.3 Secure TSF data

Hierarchical to: No other components.

Dependencies: ADV_SPM.1 Informal TOE security policy model
FMT_MTD.1 Management of TSF data

FMT_MTD.3.1 **The TSF shall ensure that only secure values are accepted for TSF data.**

## 13.4 Revocation (FMT_REV)

### Family Behaviour

314 This family addresses revocation of security attributes for a variety of entities within a TOE.

### Component levelling


315 FMT_REV.1 Revocation provides for revocation of security attributes to be enforced at some point in time.

### Management: FMT_REV.1

316 The following actions could be considered for the management functions in FMT:

a) managing the group of roles that can invoke revocation of security attributes;

b) managing the lists of users, subjects, objects and other resources for which revocation is possible;

c) managing the revocation rules.

### Audit: FMT_REV.1

317 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Unsuccessful revocation of security attributes;

b) Basic: All attempts to revoke security attributes.

### FMT_REV.1 Revocation

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_REV.1.1 **The TSF shall restrict the ability to revoke security attributes associated with the [selection: *users, subjects, objects,[assignment: other additional resources]*] within the TSC to [assignment: *the authorised identified roles*].**

FMT_REV.1.2 **The TSF shall enforce the rules [assignment: *specification of revocation rules*].**

## 13.5 Security attribute expiration (FMT_SAE)

### Family Behaviour

318 This family addresses the capability to enforce time limits for the validity of security attributes.

### Component levelling


319 FMT_SAE.1 Time-limited authorisation provides the capability for an authorised user to specify an expiration time on specified security attributes.

### Management: FMT_SAE.1

320 The following actions could be considered for the management functions in FMT:

a) managing the list of security attributes for which expiration is to be supported;

b) the actions to be taken if the expiration time has passed.

### Audit: FMT_SAE.1

321 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Specification of the expiration time for an attribute;

b) Basic: Action taken due to attribute expiration.

### FMT_SAE.1 Time-limited authorisation

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles
FPT_STM.1 Reliable time stamps

FMT_SAE.1.1 **The TSF shall restrict the capability to specify an expiration time for [assignment: *list of security attributes for which expiration is to be supported*] to [assignment: *the authorised identified roles*].**

FMT_SAE.1.2 **For each of these security attributes, the TSF shall be able to [assignment: *list of actions to be taken for each security attribute*] after the expiration time for the indicated security attribute has passed.**

## 13.6 Specification of Management Functions (FMT_SMF)

### Family Behaviour

322 This family allows the specification of the management functions to be provided by the TOE. Management functions provide TSFI that allow administrators to define the parameters that control the operation of security-related aspects of the TOE, such as data protection attributes, TOE protection attributes, audit attributes, and identification and authentication attributes. Management functions also include those functions performed by an operator to ensure continued operation of the TOE, such as backup and recovery. This family works in conjunction with the other components in the FMT: Security management class: the component in this family calls out the management functions, and other families in FMT: Security management restrict the ability to use these management functions.

### Component levelling


323 FMT_SMF.1 Specification of Management Functions requires that the TSF provide specific management functions.

### Management:FMT_SMF.1

324 There are no management activities foreseen.

### Audit: FMT_SMF.1

325 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Use of the management functions.

### FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

Dependencies: No dependencies.

FMT_SMF.1.1 **The TSF shall be capable of performing the following security management functions: [assignment: *list of security management functions to be provided by the TSF*].**

## 13.7 Security management roles (FMT_SMR)

### Family Behaviour

326 This family is intended to control the assignment of different roles to users. The capabilities of these roles with respect to security management are described in the other families in this class.

### Component levelling


327 FMT_SMR.1 Security roles specifies the roles with respect to security that the TSF recognises.

328 FMT_SMR.2 Restrictions on security roles specifies that in addition to the specification of the roles, there are rules that control the relationship between the roles.

329 FMT_SMR.3 Assuming roles, requires that an explicit request is given to the TSF to assume a role.

### Management: FMT_SMR.1

330 The following actions could be considered for the management functions in FMT:

a) managing the group of users that are part of a role.

### Management: FMT_SMR.2

331 The following actions could be considered for the management functions in FMT:

a) managing the group of users that are part of a role;

b) managing the conditions that the roles must satisfy.

### Management:FMT_SMR.3

332 There are no management activities foreseen.

### Audit: FMT_SMR.1

333 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: modifications to the group of users that are part of a role;

b) Detailed: every use of the rights of a role.

### Audit: FMT_SMR.2

334 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: modifications to the group of users that are part of a role;

b) Minimal: unsuccessful attempts to use a role due to the given conditions on the roles;

c) Detailed: every use of the rights of a role.

### Audit: FMT_SMR.3

335 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: explicit request to assume a role.

## FMT_SMR.1 Security roles

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.1.1 **The TSF shall maintain the roles [assignment: *the authorised identified roles*].**

FMT_SMR.1.2 **The TSF shall be able to associate users with roles.**

## FMT_SMR.2 Restrictions on security roles

Hierarchical to: FMT_SMR.1 Security roles

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.2.1 The TSF shall maintain the roles: [assignment: *authorised identified roles*].

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 **The TSF shall ensure that the conditions [assignment: *conditions for the different roles*] are satisfied.**

## FMT_SMR.3 Assuming roles

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMR.3.1 **The TSF shall require an explicit request to assume the following roles: [assignment: *the roles*].**

# 14 Class FPR: Privacy

336 This class contains privacy requirements. These requirements provide a user protection against discovery and misuse of identity by other users.

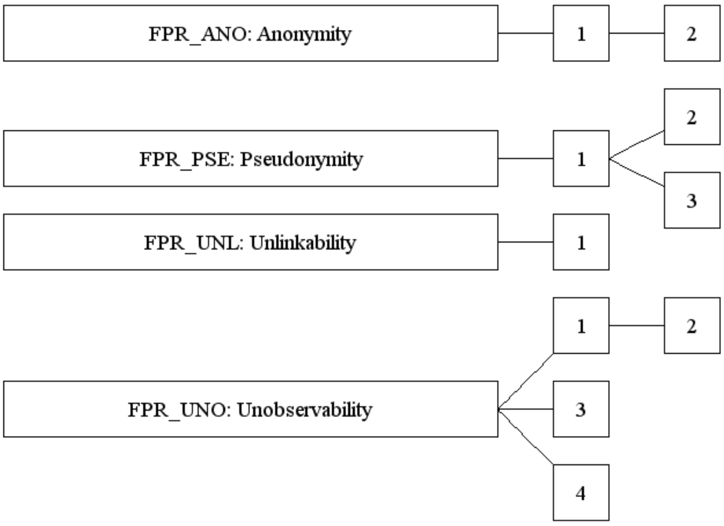


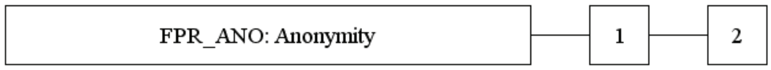

**Figure 15 - FPR: Privacy class decomposition**

## 14.1 Anonymity (FPR_ANO)

### Family Behaviour

337 This family ensures that a user may use a resource or service without disclosing the user's identity. The requirements for Anonymity provide protection of the user identity. Anonymity is not intended to protect the subject identity.

### Component levelling




338 FPR_ANO.1 Anonymity, requires that other users or subjects are unable to determine the identity of a user bound to a subject or operation.

339 FPR_ANO.2 Anonymity without soliciting information enhances the requirements of FPR_ANO.1 Anonymity by ensuring that the TSF does not ask for the user identity.

### Management:FPR_ANO.1, FPR_ANO.2

340 There are no management activities foreseen.

### Audit: FPR_ANO.1, FPR_ANO.2

341 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The invocation of the anonymity mechanism.

### FPR_ANO.1 Anonymity

Hierarchical to: No other components.

Dependencies: No dependencies.

FPR_ANO.1.1 **The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine the real user name bound to [assignment: *list of subjects and/or operations and/or objects*].**

### FPR_ANO.2 Anonymity without soliciting information

Hierarchical to: FPR_ANO.1 Anonymity

Dependencies: No dependencies.

FPR_ANO.2.1 The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine the real user name bound to [assignment: *list of subjects and/or operations and/or objects*].

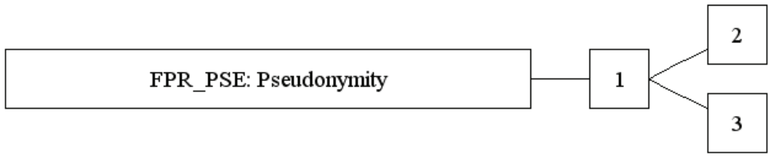
FPR_ANO.2.2 **The TSF shall provide [assignment: *list of services*] to [assignment: *list of subjects*] without soliciting any reference to the real user name.**

## 14.2 Pseudonymity (FPR_PSE)

### Family Behaviour

342 This family ensures that a user may use a resource or service without disclosing its user identity, but can still be accountable for that use.

### Component levelling




343 FPR_PSE.1 Pseudonymity requires that a set of users and/or subjects are unable to determine the identity of a user bound to a subject or operation, but that this user is still accountable for its actions.

344 FPR_PSE.2 Reversible pseudonymity, requires the TSF to provide a capability to determine the original user identity based on a provided alias.

345 FPR_PSE.3 Alias pseudonymity, requires the TSF to follow certain construction rules for the alias to the user identity.

### Management:FPR_PSE.1, FPR_PSE.2, FPR_PSE.3

346 There are no management activities foreseen.

### Audit: FPR_PSE.1, FPR_PSE.2, FPR_PSE.3

347 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The subject/user that requested resolution of the user identity should be audited.

## FPR_PSE.1 Pseudonymity

Hierarchical to: No other components.

Dependencies: No dependencies.

FPR_PSE.1.1 **The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine the real user name bound to [assignment: *list of subjects and/or operations and/or objects*].**

FPR_PSE.1.2 **The TSF shall be able to provide [assignment: *number of aliases*] aliases of the real user name to [assignment: *list of subjects*].**

FPR_PSE.1.3 **The TSF shall [selection, choose one of: *determine an alias for a user, accept the alias from the user*] and verify that it conforms to the [assignment: *alias metric*].**

## FPR_PSE.2 Reversible pseudonymity

Hierarchical to: FPR_PSE.1 Pseudonymity

Dependencies: FIA_UID.1 Timing of identification

**FPR_PSE.2.1** The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine the real user name bound to [assignment: *list of subjects and/or operations and/or objects*].

**FPR_PSE.2.2** The TSF shall be able to provide [assignment: *number of aliases*] aliases of the real user name to [assignment: *list of subjects*].

**FPR_PSE.2.3** The TSF shall [selection, choose one of: *determine an alias for a user, accept the alias from the user*] and verify that it conforms to the [assignment: *alias metric*].

**FPR_PSE.2.4** **The TSF shall provide [selection: *an authorised user,[assignment: list of trusted subjects]*] a capability to determine the user identity based on the provided alias only under the following [assignment: *list of conditions*].**

## FPR_PSE.3 Alias pseudonymity

Hierarchical to: FPR_PSE.1 Pseudonymity

Dependencies: No dependencies.

**FPR_PSE.3.1** The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine the real user name bound to [assignment: *list of subjects and/or operations and/or objects*].

**FPR_PSE.3.2** The TSF shall be able to provide [assignment: *number of aliases*] aliases of the real user name to [assignment: *list of subjects*].

**FPR_PSE.3.3** The TSF shall [selection, choose one of: *determine an alias for a user, accept the alias from the user*] and verify that it conforms to the [assignment: *alias metric*].

**FPR_PSE.3.4** **The TSF shall provide an alias to the real user name which shall be identical to an alias provided previously under the following [assignment: *list of conditions*] otherwise the alias provided shall be unrelated to previously provided aliases.**

## 14.3 Unlinkability (FPR_UNL)

### Family Behaviour

348 This family ensures that a user may make multiple uses of resources or services without others being able to link these uses together.

### Component levelling


349 FPR_UNL.1 Unlinkability, requires that users and/or subjects are unable to determine whether the same user caused certain specific operations in the system.

### Management: FPR_UNL.1

350 The following actions could be considered for the management functions in FMT:

a) the management of the unlinkability function.

### Audit: FPR_UNL.1

351 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The invocation of the unlinkability mechanism.

### FPR_UNL.1 Unlinkability

Hierarchical to: No other components.

Dependencies: No dependencies.

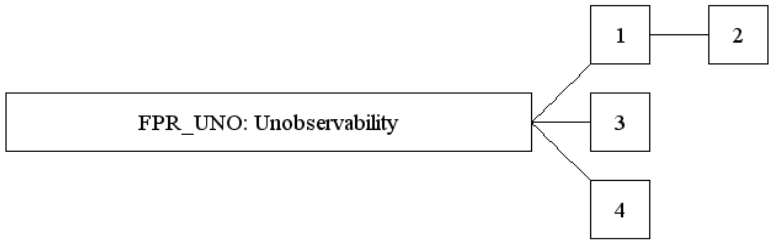
FPR_UNL.1.1 **The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine whether [assignment: *list of operations*][selection: *were caused by the same user, are related as follows[assignment: list of relations]*].**

## 14.4 Unobservability (FPR_UNO)

### Family Behaviour

352 This family ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used.

### Component levelling




353 FPR_UNO.1 Unobservability, requires that users and/or subjects cannot determine whether an operation is being performed.

354 FPR_UNO.2 Allocation of information impacting unobservability, requires that the TSF provide specific mechanisms to avoid the concentration of privacy related information within the TOE. Such concentrations might impact unobservability if a security compromise occurs.

355 FPR_UNO.3 Unobservability without soliciting information,q requires that the TSF does not try to obtain privacy related information that might be used to compromise unobservability.

356 FPR_UNO.4 Authorised user observability, requires the TSF to provide one or more authorised users with a capability to observe the usage of resources and/or services.

### Management: FPR_UNO.1, FPR_UNO.2

357 The following actions could be considered for the management functions in FMT:

a) the management of the behaviour of the unobservability function.

### Management:FPR_UNO.3

358 There are no management activities foreseen.

### Management: FPR_UNO.4

359 The following actions could be considered for the management functions in FMT:

a) the list of authorised users that are capable of determining the occurence of operations.

### Audit: FPR_UNO.1, FPR_UNO.2

360 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The invocation of the unobservability mechanism.

### Audit: FPR_UNO.3

361 There are no auditable events foreseen.

### Audit: FPR_UNO.4

362 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: The observation of the use of a resource or service by a user or subject.

## FPR_UNO.1 Unobservability

Hierarchical to: No other components.

Dependencies: No dependencies.

FPR_UNO.1.1 **The TSF shall ensure that [assignment: *list of users and/or subjects*] are unable to observe the operation [assignment: *list of operations*] on [assignment: *list of objects*] by [assignment: *list of protected users and/or subjects*].**

## FPR_UNO.2 Allocation of information impacting unobservability

Hierarchical to: FPR_UNO.1 Unobservability

Dependencies: No dependencies.

FPR_UNO.2.1 The TSF shall ensure that [assignment: *list of users and/or subjects*] are unable to observe the operation [assignment: *list of operations*] on [assignment: *list of objects*] by [assignment: *list of protected users and/or subjects*].

FPR_UNO.2.2 **The TSF shall allocate the [assignment: *unobservability related information*] among different parts of the TOE such that the following conditions hold during the lifetime of the information: [assignment: *list of conditions*].**

### FPR_UNO.3 Unobservability without soliciting information

Hierarchical to: No other components.

Dependencies: FPR_UNO.1 Unobservability

FPR_UNO.3.1 **The TSF shall provide [assignment: *list of services*] to [assignment: *list of subjects*] without soliciting any reference to [assignment: *privacy related information*].**

### FPR_UNO.4 Authorised user observability

Hierarchical to: No other components.

Dependencies: No dependencies.

FPR_UNO.4.1 **The TSF shall provide [assignment: *set of authorised users*] with the capability to observe the usage of [assignment: *list of resources and/or services*].**

# 15 Class FPT: Protection of the TSF

363 This class contains families of functional requirements that relate to the integrity and management of the mechanisms that provide the TSF (independent of TSP-specifics) and to the integrity of TSF data (independent of the specific contents of the TSP data). In some sense, families in this class may appear to duplicate components in the FDP: User data protection class; they may even be implemented using the same mechanisms. However, FDP: User data protection focuses on user data protection, while FPT: Protection of the TSF focuses on TSF data protection. In fact, components from the FPT: Protection of the TSF class are necessary to provide requirements that the SFPs in the TOE cannot be tampered with or bypassed.

364 From the point of view of this class, there are three significant portions for the TSF:

a) The TSF's abstract machine, which is the virtual or physical machine upon which the specific TSF implementation under evaluation executes.

b) The TSF's implementation, which executes on the abstract machine and implements the mechanisms that enforce the TSP.

c) The TSF's data, which are the administrative databases that guide the enforcement of the TSP.

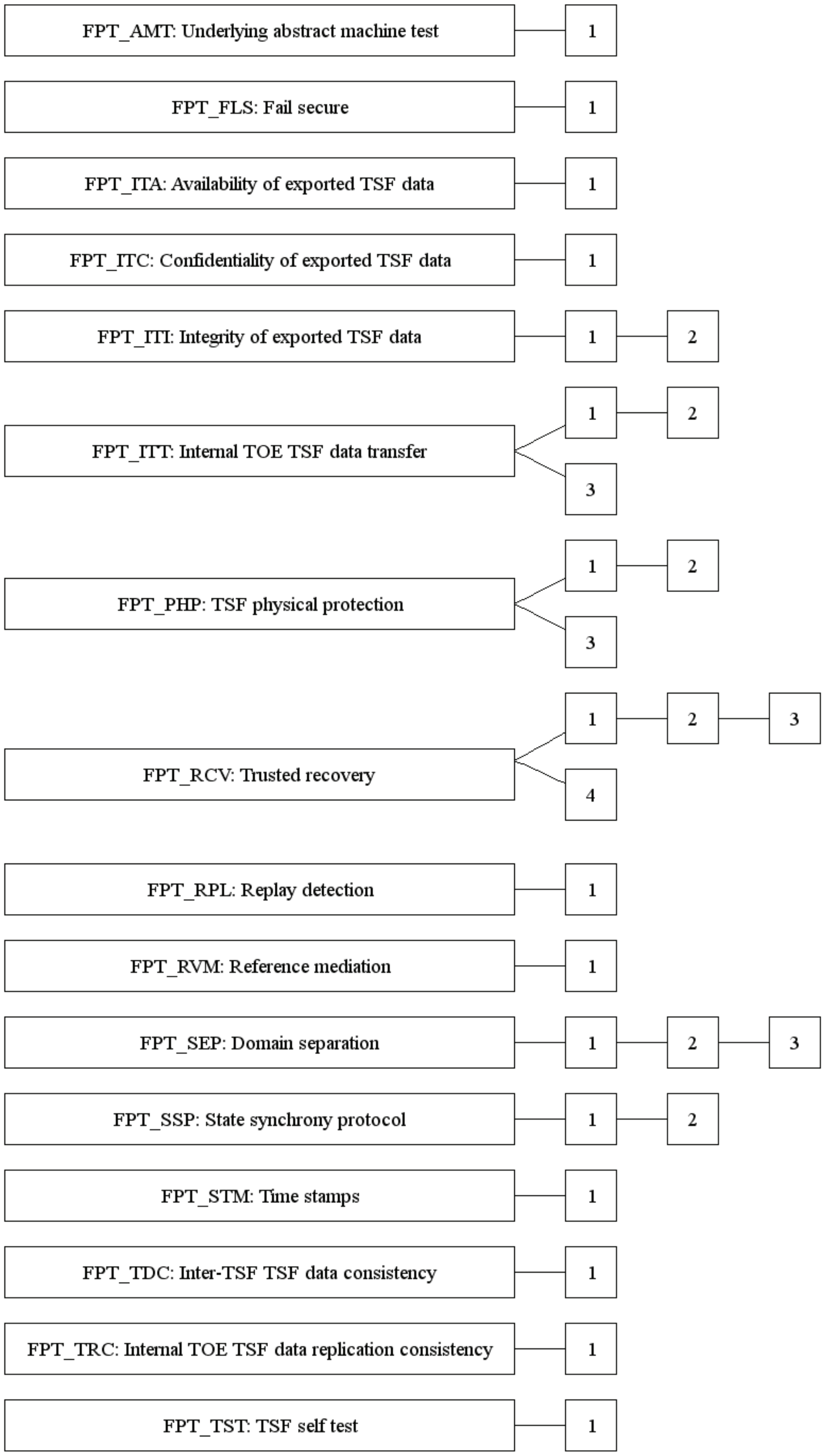



**Figure 16 - FPT: Protection of the TSF class decomposition**

## 15.1 Underlying abstract machine test (FPT_AMT)

### Family Behaviour

365 This family defines requirements for the TSF to perform testing to demonstrate the security assumptions made about the underlying abstract machine upon which the TSF relies. This “abstract” machine could be a hardware/firmware platform, or it could be some known and assessed hardware/software combination acting as a virtual machine.

### Component levelling


366 FPT_AMT.1 Abstract machine testing, provides for testing of the underlying abstract machine.

### Management: FPT_AMT.1

367 The following actions could be considered for the management functions in FMT:

a) management of the conditions under which abstract machine test occurs, such as during initial start-up, regular interval, or under specified conditions;

b) management of the time interval if appropriate.

### Audit: FPT_AMT.1

368 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Execution of the tests of the underlying machine and the results of the tests.

### FPT_AMT.1 Abstract machine testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_AMT.1.1 **The TSF shall run a suite of tests [selection: *during initial start-up, periodically during normal operation, at the request of an authorised user,[assignment: other conditions]*] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.**

## 15.2 Fail secure (FPT_FLS)

### Family Behaviour

369 The requirements of this family ensure that the TOE will not violate its TSP in the event of identified categories of failures in the TSF.

### Component levelling


370 This family consists of only one component, FPT_FLS.1 Failure with preservation of secure state, which requires that the TSF preserve a secure state in the face of the identified failures.

### Management:FPT_FLS.1

371 There are no management activities foreseen.

### Audit: FPT_FLS.1

372 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Failure of the TSF.

### FPT_FLS.1 Failure with preservation of secure state

Hierarchical to: No other components.

Dependencies: ADV_SPM.1 Informal TOE security policy model

**FPT_FLS.1.1** **The TSF shall preserve a secure state when the following types of failures occur: [assignment: *list of types of failures in the TSF*].**

## 15.3 Availability of exported TSF data (FPT_ITA)

### Family Behaviour

373 This family defines the rules for the prevention of loss of availability of TSF data moving between the TSF and a remote trusted IT product. This data could, for example, be TSF critical data such as passwords, keys, audit data, or TSF executable code.

### Component levelling


374 This family consists of only one component, FPT_ITA.1 Inter-TSF availability within a defined availability metric. This component requires that the TSF ensure, to an identified degree of probability, the availability of TSF data provided to a remote trusted IT product.

### Management: FPT_ITA.1

375 The following actions could be considered for the management functions in FMT:

a) management of the list of types of TSF data that must be available to a remote trusted IT product.

### Audit: FPT_ITA.1

376 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: the absence of TSF data when required by a TOE.

### FPT_ITA.1 Inter-TSF availability within a defined availability metric

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_ITA.1.1 **The TSF shall ensure the availability of [assignment: *list of types of TSF data*] provided to a remote trusted IT product within [assignment: *a defined availability metric*] given the following conditions [assignment: *conditions to ensure availability*].**

## 15.4 Confidentiality of exported TSF data (FPT_ITC)

### Family Behaviour

377 This family defines the rules for the protection from unauthorised disclosure of TSF data during transmission between the TSF and a remote trusted IT product. This data could, for example, be TSF critical data such as passwords, keys, audit data, or TSF executable code.

### Component levelling


378 This family consists of only one component, FPT_ITC.1 Inter-TSF confidentiality during transmission, which requires that the TSF ensure that data transmitted between the TSF and a remote trusted IT product is protected from disclosure while in transit.

### Management:FPT_ITC.1

379 There are no management activities foreseen.

### Audit: FPT_ITC.1

380 There are no auditable events foreseen.

### FPT_ITC.1 Inter-TSF confidentiality during transmission

Hierarchical to: No other components.

Dependencies: No dependencies.

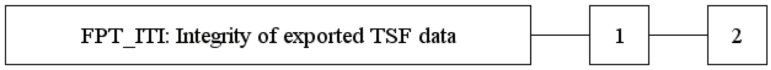
FPT_ITC.1.1 **The TSF shall protect all TSF data transmitted from the TSF to a remote trusted IT product from unauthorised disclosure during transmission.**

## 15.5 Integrity of exported TSF data (FPT_ITI)

### Family Behaviour

381 This family defines the rules for the protection, from unauthorised modification, of TSF data during transmission between the TSF and a remote trusted IT product. This data could, for example, be TSF critical data such as passwords, keys, audit data, or TSF executable code.

### Component levelling




382 FPT_ITI.1 Inter-TSF detection of modification, provides the ability to detect modification of TSF data during transmission between the TSF and a remote trusted IT product, under the assumption that the remote trusted IT product is cognisant of the mechanism used.

383 FPT_ITI.2 Inter-TSF detection and correction of modification, provides the ability for the remote trusted IT product not only to detect modification, but to correct modified TSF data under the assumption that the remote trusted IT product is cognisant of the mechanism used.

### Management:FPT_ITI.1

384 There are no management activities foreseen.

### Management: FPT_ITI.2

385 The following actions could be considered for the management functions in FMT:

a) management of the types of TSF data that the TSF should try to correct if modified in transit;

b) management of the types of action that the TSF could take if TSF data is modified in transit.

### Audit: FPT_ITI.1

386 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: the detection of modification of transmitted TSF data.

b) Basic: the action taken upon detection of modification of transmitted TSF data.

Audit: FPT_ITI.2

387 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: the detection of modification of transmitted TSF data;

b) Basic: the action taken upon detection of modification of transmitted TSF data.

c) Basic: the use of the correction mechanism.

### FPT_ITI.1 Inter-TSF detection of modification

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_ITI.1.1 **The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and a remote trusted IT product within the following metric: [assignment: *a defined modification metric*].**

FPT_ITI.1.2 **The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and a remote trusted IT product and perform [assignment: *action to be taken*] if modifications are detected.**

### FPT_ITI.2 Inter-TSF detection and correction of modification

Hierarchical to: FPT_ITI.1 Inter-TSF detection of modification

Dependencies: No dependencies.

FPT_ITI.2.1 The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and a remote trusted IT product within the following metric: [assignment: *a defined modification metric*].

FPT_ITI.2.2 The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and a remote trusted IT product and perform [assignment: *action to be taken*] if modifications are detected.

FPT_ITI.2.3 **The TSF shall provide the capability to correct [assignment: *type of modification*] of all TSF data transmitted between the TSF and a remote trusted IT product.**

## 15.6 Internal TOE TSF data transfer (FPT_ITT)

### Family Behaviour

388 This family provides requirements that address protection of TSF data when it is transferred between separate parts of a TOE across an internal channel.

### Component levelling


389 FPT_ITT.1 Basic internal TSF data transfer protection, requires that TSF data be protected when transmitted between separate parts of the TOE.

390 FPT_ITT.2 TSF data transfer separation, requires that the TSF separate user data from TSF data during transmission.

391 FPT_ITT.3 TSF data integrity monitoring, requires that the TSF data transmitted between separate parts of the TOE is monitored for identified integrity errors.

### Management: FPT_ITT.1

392 The following actions could be considered for the management functions in FMT:

a) management of the types of modification against which the TSF should protect;

b) management of the mechanism used to provide the protection of the data in transit between different parts of the TSF.

### Management: FPT_ITT.2

393 The following actions could be considered for the management functions in FMT:

a) management of the types of modification against which the TSF should protect;

b) management of the mechanism used to provide the protection of the data in transit between different parts of the TSF;

c) management of the separation mechanism.

### Management: FPT_ITT.3

394 The following actions could be considered for the management functions in FMT:

a) management of the types of modification against which the TSF should protect;

b) management of the mechanism used to provide the protection of the data in transit between different parts of the TSF;

c) management of the types of modification of TSF data the TSF should try to detect;

d) management of the action>s that will be taken.

Audit: FPT_ITT.1, FPT_ITT.2

395 There are no auditable events foreseen.

Audit: FPT_ITT.3

396 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: the detection of modification of TSF data;

b) Basic: the action taken following detection of an integrity error.

## FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_ITT.1.1 **The TSF shall protect TSF data from [selection: *disclosure, modification*] when it is transmitted between separate parts of the TOE.**

## FPT_ITT.2 TSF data transfer separation

Hierarchical to: FPT_ITT.1 Basic internal TSF data transfer protection

Dependencies: No dependencies.

FPT_ITT.2.1 The TSF shall protect TSF data from [selection: *disclosure, modification*] when it is transmitted between separate parts of the TOE.

FPT_ITT.2.2 **The TSF shall separate user data from TSF data when such data is transmitted between separate parts of the TOE.**

## FPT_ITT.3 TSF data integrity monitoring

Hierarchical to: No other components.

Dependencies: FPT_ITT.1 Basic internal TSF data transfer protection

FPT_ITT.3.1 **The TSF shall be able to detect [selection: *modification of data, substitution of data, re-ordering of data, deletion of data*,,[assignment:**

***other integrity errors]*] for TSF data transmitted between separate parts of the TOE.**

FPT_ITT.3.2 **Upon detection of a data integrity error, the TSF shall take the following actions: [assignment: *specify the action to be taken*].**

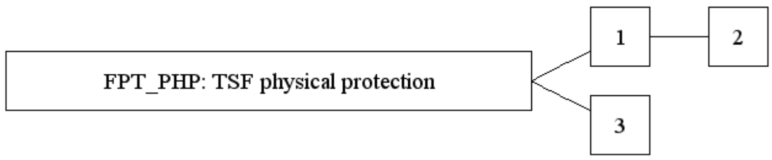
## 15.7 TSF physical protection (FPT_PHP)

### Family Behaviour

397 TSF physical protection components refer to restrictions on unauthorised physical access to the TSF, and to the deterrence of, and resistance to, unauthorised physical modification, or substitution of the TSF.

398 The requirements of components in this family ensure that the TSF is protected from physical tampering and interference. Satisfying the requirements of these components results in the TSF being packaged and used in such a manner that physical tampering is detectable, or resistance to physical tampering is enforced. Without these components, the protection functions of a TSF lose their effectiveness in environments where physical damage cannot be prevented. This family also provides requirements regarding how the TSF shall respond to physical tampering attempts.

### Component levelling




399 FPT_PHP.1 Passive detection of physical attack, provides for features that indicate when a TSF device or TSF element is subject to tampering. However, notification of tampering is not automatic; an authorised user must invoke a security administrative function or perform manual inspection to determining if tampering has occurred.

400 FPT_PHP.2 Notification of physical attack, provides for automatic notification of tampering for an identified subset of physical penetrations.

401 FPT_PHP.3 Resistance to physical attack, provides for features that prevent or resist physical tampering with TSF devices and TSF elements.

### Management: FPT_PHP.1

402 The following actions could be considered for the management functions in FMT:

a) management of the user or role that determines whether physical tampering has occurred.

### Management: FPT_PHP.2

403 The following actions could be considered for the management functions in FMT:

a) management of the user or role that gets informed about intrusions;

b) management of the list of devices that should inform the indicated user or role about the intrusion.

#### Management: FPT_PHP.3

404 The following actions could be considered for the management functions in FMT:

a) management of the automatic responses to physical tampering.

#### Audit: FPT_PHP.1

405 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: if detection by IT means, detection of intrusion.

#### Audit: FPT_PHP.2

406 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: detection of intrusion.

#### Audit: FPT_PHP.3

407 There are no auditable events foreseen.

### FPT_PHP.1 Passive detection of physical attack

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_PHP.1.1 **The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.**

FPT_PHP.1.2 **The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.**

### FPT_PHP.2 Notification of physical attack

Hierarchical to: FPT_PHP.1 Passive detection of physical attack

Dependencies: FMT_MOF.1 Management of security functions behaviour

FPT_PHP.2.1 The TSF shall provide unambiguous detection of physical tampering that might compromise the TSF.

FPT_PHP.2.2 The TSF shall provide the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred.

FPT_PHP.2.3 **For [assignment: *list of TSF devices/elements for which active detection is required*], the TSF shall monitor the devices and elements and notify [assignment: *a designated user or role*] when physical tampering with the TSF's devices or TSF's elements has occurred.**

## FPT_PHP.3 Resistance to physical attack

Hierarchical to: No other components.

Dependencies: No dependencies.

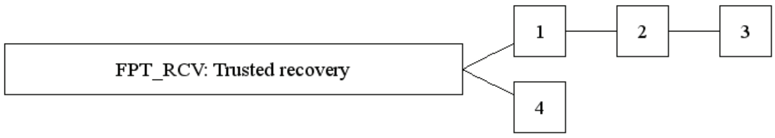
FPT_PHP.3.1 **The TSF shall resist [assignment: *physical tampering scenarios*] to the [assignment: *list of TSF devices/elements*] by responding automatically such that the TSP is not violated.**

## 15.8 Trusted recovery (FPT_RCV)

### Family Behaviour

408 The requirements of this family ensure that the TSF can determine that the TOE is started up without protection compromise and can recover without protection compromise after discontinuity of operations. This family is important because the start-up state of the TSF determines the protection of subsequent states.

### Component levelling




409 FPT_RCV.1 Manual recovery, allows a TOE to only provide mechanisms that involve human intervention to return to a secure state.

410 FPT_RCV.2 Automated recovery, provides, for at least one type of service discontinuity, recovery to a secure state without human intervention; recovery for other discontinuities may require human intervention.

411 FPT_RCV.3 Automated recovery without undue loss, also provides for automated recovery, but strengthens the requirements by disallowing undue loss of protected objects.

412 FPT_RCV.4 Function recovery, provides for recovery at the level of particular SFs, ensuring either successful completion or rollback of TSF data to a secure state.

### Management: FPT_RCV.1

413 The following actions could be considered for the management functions in FMT:

a) management of who can access the restore capability within the maintenance mode.

### Management: FPT_RCV.2, FPT_RCV.3

414 The following actions could be considered for the management functions in FMT:

a) management of who can access the restore capability within the maintenance mode;

b) management of the list of failures/service discontinuities that will be handled through the automatic procedures.

### Management:FPT_RCV.4

415 There are no management activities foreseen.

### Audit: FPT_RCV.1, FPT_RCV.2, FPT_RCV.3

416 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: the fact that a failure or service discontinuity occurred;

b) Minimal: resumption of the regular operation;

c) Basic: type of failure or service discontinuity.

### Audit: FPT_RCV.4

417 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: if possible, the impossibility to return to a secure state after failure of a security function;

b) Basic: if possible, the detection of a failure of a security function.

## FPT_RCV.1 Manual recovery

Hierarchical to: No other components.

Dependencies: AGD_ADM.1 Administrator guidance
ADV_SPM.1 Informal TOE security policy model

FPT_RCV.1.1 **After [assignment: *list of failures/service discontinuities*] the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.**

## FPT_RCV.2 Automated recovery

Hierarchical to: FPT_RCV.1 Manual recovery

Dependencies: AGD_ADM.1 Administrator guidance
ADV_SPM.1 Informal TOE security policy model

FPT_RCV.2.1 **When automated recovery from** [assignment: *list of failures/service discontinuities*] **is not possible,** the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.2.2 **For [assignment: *list of failures/service discontinuities*], the TSF shall ensure the return of the TOE to a secure state using automated procedures.**

## FPT_RCV.3 Automated recovery without undue loss

Hierarchical to: FPT_RCV.2 Automated recovery

Dependencies: AGD_ADM.1 Administrator guidance
ADV_SPM.1 Informal TOE security policy model

FPT_RCV.3.1 When automated recovery from [assignment: *list of failures/service discontinuities*] is not possible, the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RCV.3.2 For [assignment: *list of failures/service discontinuities*], the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.3.3 **The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding [assignment: *quantification*] for loss of TSF data or objects within the TSC.**

FPT_RCV.3.4 **The TSF shall provide the capability to determine the objects that were or were not capable of being recovered.**

## FPT_RCV.4 Function recovery

Hierarchical to: No other components.

Dependencies: ADV_SPM.1 Informal TOE security policy model

FPT_RCV.4.1 **The TSF shall ensure that [assignment: *list of SFs and failure scenarios*] have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.**

## 15.9 Replay detection (FPT_RPL)

### Family Behaviour

418 This family addresses detection of replay for various types of entities (e.g. messages, service requests, service responses) and subsequent actions to correct. In the case where replay may be detected, this effectively prevents it.

### Component levelling


419 The family consists of only one component, FPT_RPL.1 Replay detection, which requires that the TSF shall be able to detect the replay of identified entities.

### Management: FPT_RPL.1

420 The following actions could be considered for the management functions in FMT:

a) management of the list of identified entities for which replay shall be detected;

b) management of the list of actions that need to be taken in case of replay.

### Audit: FPT_RPL.1

421 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Detected replay attacks.

b) Detailed: Action to be taken based on the specific actions.

### FPT_RPL.1 Replay detection

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_RPL.1.1 **The TSF shall detect replay for the following entities: [assignment: *list of identified entities*].**

FPT_RPL.1.2 **The TSF shall perform [assignment: *list of specific actions*] when replay is detected.**

## 15.10 Reference mediation (FPT_RVM)

### Family Behaviour

422 The requirements of this family address the "always invoked" aspect of a traditional reference monitor. The goal of this family is to ensure, with respect to a given SFP, that all actions requiring policy enforcement are validated by the TSF against the SFP. If the portion of the TSF that enforces the SFP also meets the requirements of appropriate components from Domain separation (FPT_SEP) and TSF internals (ADV_INT), then that portion of the TSF provides a "reference monitor" for that SFP.

423 A TSF that implements a SFP provides effective protection against unauthorised operation if and only if all enforceable actions (e.g. accesses to objects) requested by untrusted subjects with respect to any or all of that SFP are validated by the TSF before succeeding. If an action that could be enforceable by the TSF, is incorrectly enforced or incorrectly bypassed, the overall enforcement of the SFP could be compromised. Subjects could then bypass the SFP in a variety of unauthorised ways (e.g. circumvent access checks for some subjects or objects, bypass checks for objects whose protection was assumed by applications, retain access rights beyond their intended lifetime, bypass auditing of audited actions, or bypass authentication). Note that some subjects, the so called "trusted subjects" with respect to a specific SFP, might be trusted to enforce the SFP by themselves, and bypass the mediation of the SFP.

### Component levelling


424 This family consists of only one component, FPT_RVM.1 Non-bypassability of the TSP, which requires non-bypassability for all SFPs in the TSP.

### Management:FPT_RVM.1

425 There are no management activities foreseen.

### Audit: FPT_RVM.1

426 There are no auditable events foreseen.

### FPT_RVM.1 Non-bypassability of the TSP

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_RVM.1.1 **The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.**

## 15.11 Domain separation (FPT_SEP)

### Family Behaviour

427 The components of this family ensure that at least one security domain is available for the TSF's own execution and that the TSF is protected from external interference and tampering (e.g. by modification of TSF code or data structures) by untrusted subjects. Satisfying the requirements of this family makes the TSF self-protecting, meaning that an untrusted subject cannot modify or damage the TSF.

428 This family requires the following:

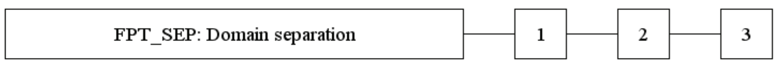
a) The resources of the TSF's security domain (“protected domain”) and those of subjects and unconstrained entities external to the domain are separated such that the entities external to the protected domain cannot observe or modify TSF data or TSF code internal to the protected domain.

b) The transfers between domains are controlled such that arbitrary entry to, or return from, the protected domain is not possible.

c) The user or application parameters passed to the protected domain by addresses are validated with respect to the protected domain's address space, and those passed by value are validated with respect to the values expected by the protected domain.

d) The security domains of subjects are distinct except for controlled sharing via the TSF.

### Component levelling




429 FPT_SEP.1 TSF domain separation, provides a distinct protected domain for the TSF and provides separation between subjects within the TSC.

430 FPT_SEP.2 SFP domain separation, requires that the TSF be further subdivided, with distinct domain(s) for an identified set of SFPs that act as reference monitors for their policies, and a domain for the remainder of the TSF, as well as domains for the non-TSF portions of the TOE.

431 FPT_SEP.3 Complete reference monitor, requires that there be distinct domain(s) for TSP enforcement, a domain for the remainder of the TSF, as well as domains for the non-TSF portions of the TOE.

### Management:FPT_SEP.1, FPT_SEP.2, FPT_SEP.3

432 There are no management activities foreseen.

Audit: FPT_SEP.1, FPT_SEP.2, FPT_SEP.3

433 There are no auditable events foreseen.

### FPT_SEP.1 TSF domain separation

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_SEP.1.1 **The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.**

FPT_SEP.1.2 **The TSF shall enforce separation between the security domains of subjects in the TSC.**

### FPT_SEP.2 SFP domain separation

Hierarchical to: FPT_SEP.1 TSF domain separation

Dependencies: No dependencies.

FPT_SEP.2.1 The **unisolated portion of the** TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.2.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

FPT_SEP.2.3 **The TSF shall maintain the part of the TSF related to [assignment: *list of access control and/or information flow control SFPs*] in a security domain for their own execution that protects them from interference and tampering by the remainder of the TSF and by subjects untrusted with respect to those SFPs.**

### FPT_SEP.3 Complete reference monitor

Hierarchical to: FPT_SEP.2 SFP domain separation

Dependencies: No dependencies.

FPT_SEP.3.1 The unisolated portion of the TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.3.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

FPT_SEP.3.3 The TSF shall maintain the part of the TSF **that enforces the** access control and/or information flow control **SFPs** in a security domain for **its** own execution that protects them from interference and tampering by the remainder of the TSF and by subjects untrusted with respect to **the TSP,**

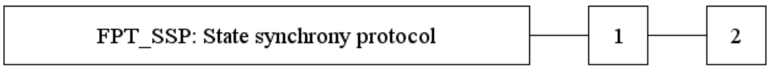
## 15.12 State synchrony protocol (FPT_SSP)

### Family Behaviour

434 Distributed systems may give rise to greater complexity than monolithic systems through the potential for differences in state between parts of the system, and through delays in communication. In most cases synchronisation of state between distributed functions involves an exchange protocol, not a simple action. When malice exists in the distributed environment of these protocols, more complex defensive protocols are required.

435 State synchrony protocol (FPT_SSP) establishes the requirement for certain critical security functions of the TSF to use this trusted protocol. State synchrony protocol (FPT_SSP) ensures that two distributed parts of the TOE (e.g. hosts) have synchronised their states after a security-relevant action.

### Component levelling




436 FPT_SSP.1 Simple trusted acknowledgement, requires only a simple acknowledgment by the data recipient.

437 FPT_SSP.2 Mutual trusted acknowledgement, requires mutual acknowledgment of the data exchange.

### Management:FPT_SSP.1, FPT_SSP.2

438 There are no management activities foreseen.

### Audit: FPT_SSP.1, FPT_SSP.2

439 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: failure to receive an acknowledgement when expected.

## FPT_SSP.1 Simple trusted acknowledgement

Hierarchical to: No other components.

Dependencies: FPT_ITT.1 Basic internal TSF data transfer protection

FPT_SSP.1.1 **The TSF shall acknowledge, when requested by another part of the TSF, the receipt of an unmodified TSF data transmission.**

## FPT_SSP.2 Mutual trusted acknowledgement

Hierarchical to: FPT_SSP.1 Simple trusted acknowledgement

Dependencies: FPT_ITT.1 Basic internal TSF data transfer protection

**FPT_SSP.2.1** The TSF shall acknowledge, when requested by another part of the TSF, the receipt of an unmodified TSF data transmission.

**FPT_SSP.2.2** **The TSF shall ensure that the relevant parts of the TSF know the correct status of transmitted data among its different parts, using acknowledgements.**

## 15.13 Time stamps (FPT_STM)

### Family Behaviour

440 This family addresses requirements for a reliable time stamp function within a TOE.

### Component levelling


441 This family consists of only one component, FPT_STM.1 Reliable time stamps, which requires that the TSF provide reliable time stamps for TSF functions.

### Management: FPT_STM.1

442 The following actions could be considered for the management functions in FMT:

a) management of the time.

### Audit: FPT_STM.1

443 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: changes to the time;

b) Detailed: providing a timestamp.

### FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_STM.1.1 **The TSF shall be able to provide reliable time stamps for its own use.**

## 15.14 Inter-TSF TSF data consistency (FPT_TDC)

### Family Behaviour

444 In a distributed or composite system environment, a TOE may need to exchange TSF data (e.g. the SFP-attributes associated with data, audit information, identification information) with another trusted IT product, This family defines the requirements for sharing and consistent interpretation of these attributes between the TSF of the TOE and a different trusted IT product.

### Component levelling


445 FPT_TDC.1 Inter-TSF basic TSF data consistency, requires that the TSF provide the capability to ensure consistency of attributes between TSFs.

### Management:FPT_TDC.1

446 There are no management activities foreseen.

### Audit: FPT_TDC.1

447 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Successful use of TSF data consistency mechanisms.

b) Basic: Use of the TSF data consistency mechanisms.

c) Basic: Identification of which TSF data have been interpreted.

d) Basic: Detection of modified TSF data.

### FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TDC.1.1 **The TSF shall provide the capability to consistently interpret [assignment: *list of TSF data types*] when shared between the TSF and another trusted IT product.**

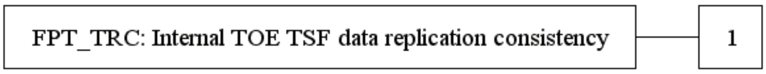
FPT_TDC.1.2 **The TSF shall use [assignment: *list of interpretation rules to be applied by the TSF*] when interpreting the TSF data from another trusted IT product.**

## 15.15 Internal TOE TSF data replication consistency (FPT_TRC)

### Family Behaviour

448 The requirements of this family are needed to ensure the consistency of TSF data when such data is replicated internal to the TOE. Such data may become inconsistent if the internal channel between parts of the TOE becomes inoperative. If the TOE is internally structured as a network and parts of the TOE network connections are broken, this may occur when parts become disabled.

### Component levelling




449 This family consists of only one component, FPT_TRC.1 Internal TSF consistency, which requires that the TSF ensure the consistency of TSF data that is replicated in multiple locations.

### Management:FPT_TRC.1

450 There are no management activities foreseen.

### Audit: FPT_TRC.1

451 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: restoring consistency upon reconnection.

b) Basic: Detected inconsistency between TSF data.

### FPT_TRC.1 Internal TSF consistency

Hierarchical to: No other components.

Dependencies: FPT_ITT.1 Basic internal TSF data transfer protection

FPT_TRC.1.1 **The TSF shall ensure that TSF data is consistent when replicated between parts of the TOE.**

FPT_TRC.1.2 **When parts of the TOE containing replicated TSF data are disconnected, the TSF shall ensure the consistency of the replicated TSF data upon reconnection before processing any requests for [assignment: *list of SFs dependent on TSF data replication consistency*].**

## 15.16 TSF self test (FPT_TST)

### Family Behaviour

452 The family defines the requirements for the self-testing of the TSF with respect to some expected correct operation. Examples are interfaces to enforcement functions, and sample arithmetical operations on critical parts of the TOE. These tests can be carried out at start-up, periodically, at the request of the authorised user, or when other conditions are met. The actions to be taken by the TOE as the result of self testing are defined in other families.

453 The requirements of this family are also needed to detect the corruption of TSF executable code (i.e. TSF software) and TSF data by various failures that do not necessarily stop the TOE's operation (which would be handled by other families). These checks must be performed because these failures may not necessarily be prevented. Such failures can occur either because of unforeseen failure modes or associated oversights in the design of hardware, firmware, or software, or because of malicious corruption of the TSF due to inadequate logical and/or physical protection.

### Component levelling


454 FPT_TST.1 TSF testing, provides the ability to test the TSF's correct operation. These tests may be performed at start-up, periodically, at the request of the authorised user, or when other conditions are met. It also provides the ability to verify the integrity of TSF data and executable code.

### Management: FPT_TST.1

455 The following actions could be considered for the management functions in FMT:

a) management of the conditions under which TSF self testing occurs, such as during initial start-up, regular interval, or under specified conditions;

b) management of the time interval if appropriate.

### Audit: FPT_TST.1

456 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Basic: Execution of the TSF self tests and the results of the tests.

### FPT_TST.1 TSF testing

Hierarchical to: No other components.

Dependencies: FPT_AMT.1 Abstract machine testing

FPT_TST.1.1 **The TSF shall run a suite of self tests [selection: *during initial start-up, periodically during normal operation, at the request of the authorised user, at the conditions[assignment: conditions under which self test should occur]*] to demonstrate the correct operation of the TSF. operation of [selection: *[assignment: parts of TSF], the TSF*].**

FPT_TST.1.2 **The TSF shall provide authorised users with the capability to verify the integrity of [selection: *[assignment: parts of TSF], TSF data*].**

FPT_TST.1.3 **The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.**

# 16 Class FRU: Resource utilisation

457 This class provides three families that support the availability of required resources such as processing capability and/or storage capacity. The family Fault Tolerance provides protection against unavailability of capabilities caused by failure of the TOE. The family Priority of Service ensures that the resources will be allocated to the more important or time-critical tasks and cannot be monopolised by lower priority tasks. The family Resource Allocation provides limits on the use of available resources, therefore preventing users from monopolising the resources.

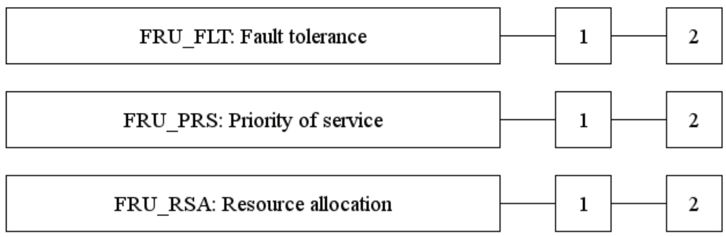


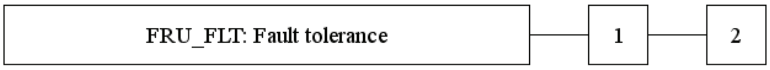

**Figure 17 - FRU: Resource utilisation class decomposition**

## 16.1 Fault tolerance (FRU_FLT)

### Family Behaviour

458 The requirements of this family ensure that the TOE will maintain correct operation even in the event of failures.

### Component levelling




459 FRU_FLT.1 Degraded fault tolerance, requires the TOE to continue correct operation of identified capabilities in the event of identified failures.

460 FRU_FLT.2 Limited fault tolerance, requires the TOE to continue correct operation of all capabilities in the event of identified failures.

### Management:FRU_FLT.1, FRU_FLT.2

461 There are no management activities foreseen.

### Audit: FRU_FLT.1

462 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Any failure detected by the TSF.

b) Basic: All TOE capabilities being discontinued due to a failure.

### Audit: FRU_FLT.2

463 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Any failure detected by the TSF.

## FRU_FLT.1 Degraded fault tolerance

Hierarchical to: No other components.

Dependencies: FPT_FLS.1 Failure with preservation of secure state

FRU_FLT.1.1 **The TSF shall ensure the operation of [assignment: *list of TOE capabilities*] when the following failures occur: [assignment: *list of type of failures*].**

## FRU_FLT.2 Limited fault tolerance

Hierarchical to: FRU_FLT.1 Degraded fault tolerance

Dependencies: FPT_FLS.1 Failure with preservation of secure state

**FRU_FLT.2.1** The TSF shall ensure the operation of **all the TOE's capabilities** when the following failures occur: [assignment: *list of type of failures*].

## 16.2 Priority of service (FRU_PRS)

### Family Behaviour

464 The requirements of this family allow the TSF to control the use of resources within the TSC by users and subjects such that high priority activities within the TSC will always be accomplished without undue interference or delay caused by low priority activities.

### Component levelling


465 FRU_PRS.1 Limited priority of service, provides priorities for a subject's use of a subset of the resources within the TSC.

466 FRU_PRS.2 Full priority of service, provides priorities for a subject's use of all of the resources within the TSC.

### Management: FRU_PRS.1, FRU_PRS.2

467 The following actions could be considered for the management functions in FMT:

a) assignment of priorities to each subject in the TSF.

### Audit: FRU_PRS.1, FRU_PRS.2

468 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Rejection of operation based on the use of priority within an allocation.

b) Basic: All attempted uses of the allocation function which involves the priority of the service functions.

### FRU_PRS.1 Limited priority of service

Hierarchical to: No other components.

Dependencies: No dependencies.

FRU_PRS.1.1 **The TSF shall assign a priority to each subject in the TSF.**

FRU_PRS.1.2 **The TSF shall ensure that each access to [assignment: *controlled resources*] shall be mediated on the basis of the subjects assigned priority.**

## FRU_PRS.2 Full priority of service

Hierarchical to: FRU_PRS.1 Limited priority of service

Dependencies: No dependencies.

**FRU_PRS.2.1** The TSF shall assign a priority to each subject in the TSF.

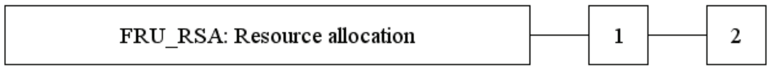
**FRU_PRS.2.2** The TSF shall ensure that each access to **all shareable resources** shall be mediated on the basis of the subjects assigned priority.

## 16.3 Resource allocation (FRU_RSA)

### Family Behaviour

469 The requirements of this family allow the TSF to control the use of resources by users and subjects such that denial of service will not occur because of unauthorised monopolisation of resources.

### Component levelling




470 FRU_RSA.1 Maximum quotas, provides requirements for quota mechanisms that ensure that users and subjects will not monopolise a controlled resource.

471 FRU_RSA.2 Minimum and maximum quotas, provides requirements for quota mechanisms that ensure that users and subjects will always have at least a minimum of a specified resource and that they will not be able to monopolise a controlled resource.

### Management: FRU_RSA.1

472 The following actions could be considered for the management functions in FMT:

a) specifying maximum limits for a resource for groups and/or individual users and/or subjects by an administrator.

### Management: FRU_RSA.2

473 The following actions could be considered for the management functions in FMT:

a) specifying minimum and maximum limits for a resource for groups and/or individual users and/or subjects by an administrator.

### Audit: FRU_RSA.1, FRU_RSA.2

474 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Rejection of allocation operation due to resource limits.

b) Basic: All attempted uses of the resource allocation functions for resources that are under control of the TSF.

## FRU_RSA.1 Maximum quotas

Hierarchical to: No other components.

Dependencies: No dependencies.

FRU_RSA.1.1 **The TSF shall enforce maximum quotas of the following resources: [assignment: *controlled resources*] that [selection: *individual user, defined group of users, subjects*] can use [selection: *simultaneously, over a specified period of time*].**

## FRU_RSA.2 Minimum and maximum quotas

Hierarchical to: FRU_RSA.1 Maximum quotas

Dependencies: No dependencies.

FRU_RSA.2.1 The TSF shall enforce maximum quotas of the following resources [assignment: *controlled resources*] that [selection: *individual user, defined group of* ***users*****]** can use [selection: *simultaneously, over a specified period of time*].

FRU_RSA.2.2 **The TSF shall ensure the provision of minimum quantity of each [assignment: *controlled resource*] that is available for [selection: *an individual user, defined group of users, subjects*] to use [selection: *simultaneously, over a specified period of time*].**

# 17 Class FTA: TOE access

475 This family specifies functional requirements for controlling the establishment of a user's session.

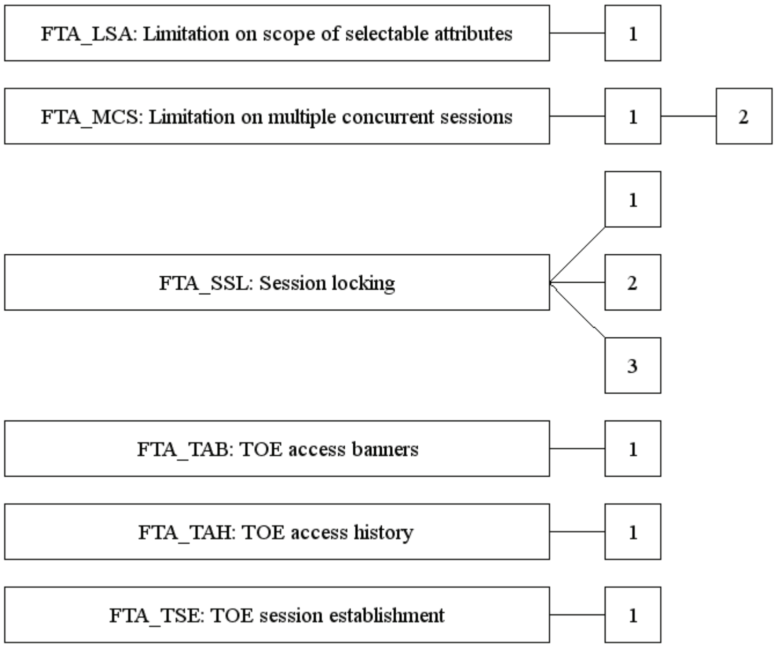


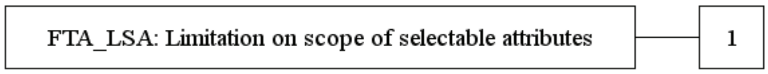

**Figure 18 - FTA: TOE access class decomposition**

## 17.1 Limitation on scope of selectable attributes (FTA_LSA)

### Family Behaviour

476 This family defines requirements to limit the scope of session security attributes that a user may select for a session.

### Component levelling




477 FTA_LSA.1 Limitation on scope of selectable attributes, provides the requirement for a TOE to limit the scope of the session security attributes during session establishment.

### Management: FTA_LSA.1

478 The following actions could be considered for the management functions in FMT:

a) management of the scope of the session security attributes by an administrator.

### Audit: FTA_LSA.1

479 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All failed attempts at selecting a session security attributes;

b) Basic: All attempts at selecting a session security attributes;

c) Detailed: Capture of the values of each session security attributes.

### FTA_LSA.1 Limitation on scope of selectable attributes

Hierarchical to: No other components.

Dependencies: No dependencies.

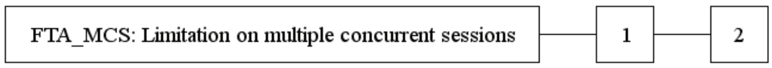
FTA_LSA.1.1 **The TSF shall restrict the scope of the session security attributes [assignment: *session security attributes*], based on [assignment: *attributes*].**

## 17.2 Limitation on multiple concurrent sessions (FTA_MCS)

### Family Behaviour

480 This family defines requirements to place limits on the number of concurrent sessions that belong to the same user.

### Component levelling




481 FTA_MCS.1 Basic limitation on multiple concurrent sessions, provides limitations that apply to all users of the TSF.

482 FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions extends FTA_MCS.1 Basic limitation on multiple concurrent sessions by requiring the ability to specify limitations on the number of concurrent sessions based on the related security attributes.

### Management: FTA_MCS.1

483 The following actions could be considered for the management functions in FMT:

a) management of the maximum allowed number of concurrent user sessions by an administrator.

### Management: FTA_MCS.2

484 The following actions could be considered for the management functions in FMT:

a) management of the rules that govern the maximum allowed number of concurrent user sessions by an administrator.

### Audit: FTA_MCS.1, FTA_MCS.2

485 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Rejection of a new session based on the limitation of multiple concurrent sessions.

b) Detailed: Capture of the number of currently concurrent user sessions and the user security attribute(s).

## FTA_MCS.1 Basic limitation on multiple concurrent sessions

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

FTA_MCS.1.1 **The TSF shall restrict the maximum number of concurrent sessions that belong to the same user.**

FTA_MCS.1.2 **The TSF shall enforce, by default, a limit of [assignment: *default number*] sessions per user.**

## FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions

Hierarchical to: FTA_MCS.1 Basic limitation on multiple concurrent sessions

Dependencies: FIA_UID.1 Timing of identification

FTA_MCS.2.1 The TSF shall restrict the maximum number of concurrent sessions that belong to the same user **according to the rules [assignment: *rules for the number of maximum concurrent sessions*].**

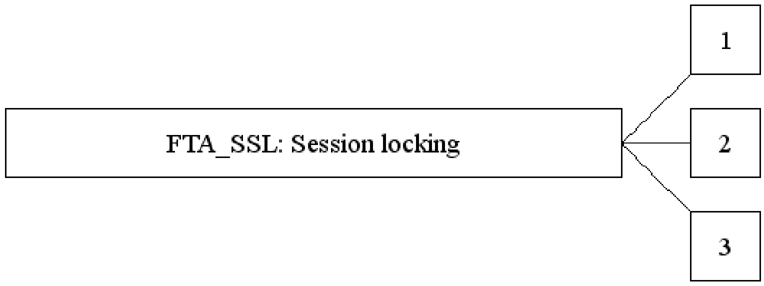
FTA_MCS.2.2 The TSF shall enforce, by default, a limit of [assignment: *default number*] sessions per user.

## 17.3 Session locking (FTA_SSL)

### Family Behaviour

486 This family defines requirements for the TSF to provide the capability for TSF-initiated and user-initiated locking and unlocking of interactive sessions.

### Component levelling




487 FTA_SSL.1 TSF-initiated session locking includes system initiated locking of an interactive session after a specified period of user inactivity.

488 FTA_SSL.2 User-initiated locking, provides capabilities for the user to lock and unlock the user's own interactive sessions.

489 FTA_SSL.3 TSF-initiated termination, provides requirements for the TSF to terminate the session after a period of user inactivity.

### Management: FTA_SSL.1

490 The following actions could be considered for the management functions in FMT:

a) specification of the time of user inactivity after which lock-out occurs for an individual user;

b) specification of the default time of user inactivity after which lock-out occurs;

c) management of the events that should occur prior to unlocking the session.

### Management: FTA_SSL.2

491 The following actions could be considered for the management functions in FMT:

a) management of the events that should occur prior to unlocking the session.

### Management: FTA_SSL.3

492 The following actions could be considered for the management functions in FMT:

a) specification of the time of user inactivity after which termination of the interactive session occurs for an individual user;

b) specification of the default time of user inactivity after which termination of the interactive session occurs.

### Audit: FTA_SSL.1, FTA_SSL.2

493 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Locking of an interactive session by the session locking mechanism.

b) Minimal: Successful unlocking of an interactive session.

c) Basic: Any attempts at unlocking an interactive session.

### Audit: FTA_SSL.3

494 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Termination of an interactive session by the session locking mechanism.

## FTA_SSL.1 TSF-initiated session locking

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL.1.1 **The TSF shall lock an interactive session after [assignment: *time interval of user inactivity*] by:**

**a) clearing or overwriting display devices, making the current contents unreadable;**

**b) disabling any activity of the user's data access/display devices other than unlocking the session.**

FTA_SSL.1.2 **The TSF shall require the following events to occur prior to unlocking the session: [assignment: *events to occur*].**

## FTA_SSL.2 User-initiated locking

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL.2.1 **The TSF shall allow user-initiated locking of the user's own interactive session, by:**

**a) clearing or overwriting display devices, making the current contents unreadable;**

**b) disabling any activity of the user's data access/display devices other than unlocking the session.**

FTA_SSL.2.2 **The TSF shall require the following events to occur prior to unlocking the session: [assignment: *events to occur*].**

## FTA_SSL.3 TSF-initiated termination

Hierarchical to: No other components.

Dependencies: No dependencies.

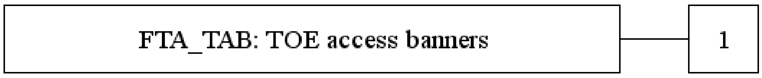
FTA_SSL.3.1 **The TSF shall terminate an interactive session after a [assignment: *time interval of user inactivity*].**

## 17.4 TOE access banners (FTA_TAB)

### Family Behaviour

495 This family defines requirements to display a configurable advisory warning message to users regarding the appropriate use of the TOE.

### Component levelling




496 FTA_TAB.1 Default TOE access banners, provides the requirement for a TOE Access Banner. This banner is displayed prior to the establishment dialogue for a session.

### Management: FTA_TAB.1

497 The following actions could be considered for the management functions in FMT:

a) maintenance of the banner by the authorised administrator.

### Audit: FTA_TAB.1

498 There are no auditable events foreseen.

## FTA_TAB.1 Default TOE access banners

Hierarchical to: No other components.

Dependencies: No dependencies.

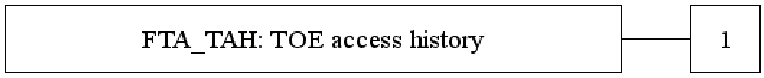
FTA_TAB.1.1 **Before establishing a user session, the TSF shall display an advisory warning message regarding unauthorised use of the TOE.**

## 17.5 TOE access history (FTA_TAH)

### Family Behaviour

499 This family defines requirements for the TSF to display to a user, upon successful session establishment, a history of successful and unsuccessful attempts to access the user's account.

### Component levelling




500 FTA_TAH.1 TOE access history, provides the requirement for a TOE to display information related to previous attempts to establish a session.

### Management:FTA_TAH.1

501 There are no management activities foreseen.

### Audit: FTA_TAH.1

502 There are no auditable events foreseen.

### FTA_TAH.1 TOE access history

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_TAH.1.1 **Upon successful session establishment, the TSF shall display the [selection: *date, time, method, location*] of the last successful session establishment to the user.**

FTA_TAH.1.2 **Upon successful session establishment, the TSF shall display the [selection: *date, time, method, location*] of the last unsuccessful attempt to session establishment and the number of unsuccessful attempts since the last successful session establishment.**

FTA_TAH.1.3 **The TSF shall not erase the access history information from the user interface without giving the user an opportunity to review the information.**

## 17.6 TOE session establishment (FTA_TSE)

### Family Behaviour

503 This family defines requirements to deny a user permission to establish a session with the TOE.

### Component levelling


504 FTA_TSE.1 TOE session establishment, provides requirements for denying users access to the TOE based on attributes.

### Management: FTA_TSE.1

505 The following actions could be considered for the management functions in FMT:

a) management of the session establishment conditions by the authorised administrator.

### Audit: FTA_TSE.1

506 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Denial of a session establishment due to the session establishment mechanism.

b) Basic: All attempts at establishment of a user session.

c) Detailed: Capture of the value of the selected access parameters (e.g. location of access, time of access).

### FTA_TSE.1 TOE session establishment

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_TSE.1.1 **The TSF shall be able to deny session establishment based on [assignment: *attributes*].**

# 18 Class FTP: Trusted path/channels

507 Families in this class provide requirements for a trusted communication path between users and the TSF, and for a trusted communication channel between the TSF and other trusted IT products. Trusted paths and channels have the following general characteristics:

- The communications path is constructed using internal and external communications channels (as appropriate for the component) that isolate an identified subset of TSF data and commands from the remainder of the TSF and user data.

- Use of the communications path may be initiated by the user and/or the TSF (as appropriate for the component).

- The communications path is capable of providing assurance that the user is communicating with the correct TSF, and that the TSF is communicating with the correct user (as appropriate for the component).

508 In this paradigm, a trusted channel is a communication channel that may be initiated by either side of the channel, and provides non-repudiation characteristics with respect to the identity of the sides of the channel.

509 A trusted path provides a means for users to perform functions through an assured direct interaction with the TSF. Trusted path is usually desired for user actions such as initial identification and/or authentication, but may also be desired at other times during a user's session. Trusted path exchanges may be initiated by a user or the TSF. User responses via the trusted path are guaranteed to be protected from modification by or disclosure to untrusted applications.

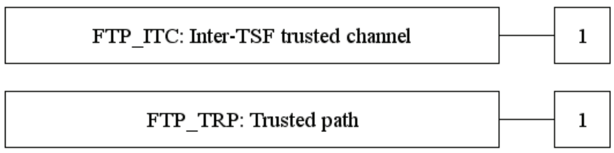



**Figure 19 - FTP: Trusted path/channels class decomposition**

## 18.1 Inter-TSF trusted channel (FTP_ITC)

### Family Behaviour

510 This family defines requirements for the creation of a trusted channel between the TSF and other trusted IT products for the performance of security critical operations. This family should be included whenever there are requirements for the secure communication of user or TSF data between the TOE and other trusted IT products.

### Component levelling


511 FTP_ITC.1 Inter-TSF trusted channel, requires that the TSF provide a trusted communication channel between itself and another trusted IT product.

### Management: FTP_ITC.1

512 The following actions could be considered for the management functions in FMT:

a) Configuring the actions that require trusted channel, if supported.

### Audit: FTP_ITC.1

513 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Failure of the trusted channel functions.

b) Minimal: Identification of the initiator and target of failed trusted channel functions.

c) Basic: All attempted uses of the trusted channel functions.

d) Basic: Identification of the initiator and target of all trusted channel functions.

### FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1 **The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.**

FTP_ITC.1.2 **The TSF shall permit [selection: *the TSF, the remote trusted IT product*] to initiate communication via the trusted channel.**

FTP_ITC.1.3 **The TSF shall initiate communication via the trusted channel for [assignment: *list of functions for which a trusted channel is required*].**

## 18.2 Trusted path (FTP_TRP)

### Family Behaviour

514 This family defines the requirements to establish and maintain trusted communication to or from users and the TSF. A trusted path may be required for any security-relevant interaction. Trusted path exchanges may be initiated by a user during an interaction with the TSF, or the TSF may establish communication with the user via a trusted path.

### Component levelling


515 FTP_TRP.1 Trusted path, requires that a trusted path between the TSF and a user be provided for a set of events defined by a PP/ST author. The user and/or the TSF may have the ability to initiate the trusted path.

### Management: FTP_TRP.1

516 The following actions could be considered for the management functions in FMT:

a) Configuring the actions that require trusted path, if supported.

### Audit: FTP_TRP.1

517 The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Failures of the trusted path functions.

b) Minimal: Identification of the user associated with all trusted path failures, if available.

c) Basic: All attempted uses of the trusted path functions.

d) Basic: Identification of the user associated with all trusted path invocations, if available.

### FTP_TRP.1 Trusted path

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1 **The TSF shall provide a communication path between itself and [selection: *remote, local*] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.**

FTP_TRP.1.2 **The TSF shall permit [selection: *the TSF, local users, remote users*] to initiate communication via the trusted path.**

FTP_TRP.1.3 **The TSF shall require the use of the trusted path for [selection: *initial user authentication,[assignment: other services for which trusted path is required]*].**

# A Security functional requirements application notes (normative)

518 This annex contains additional guidance for the families and components defined in the elements of this CC Part 2, which may be required by users, developers or evaluators to use the components. To facilitate finding the appropriate information, the presentation of the classes, families and components in this annex is similar to the presentation within the elements.

## A.1 Structure of the notes

519 This chapter defines the content and presentation of the notes related to functional requirements of the CC.

### A.1.1 Class structure

520 Figure 20 below illustrates the functional class structure in this annex.

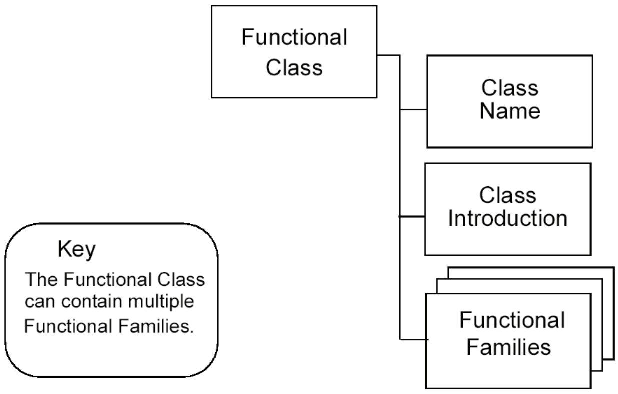



**Figure 20 - Functional class structure**

#### A.1.1.1 Class name

521 This is the unique name of the class defined within the normative elements of this part of the CC.

#### A.1.1.2 Class introduction

522 The class introduction in this annex provides information about the use of the families and components of the class. This information is completed with the informative diagram that describes the organisation of each class with the families in each class and the hierarchical relationship between components in each family.

## A.1.2 Family structure

523 Figure 21 illustrates the functional family structure for application notes in diagrammatic form.

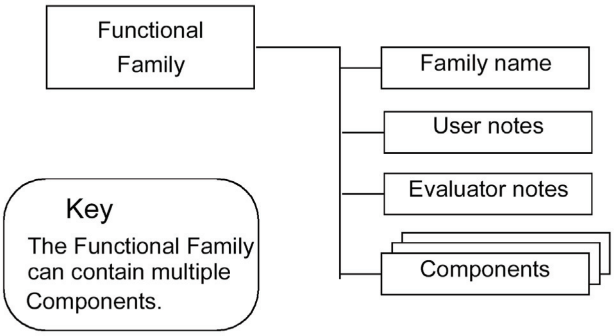



**Figure 21 - Functional family structure for application notes**

### A.1.2.1 Family name

524 This is the unique name of the family defined within the normative elements of this part of the CC.

### A.1.2.2 User notes

525 The user notes contain additional information that is of interest to potential users of the family, that is PP, ST and functional package authors, and developers of TOEs incorporating the functional components. The presentation is informative, and might cover warnings about limitations of use and areas where specific attention might be required when using the components.

### A.1.2.3 Evaluator notes

526 The evaluator notes contain any information that is of interest to developers and evaluators of TOEs that claim compliance with a component of the family. The presentation is informative and can cover a variety of areas where specific attention might be needed when evaluating the TOE. This can include clarifications of meaning and specification of the way to interpret requirements, as well as caveats and warnings of specific interest to evaluators.

527 These User Notes and Evaluator Notes sections are not mandatory and appear only if appropriate.

## A.1.3 Component structure

528 Figure 22 illustrates the functional component structure for the application notes.

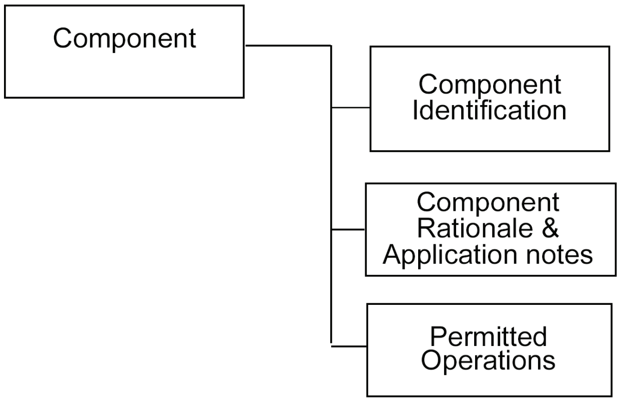



**Figure 22 - Functional component structure**

### A.1.3.1 Component identification

529 This is the unique name of the component defined within the normative elements of this part of the CC.

### A.1.3.2 Component rationale and application notes

530 Any specific information related to the component can be found in this section.

- The *rationale* contains the specifics of the rationale that refine the general statements on rationale for the specific level, and should only be used if level specific amplification is required.

- The *application notes* contain additional refinement in terms of narrative qualification as it pertains to a specific component. This refinement can pertain to user notes, and/or evaluator notes as described in Section A.1.2. This refinement can be used to explain the nature of the dependencies (e.g. shared information, or shared operation).

531 This section is not mandatory and appears only if appropriate.

### A.1.3.3 Permitted operations

532 This portion of each component contains advice relating to the permitted operations of the component.

533 This section is not mandatory and appears only if appropriate.

## A.2 Dependency tables

534 The following dependency tables for functional components show their direct, indirect and optional dependencies. Each of the components that is a dependency of some functional component is allocated a column. Each functional component is allocated a row. The value in the table cell indicate whether the column label component is directly required (indicated by a

cross "X"), indirectly required (indicated by a dash "-"), or optionally required (indicated by a "o") by the row label component. An example of a component with optional dependencies is FDP_ETC.1 Export of user data without security attributes, which requires either FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control to be present. So if FDP_ACC.1 Subset access control is present, FDP_IFC.1 Subset information flow control is not necessary and vice versa. If no character is presented, the component is not dependent upon another component.

| | FAU_GEN.1 | FAU_SAA.1 | FAU_SAR.1 | FAU_STG.1 | FIA_UID.1 | FMT_MTD.1 | FMT_SMF.1 | FMT_SMR.1 | FPT_STM.1 |
|---|---|---|---|---|---|---|---|---|---|
| FAU_ARP.1 | - | X | | | | | | | - |
| FAU_GEN.1 | | | | | | | | | X |
| FAU_GEN.2 | X | | | | X | | | | - |
| FAU_SAA.1 | X | | | | | | | | - |
| FAU_SAA.2 | | | | | X | | | | |
| FAU_SAA.3 | | | | | | | | | |
| FAU_SAA.4 | | | | | | | | | |
| FAU_SAR.1 | X | | | | | | | | - |
| FAU_SAR.2 | - | | X | | | | | | - |
| FAU_SAR.3 | - | | X | | | | | | - |
| FAU_SEL.1 | X | | | | - | X | - | - | - |
| FAU_STG.1 | X | | | | | | | | - |
| FAU_STG.2 | X | | | | | | | | - |
| FAU_STG.3 | - | | | X | | | | | - |
| FAU_STG.4 | - | | | X | | | | | - |

**Table 1 - Dependency table for Class FAU: Security audit**

| | FIA_UID.1 |
|---|---|
| FCO_NRO.1 | X |
| FCO_NRO.2 | X |
| FCO_NRR.1 | X |
| FCO_NRR.2 | X |

**Table 2 - Dependency table for Class FCO: Communication**

| | ADV_SPM.1 | FCS_CKM.1 | FCS_CKM.2 | FCS_CKM.4 | FCS_COP.1 | FDP_ACC.1 | FDP_ACF.1 | FDP_IFC.1 | FDP_IFF.1 | FDP_ITC.1 | FDP_ITC.2 | FIA_UID.1 | FMT_MSA.1 | FMT_MSA.2 | FMT_MSA.3 | FMT_SMF.1 | FMT_SMR.1 | FPT_TDC.1 | FTP_ITC.1 | FTP_TRP.1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FCS_CKM.1 | - | - | O | X | O | - | - | - | - | - | - | - | - | X | - | - | - | - | - | - |
| FCS_CKM.2 | - | O | - | X | - | - | - | - | - | O | O | - | - | X | - | - | - | - | - | - |
| FCS_CKM.3 | - | O | - | X | - | - | - | - | - | O | O | - | - | X | - | - | - | - | - | - |
| FCS_CKM.4 | - | O | - | - | - | - | - | - | - | O | O | - | - | X | - | - | - | - | - | - |
| FCS_COP.1 | - | O | - | X | - | - | - | - | - | O | O | - | - | X | - | - | - | - | - | - |

**Table 3 - Dependency table for Class FCS: Cryptographic support**

| | AVA_CCA.1 | AVA_CCA.3 | FDP_ACC.1 | FDP_ACF.1 | FDP_IFC.1 | FDP_IFF.1 | FDP_ITT.1 | FDP_ITT.2 | FDP_UIT.1 | FIA_UID.1 | FMT_MSA.1 | FMT_MSA.3 | FMT_SMF.1 | FMT_SMR.1 | FPT_TDC.1 | FTP_ITC.1 | FTP_TRP.1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FDP_ACC.1 | | | - | X | - | - | | | | - | - | - | - | - | | | |
| FDP_ACC.2 | | | - | X | - | - | | | | - | - | - | - | - | | | |
| FDP_ACF.1 | | | X | - | - | - | | | | - | - | X | - | - | | | |
| FDP_DAU.1 | | | | | | | | | | | | | | | | | |
| FDP_DAU.2 | | | | | | | | | | X | | | | | | | |
| FDP_ETC.1 | | | O | - | O | - | | | | - | - | - | - | - | | | |
| FDP_ETC.2 | | | O | - | O | - | | | | - | - | - | - | - | | | |
| FDP_IFC.1 | | | - | - | - | X | | | | - | - | - | - | - | | | |
| FDP_IFC.2 | | | - | - | - | X | | | | - | - | - | - | - | | | |
| FDP_IFF.1 | | | - | - | X | - | | | | - | - | X | - | - | | | |
| FDP_IFF.2 | | | - | - | X | - | | | | - | - | X | - | - | | | |
| FDP_IFF.3 | X | | - | - | X | - | | | | - | - | - | - | - | | | |
| FDP_IFF.4 | X | | - | - | X | - | | | | - | - | - | - | - | | | |
| FDP_IFF.5 | | X | - | - | X | - | | | | - | - | - | - | - | | | |
| FDP_IFF.6 | X | | - | - | X | - | | | | - | - | - | - | - | | | |
| FDP_ITC.1 | | | O | - | O | - | | | | - | - | X | - | - | | | |
| FDP_ITC.2 | | | O | - | O | - | | | | - | - | - | - | - | X | O | O |
| FDP_ITT.1 | | | O | - | O | - | | | | - | - | - | - | - | | | |
| FDP_ITT.2 | | | O | - | O | - | | | | - | - | - | - | - | | | |
| FDP_ITT.3 | | | O | - | O | - | X | | | - | - | - | - | - | | | |
| FDP_ITT.4 | | | O | - | O | - | | X | | - | - | - | - | - | | | |
| FDP_RIP.1 | | | | | | | | | | | | | | | | | |
| FDP_RIP.2 | | | | | | | | | | | | | | | | | |
| FDP_ROL.1 | | | O | - | O | - | | | | - | - | - | - | - | | | |
| FDP_ROL.2 | | | O | - | O | - | | | | - | - | - | - | - | | | |
| FDP_SDI.1 | | | | | | | | | | | | | | | | | |
| FDP_SDI.2 | | | | | | | | | | | | | | | | | |
| FDP_UCT.1 | | | O | - | O | - | | | | - | - | - | - | - | | O | O |
| FDP_UIT.1 | | | O | - | O | - | | | | - | - | - | - | - | | O | O |
| FDP_UIT.2 | | | O | - | O | - | | | O | - | - | - | - | - | | O | - |
| FDP_UIT.3 | | | O | - | O | - | | | O | - | - | - | - | - | | O | - |

**Table 4 - Dependency table for Class FDP: User data protection**

| | FIA_ATD.1 | FIA_UAU.1 | FIA_UID.1 |
|---|---|---|---|
| FIA_AFL.1 | | X | - |
| FIA_ATD.1 | | | |
| FIA_SOS.1 | | | |
| FIA_SOS.2 | | | |
| FIA_UAU.1 | | | X |
| FIA_UAU.2 | | | X |
| FIA_UAU.3 | | | |
| FIA_UAU.4 | | | |
| FIA_UAU.5 | | | |
| FIA_UAU.6 | | | |
| FIA_UAU.7 | | X | - |
| FIA_UID.1 | | | |
| FIA_UID.2 | | | |
| FIA_USB.1 | X | | |

**Table 5 - Dependency table for Class FIA: Identification and authentication**

| | ADV_SPM.1 | FDP_ACC.1 | FDP_ACF.1 | FDP_IFC.1 | FDP_IFF.1 | FIA_UID.1 | FMT_MSA.1 | FMT_MSA.3 | FMT_MTD.1 | FMT_SMF.1 | FMT_SMR.1 | FPT_STM.1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FMT_MOF.1 | | | | | | - | | | | X | X | |
| FMT_MSA.1 | | O | - | O | - | - | - | - | | X | X | |
| FMT_MSA.2 | X | O | - | O | - | - | X | - | | - | X | |
| FMT_MSA.3 | | - | - | - | - | - | X | - | | - | X | |
| FMT_MTD.1 | | | | | | - | | | | X | X | |
| FMT_MTD.2 | | | | | | - | | | X | - | X | |
| FMT_MTD.3 | X | | | | | - | | | X | - | - | |
| FMT_REV.1 | | | | | | - | | | | | X | |
| FMT_SAE.1 | | | | | | - | | | | | X | X |
| FMT_SMF.1 | | | | | | | | | | | | |
| FMT_SMR.1 | | | | | | X | | | | | | |
| FMT_SMR.2 | | | | | | X | | | | | | |
| FMT_SMR.3 | | | | | | - | | | | | X | |

**Table 6 - Dependency table for Class FMT: Security management**

| | FIA_UID.1 | FPR_UNO.1 |
|---|---|---|
| FPR_ANO.1 | | |
| FPR_ANO.2 | | |
| FPR_PSE.1 | | |
| FPR_PSE.2 | X | |
| FPR_PSE.3 | | |
| FPR_UNL.1 | | |
| FPR_UNO.1 | | |
| FPR_UNO.2 | | |
| FPR_UNO.3 | | X |
| FPR_UNO.4 | | |

**Table 7 - Dependency table for Class FPR: Privacy**

| | ADV_SPM.1 | AGD_ADM.1 | FIA_UID.1 | FMT_MOF.1 | FMT_SMF.1 | FMT_SMR.1 | FPT_AMT.1 | FPT_ITT.1 |
|---|---|---|---|---|---|---|---|---|
| FPT_AMT.1 | | | | | | | | |
| FPT_FLS.1 | X | | | | | | | |
| FPT_ITA.1 | | | | | | | | |
| FPT_ITC.1 | | | | | | | | |
| FPT_ITI.1 | | | | | | | | |
| FPT_ITI.2 | | | | | | | | |
| FPT_ITT.1 | | | | | | | | |
| FPT_ITT.2 | | | | | | | | |
| FPT_ITT.3 | | | | | | | | X |
| FPT_PHP.1 | | | | | | | | |
| FPT_PHP.2 | | | - | X | - | - | | |
| FPT_PHP.3 | | | | | | | | |
| FPT_RCV.1 | X | X | | | | | | |
| FPT_RCV.2 | X | X | | | | | | |
| FPT_RCV.3 | X | X | | | | | | |
| FPT_RCV.4 | X | | | | | | | |
| FPT_RPL.1 | | | | | | | | |
| FPT_RVM.1 | | | | | | | | |
| FPT_SEP.1 | | | | | | | | |
| FPT_SEP.2 | | | | | | | | |
| FPT_SEP.3 | | | | | | | | |
| FPT_SSP.1 | | | | | | | | X |
| FPT_SSP.2 | | | | | | | | X |
| FPT_STM.1 | | | | | | | | |
| FPT_TDC.1 | | | | | | | | |
| FPT_TRC.1 | | | | | | | | X |
| FPT_TST.1 | | | | | | | X | |

**Table 8 - Dependency table for Class FPT: Protection of the TSF**

| | ADV_SPM.1 | FPT_FLS.1 |
|---|---|---|
| FRU_FLT.1 | - | X |
| FRU_FLT.2 | - | X |
| FRU_PRS.1 | | |
| FRU_PRS.2 | | |
| FRU_RSA.1 | | |
| FRU_RSA.2 | | |

**Table 9 - Dependency table for Class FRU: Resource utilisation**

| | FIA_UAU.1 | FIA_UID.1 |
|---|---|---|
| FTA_LSA.1 | | |
| FTA_MCS.1 | | X |
| FTA_MCS.2 | | X |
| FTA_SSL.1 | X | - |
| FTA_SSL.2 | X | - |
| FTA_SSL.3 | | |
| FTA_TAB.1 | | |
| FTA_TAH.1 | | |
| FTA_TSE.1 | | |

**Table 10 - Dependency table for Class FTA: TOE access**

# B Functional classes, families, and components

# (normative)

535 The following annexes C through M provide the application notes for the functional classes defined in the main body of this part of the CC.

# C Class FAU: Security audit (normative)

536 CC audit families allow PP/ST authors the ability to define requirements for monitoring user activities and, in some cases, detecting real, potential, or imminent violations of the TSP. The TOE's security audit functions are defined to help monitor security-relevant events, and act as a deterrent against security violations. The requirements of the audit families refer to functions that include audit data protection, record format, and event selection, as well as analysis tools, violation alarms, and real-time analysis. The audit trail should be presented in human-readable format either directly (e.g. storing the audit trail in human-readable format) or indirectly (e.g. using audit reduction tools), or both.

537 While developing the security audit requirements, the PP/ST author should take note of the inter-relationships among the audit families and components. The potential exists to specify a set of audit requirements that comply with the family/component dependencies lists, while at the same time resulting in a deficient audit function (e.g. an audit function that requires all security relevant events to be audited but without the selectivity to control them on any reasonable basis such as individual user or object).

## C.1 Audit requirements in a distributed environment

538 The implementation of audit requirements for networks and other large systems may differ significantly from those needed for stand-alone systems. Larger, more complex and active systems require more thought concerning which audit data to collect and how this should be managed, due to lowered feasibility of interpreting (or even storing) what gets collected. The traditional notion of a time-sorted list or “trail” of audited events may not be applicable in a global asynchronous network with arbitrarily many events occurring at once.

539 Also, different hosts and servers on a distributed TOE may have differing naming policies and values. Symbolic names presentation for audit review may require a net-wide convention to avoid redundancies and “name clashes.”

540 A multi-object audit repository, portions of which are accessible by a potentially wide variety of authorised users, may be required if audit repositories are to serve a useful function in distributed systems.

541 Finally, misuse of authority by authorised users should be addressed by systematically avoiding local storage of audit data pertaining to administrator actions.

542 Figure 23 shows the decomposition of this class into its constituent components.

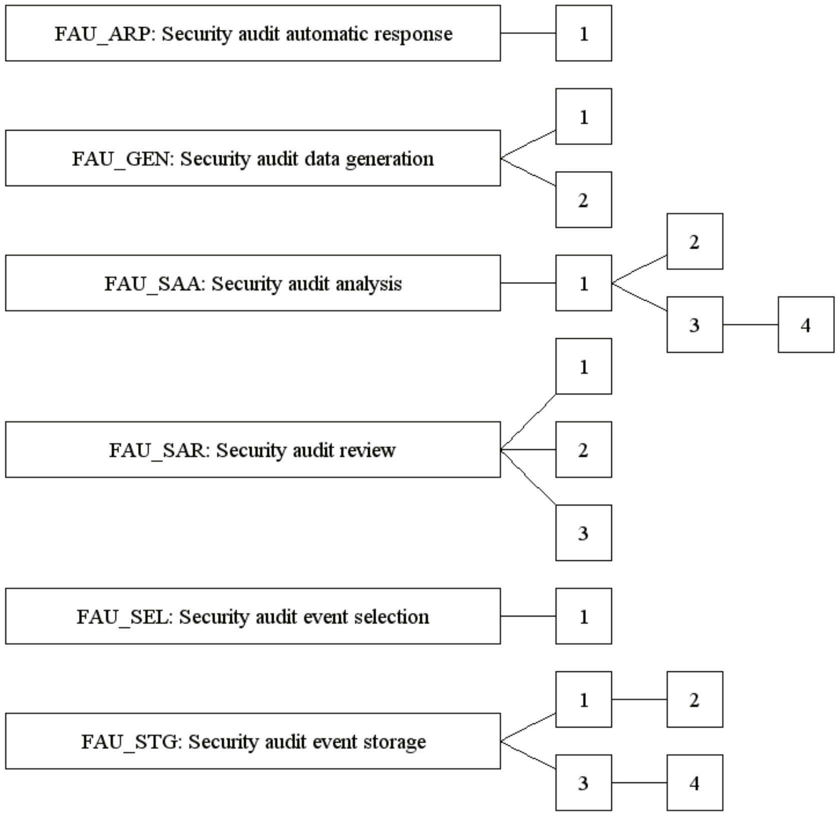



**Figure 23 - FAU: Security audit class decomposition**

## C.2 Security audit automatic response (FAU_ARP)

### Application notes

543 The Security audit automatic response family describes requirements for the handling of audit events. The requirement could include requirements for alarms or TSF action (automatic response). For example, the TSF could include the generation of real time alarms, termination of the offending process, disabling of a service, or disconnection or invalidation of a user account.

544 An audit event is defined to be an “potential security violation” if so indicated by the Security audit analysis (FAU_SAA) components.

### FAU_ARP.1 Security alarms

#### User application notes

545 An action should be taken for follow up action in the event of an alarm. This action can be to inform the authorised user, to present the authorised user with a set of possible containment actions, or to take corrective actions. The timing of the actions should be carefully considered by the PP/ST author.

### Operations

#### Assignment:

546 In **FAU_ARP.1.1**, the PP/ST author should specify the actions to be taken in case of a potential security violation. An example of such a list is: “inform the authorised user, disable the subject that created the potential security violation.” It can also specify that the action to be taken can be specified by an authorised user.

## C.3 Security audit data generation (FAU_GEN)

### Application notes

547 The Security audit data generation family includes requirements to specify the audit events that should be generated by the TSF for security-relevant events.

548 This family is presented in a manner that avoids a dependency on all components requiring audit support. Each component has an audit section developed in which the events to be audited for that functional area are listed. When the PP/ST author assembles the PP/ST, the items in the audit area are used to complete the variable in this component. Thus, the specification of what could be audited for a functional area is localised in that functional area.

549 The list of auditable events is entirely dependent on the other functional families within the PP/ST. Each family definition should therefore include a list of its family-specific auditable events. Each auditable event in the list of auditable events specified in the functional family should correspond to one of the levels of audit event generation specified in this family (i.e. minimal, basic, detailed). This provides the PP/ST author with information necessary to ensure that all appropriate auditable events are specified in the PP/ST. The following example shows how auditable events are to be specified in appropriate functional families:

550 “The following actions should be auditable if Security audit data generation (FAU_GEN) is included in the PP/ST:

a) Minimal: Successful use of the user security attribute administration functions;

b) Basic: All attempted uses of the user security attribute administration functions;

c) Basic: Identification of which user security attributes have been modified;

d) Detailed: With the exception of specific sensitive attribute data items (e.g. passwords, cryptographic keys), the new values of the attributes should be captured.”

551 For each functional component that is chosen, the auditable events that are indicated in that component, at and below the level indicated in Security audit data generation (FAU_GEN) should be auditable. If, for example, in the previous example "Basic" would be selected in Security audit data generation (FAU_GEN), the auditable events mentioned in a), b) and c) should be auditable.

552 Observe that the categorisation of auditable events is hierarchical. For example, when Basic Audit Generation is desired, all auditable events identified as being either Minimal or Basic, should also be included in the PP/ST through the use of the appropriate assignment operation, except when the higher level event simply provides more detail than the lower level event. When Detailed Audit Generation is desired, all identified auditable events (Minimal, Basic, and Detailed) should be included in the PP/ST.

553 A PP/ST author may decide to include other auditable events beyond those required for a given audit level. For example, the PP/ST may claim only minimal audit capabilities while including most of the basic capabilities because the few excluded capabilities conflict with other PP/ST constraints (e.g. because they require the collection of unavailable data).

554 The functionality that creates the auditable event should be specified in the PP or ST as a functional requirement.

555 The following are examples of the types of the events that should be defined as auditable within each PP/ST functional component:

a) Introduction of objects within the TSC into a subject's address space;

b) Deletion of objects;

c) Distribution or revocation of access rights or capabilities;

d) Changes to subject or object security attributes;

e) Policy checks performed by the TSF as a result of a request by a subject;

f) The use of access rights to bypass a policy check;

g) Use of Identification and Authentication functions;

h) Actions taken by an operator, and/or authorised user (e.g. suppression of a TSF protection mechanism as human-readable labels);

i) Import/export of data from/to removable media (e.g. printed output, tapes, diskettes).

## FAU_GEN.1 Audit data generation

### User application notes

556 This component defines requirements to identify the auditable events for which audit records should be generated, and the information to be provided in the audit records.

557 FAU_GEN.1 Audit data generation by itself might be used when the TSP does not require that individual user identities be associated with audit events. This could be appropriate when the PP/ST also contains privacy requirements. If the user identity must be incorporated FAU_GEN.2 User identity association could be used in addition.

### Evaluator notes

558 There is a dependency on Time stamps (FPT_STM). If correctness of time is not an issue for this TOE, elimination of this dependency could be justified.

### Operations

#### Selection:

559 In **FAU_GEN.1.1**, the PP/ST author should select the level of auditable events called out in the audit section of other functional components included in the PP/ST. This level is one of the following: “minimum”, “basic”, “detailed” or “not specified”.

#### Assignment:

560 In **FAU_GEN.1.1**, the PP/ST author should assign a list of other specifically defined auditable events to be included in the list of auditable events. The assignment may comprise none, or events that could be auditable events of a functional requirement that are of a higher audit level than requested in b), as well as the events generated through the use of a specified Application Programming Interface (API).

561 In **FAU_GEN.1.2**, the PP/ST author should assign, for each auditable events included in the PP/ST, either a list of other audit relevant information to be included in audit events records or none.

## FAU_GEN.2 User identity association

### User application notes

562 This component addresses the requirement of accountability of auditable events at the level of individual user identity. This component should be used in addition to FAU_GEN.1 Audit data generation.

563 There is a potential conflict between the audit and privacy requirements. For audit purposes it may be desirable to know who performed an action. The

user may want to keep his/her actions to himself/herself and not be identified by other persons (e.g. a site with job offers). Or it might be required in the Organisational Security Policy that the identity of the users must be protected. In those cases the objectives for audit and privacy might contradict each other. Therefore if this requirement is selected and privacy is important, inclusion of the component user pseudonimity might be considered. Requirements on determining the real user name based on its pseudonym are specified in the privacy class.

## C.4 Security audit analysis (FAU_SAA)

### Application notes

564 This family defines requirements for automated means that analyse system activity and audit data looking for possible or real security violations. This analysis may work in support of intrusion detection, or automatic response to an imminent security violation.

565 The action to be performed by the TSF on detection of a possible imminent or potential violation is defined in Security audit automatic response (FAU_ARP) components.

566 For real-time analysis, audit data could be transformed into a useful format for automated treatment, but into a different useful format for delivery to authorised users for review.

### FAU_SAA.1 Potential violation analysis

#### User application notes

567 This component is used to specify the set of auditable events whose occurrence or accumulated occurrence held to indicate a potential violation of the TSP, and any rules to be used to perform the violation analysis.

#### Operations

##### Assignment:

568 In **FAU_SAA.1.2**, the PP/ST author should identify the subset of defined auditable events whose occurrence or accumulated occurrence need to be detected as an indication of a potential violation of the TSP.

569 In **FAU_SAA.1.2**, the PP/ST author should specify any other rules that the TSF should use in its analysis of the audit trail. Those rules could include specific requirements to express the needs for the events to occur in a certain period of time (e.g. period of the day, duration). If there are no additional rules that the TSF should use in the analysis of the audit trail, this assignment can be completed with “none”.

## FAU_SAA.2 Profile based anomaly detection

### User application notes

570 A *profile* is a structure that characterises the behaviour of users and/or subjects; it represents how the users/subjects interact with the TSF in a variety of ways. Patterns of usage are established with respect to the various types of activity the users/subjects engage in (e.g. patterns in exceptions raised, patterns in resource utilisation (when, which, how), patterns in actions performed). The ways in which the various types of activity are recorded in the profile (e.g. resource measures, event counters, timers) are referred to as *profile metrics*.

571 Each profile represents the expected patterns of usage performed by members of the *profile target group*. This pattern may be based on past use (historical patterns) or on normal use for users of similar target groups (expected behaviour). A profile target group refers to one or more users who interact with the TSF. The activity of each member of the profile group is used by the analysis tool in establishing the usage patterns represented in the profile. The following are some examples of profile target groups:

a) **Single user account**: one profile per user;

b) **Group ID or Group Account**: one profile for all users who possess the same group ID or operate using the same group account;

c) **Operating Role**: one profile for all users sharing a given operating role;

d) **System**: one profile for all users of a system.

572 Each member of a profile target group is assigned an individual *suspicion rating* that represents how closely that member's new activity corresponds to the established patterns of usage represented in the group profile.

573 The sophistication of the anomaly detection tool will largely be determined by the number of target profile groups required by the PP/ST and the complexity of the required profile metrics.

574 This component is used to specify the set of auditable events whose occurrence or accumulated occurrence indicates a potential violation of the TSP, and any rules to be used to perform the violation analysis. This set of events or rules could be modified by the authorised user, through addition, modification or deletion of events or rules.

575 The PP/ST author should enumerate specifically what activity should be monitored and/or analysed by the TSF. The PP/ST author should also identify specifically what information pertaining to the activity is necessary to construct the usage profiles.

576 FAU_SAA.2 Profile based anomaly detection requires that the TSF maintain profiles of system usage. The word maintain implies that the anomaly

detector is actively updating the usage profile based on new activity performed by the profile target members. It is important here that the metrics for representing user activity are defined by the PP/ST author. For example, there may be a thousand different actions an individual may be capable of performing, but the anomaly detector may choose to monitor a subset of that activity. Anomalous activity gets integrated into the profile just like non-anomalous activity (assuming the tool is monitoring those actions). Things that may have appeared anomalous four months ago, might over time become the norm (and vice-versa) as the user's work duties change. The TSF wouldn't be able to capture this notion if it filtered out anomalous activity from the profile updating algorithms.

577 Administrative notification should be provided such that the authorised user understands the significance of the suspicion rating.

578 The PP/ST author should define how to interpret suspicion ratings and the conditions under which anomalous activity is indicated to the Security audit automatic response (FAU_ARP) mechanism.

### Operations

#### Assignment:

579 In **FAU_SAA.2.1**, the PP/ST author should specify the profile target group. A single PP/ST may include multiple profile target groups.

580 In **FAU_SAA.2.3**, the PP/ST author should specify conditions under which anomalous activity is reported by the TSF. Conditions may include the suspicion rating reaching a certain value, or be based on the type of anomalous activity observed.

## FAU_SAA.3 Simple attack heuristics

### User application notes

581 In practice, it is at best rare when an analysis tool can detect with certainty when a security violation is imminent. However, there do exist some system events that are so significant that they are always worthy of independent review. Example of such events include the deletion of a key TSF security data file (e.g. the password file) or activity such as a remote user attempting to gain administrative privilege. These events are referred to as signature events in that their occurrence in isolation from the rest of the system activity are indicative of intrusive activity.

582 The complexity of a given tool will depend greatly on the assignments defined by the PP/ST author in identifying the base set of *signature events*.

583 The PP/ST author should enumerate specifically what events should be monitored by the TSF in order to perform the analysis. The PP/ST author should identify specifically what information pertaining to the event is necessary to determine if the event maps to a signature event.

584 Administrative notification should be provided such that the authorised user understands the significance of the event and the appropriate possible responses.

585 An effort was made in the specification of these requirements to avoid a dependency on audit data as the sole input for monitoring system activity. This was done in recognition of the existence of previously developed intrusion detection tools that do not perform their analyses of system activity solely through the use of audit data (examples of other input data include network datagrams, resource/accounting data, or combinations of various system data).

586 The elements of FAU_SAA.3 Simple attack heuristics do not require that the TSF implementing the immediate attack heuristics be the same TSF whose activity is being monitored. Thus, one can develop an intrusion detection component that operates independently of the system whose system activity is being analysed.

### Operations

#### Assignment:

587 In **FAU_SAA.3.1**, the PP/ST author should identify a base subset of system events whose occurrence, in isolation from all other system activity, may indicate a violation of the TSP. These include events that by themselves indicate a clear violation to the TSP, or whose occurrence is so significant that they warrant actions.

588 In **FAU_SAA.3.2**, the PP/ST author should specify the information used to determine system activity. This information is the input data used by the analysis tool to determine the system activity that has occurred on the TOE. This data may include audit data, combinations of audit data with other system data, or may consist of data other than the audit data. The PP/ST author should define precisely what system events and event attributes are being monitored within the input data.

## FAU_SAA.4 Complex attack heuristics

### User application notes

589 In practice, it is at best rare when an analysis tool can detect with certainty when a security violation is imminent. However, there do exist some system events that are so significant they are always worthy of independent review. Example of such events include the deletion of a key TSF security data file (e.g. the password file) or activity such as a remote user attempting to gain administrative privilege. These events are referred to as signature events in that their occurrence in isolation from the rest of the system activity are indicative of intrusive activity. Event sequences are an ordered set of signature events that might indicate intrusive activity.

590 The complexity of a given tool will depend greatly on the assignments defined by the PP/ST author in identifying the base set of signature events and event sequences.

591 The PP/ST author should define a base set of signature events and event sequences to be represented by the TSF. Additional signature events and event sequences may be defined by the system developer.

592 The PP/ST author should enumerate specifically what events should be monitored by the TSF in order to perform the analysis. The PP/ST author should identify specifically what information pertaining to the event is necessary to determine if the event maps to a signature event.

593 Administrative notification should be provided such that the authorised user understands the significance of the event and the appropriate possible responses.

594 An effort was made in the specification of these requirements to avoid a dependency on audit data as the sole input for monitoring system activity. This was done in recognition of the existence of previously developed intrusion detection tools that do not perform their analyses of system activity solely through the use of audit data (examples of other input data include network datagrams, resource/accounting data, or combinations of various system data). Levelling, therefore, requires the PP/ST author to specify the type of input data used to monitor system activity.

595 The elements of FAU_SAA.4 Complex attack heuristics do not require that the TSF implementing the complex attack heuristics be the same TSF whose activity is being monitored. Thus, one can develop an intrusion detection component that operates independently of the system whose system activity is being analysed.

### Operations

#### Assignment:

596 In **FAU_SAA.4.1**, the PP/ST author should identify a base set of list of sequences of system events whose occurrence are representative of known penetration scenarios. These event sequences represent known penetration scenarios. Each event represented in the sequence should map to a monitored system event, such that as the system events are performed, they are bound (mapped) to the known penetration event sequences.

597 In **FAU_SAA.4.1**, the PP/ST author should identify a base subset of system events whose occurrence, in isolation from all other system activity, may indicate a violation of the TSP. These include events that by themselves indicate a clear violation to the TSP, or whose occurrence is so significant they warrant action.

598 In **FAU_SAA.4.2**, the PP/ST author should specify the information used to determine system activity. This information is the input data used by the analysis tool to determine the system activity that has occurred on the TOE. This data may include audit data, combinations of audit data with other system data, or may consist of data other than the audit data. The PP/ST author should define precisely what system events and event attributes are being monitored within the input data.

## C.5 Security audit review (FAU_SAR)

### Application notes

599 The Security audit review family defines requirements related to review of the audit information.

600 These functions should allow pre-storage or post-storage audit selection that includes, for example, the ability to selectively review:

- the actions of one or more users (e.g. identification, authentication, TOE entry, and access control actions);
- the actions performed on a specific object or TOE resource;
- all of a specified set of audited exceptions; or
- actions associated with a specific TSP attribute.

601 The distinction between audit reviews is based on functionality. Audit review (only) encompasses the ability to view audit data. Selectable review is more sophisticated, and requires the ability to perform searches based on a single criterion or multiple criteria with logical (i.e. and/or) relations, sort audit data, filter audit data, before audit data are reviewed.

### FAU_SAR.1 Audit review

#### Rationale

602 This component will provide authorised users the capability to obtain and interpret the information. In case of human users this information needs to be in a human understandable presentation. In case of external IT entities the information needs to be unambiguously represented in an electronic fashion.

#### User application notes

603 This component is used to specify that users and/or authorised users can read the audit records. These audit records will be provided in a manner appropriate to the user. There are different types of users (human users, machine users) that might have different needs.

604 The content of the audit records that can be viewed can be specified.

### Operations

#### Assignment:

605 In **FAU_SAR.1.1**, the PP/ST author should specify the authorised users that can use this capability. If appropriate the PP/ST author may include security roles (see FMT_SMR.1 Security roles).

606 In **FAU_SAR.1.1**, the PP/ST author should specify the type of information the specified user is permitted to obtain from the audit records. Examples are “all”, “subject identity”, “all information belonging to audit records referencing this user”.

## FAU_SAR.2 Restricted audit review

### User application notes

607 This component specifies that any users not identified in FAU_SAR.1 Audit review will not be able to read the audit records.

## FAU_SAR.3 Selectable audit review

### User application notes

608 This component is used to specify that it should be possible to perform selection of the audit data to be reviewed. If based on multiple criteria, those criteria should be related together with logical (i.e. “and” or “or”) relations, and the tools should provide the ability to manipulate audit data (e.g. sort, filter).

### Operations

#### Selection:

609 In **FAU_SAR.3.1**, the PP/ST author should select whether searches, sorting and/or ordering can be performed by the TSF.

#### Assignment:

610 In **FAU_SAR.3.1**, the PP/ST author should assign the criteria, possibly with logical relations, to be used to select the audit data for review. The logical relations are intended to specify whether the operation can be on an individual attribute or a collection of attributes. An example of this assignment could be: “application, user account and/or location”. In this case the operation could be specified using any combination of the three attributes: application, user account and location.

## C.6 Security audit event selection (FAU_SEL)

### Application notes

611 The Security audit event selection family provides requirements related to the capabilities of identifying which of the possible auditable events are to be audited. The auditable events are defined in the Security audit data generation (FAU_GEN) family, but those events should be defined as being selectable in this component to be audited.

612 This family ensures that it is possible to keep the audit trail from becoming so large that it becomes useless, by defining the appropriate granularity of the selected security audit events.

### FAU_SEL.1 Selective audit

#### User application notes

613 This component defines the criteria used for the selection of events to be audited. Those criteria could permit inclusion or exclusion of events from the set of auditable events, based on user attributes, subject attributes, objects attributes, or event types.

614 The existence of individual user identities is not assumed for this component. This allows for TOEs such as routers that may not support the notion of users.

615 For a distributed environment, the host identity could be used as a selection criteria for events to be audited.

616 The management function FMT_MTD.1 Management of TSF data will handle the rights of authorised users to query or modify the selections.

#### Operations

##### Selection:

617 In **FAU_SEL.1.1**, the PP/ST author should select whether the security attributes upon which audit selectivity is based, is related to object identity, user identity, subject identity, host identity, or event type.

##### Assignment:

618 In **FAU_SEL.1.1**, the PP/ST author should specify any additional attributes upon which audit selectivity is based. If there are no additional rules upon which audit selectivity is based, this assignment can be completed with “none”.

## C.7 Security audit event storage (FAU_STG)

#### Application notes

619 The Security audit event storage family describes requirements for storing audit data for later use, including requirements controlling the loss of audit information due to system failure, attack and/or exhaustion of storage space.

### FAU_STG.1 Protected audit trail storage

#### User application notes

620 In a distributed environment, as the location of the audit trail is in the TSC, but not necessarily co-located with the function generating the audit data, the PP/ST author could request authentication of the originator of the audit record, or non-repudiation of the origin of the record prior storing this record in the audit trail.

621 The TSF will protect the audit trail from unauthorised deletion and modification. It is noted that in some systems the auditor (role) might not be authorised to delete the audit records for a certain period of time.

#### Operations

##### Selection:

622 In **FAU_STG.1.2**, the PP/ST author should specify whether the TSF shall prevent or only be able to detect modifications of the audit trail. Only one of these options may be chosen.

### FAU_STG.2 Guarantees of audit data availability

#### User application notes

623 This component allows the PP/ST author to specify to which metrics the audit trail should conform.

624 In a distributed environment, as the location of the audit trail is in the TSC, but not necessarily co-located with the function generating the audit data, the PP/ST author could request authentication of the originator of the audit record, or non-repudiation of the origin of the record prior storing this record in the audit trail.

#### Operations

##### Selection:

625 In **FAU_STG.2.2**, the PP/ST author should specify whether the TSF shall prevent or only be able to detect modifications of the audit trail. Only one of these options may be chosen.

Assignment:

626 In **FAU_STG.2.3**, the PP/ST author should specify the metric that the TSF must ensure with respect to the audit trail. This metric limits the data loss by enumerating the number of records that must be kept, or the time that records are guaranteed to be maintained. An example of the metric could be "100,000" indicating that 100,000 audit records can be stored.

Selection:

627 In **FAU_STG.2.3**, the PP/ST author should specify the condition under which the TSF shall still be able to maintain a defined amount of audit data. This condition can be any of the following: audit storage exhaustion, failure, attack.

## FAU_STG.3 Action in case of possible audit data loss

User application notes

628 This component requires that actions will be taken when the audit trail exceeds certain pre-defined limits.

Operations

Assignment:

629 In **FAU_STG.3.1**, the PP/ST author should indicate the pre-defined limit. If the management functions indicate that this number might be changed by the authorised user, this value is the default value. The PP/ST author might choose to let the authorised user define this limit. In that case the assignment can be for example "an authorised user set limit".

630 In **FAU_STG.3.1**, the PP/ST author should specify actions that should be taken in case of imminent audit storage failure indicated by exceeding the threshold. Actions might include informing an authorised user.

## FAU_STG.4 Prevention of audit data loss

User application notes

631 This component specifies the behaviour of the TOE if the audit trail is full: either audit records are ignored, or the TOE is frozen such that no auditable events can take place. The requirement also states that no matter how the requirement is instantiated, the authorised user with specific rights to this effect, can continue to generate auditable events (actions). The reason is that otherwise the authorised user could not even reset the system. Consideration should be given to the choice of the action to be taken by the TSF in the case of audit storage exhaustion, as ignoring events, which provides better

availability of the TOE, will also permit actions to be performed without being recorded and without the user being accountable.

### Operations

#### Selection:

632 In **FAU_STG.4.1**, the PP/ST author should select whether the TSF shall ignore auditable actions, or whether it should prevent auditable actions from happening, or whether the oldest audit records should be overwritten when the TSF can no longer store audit records. Only one of these options may be chosen.

#### Assignment:

633 In **FAU_STG.4.1**, the PP/ST author should specify other actions that should be taken in case of audit storage failure, such as informing the authorised user. If there is no other action to be taken in case of audit storage failure, this assignment can be completed with “none”.

# D Class FCO: Communication (normative)

634 This class describes requirements specifically of interest for TOEs that are used for the transport of information. Families within this class deal with non-repudiation.

635 In this class the concept of “information” is used. This information should be interpreted as the object being communicated, and could contain an electronic mail message, a file, or a set of predefined attribute types.

636 In the literature, the terms “proof of receipt” and “proof of origin” are commonly used terms. However it is recognised that the term “proof” might be interpreted in a legal sense to imply a form of mathematical rationale. The components in this class interpret the de-facto use of the word “proof” in the context of “evidence” that the TSF demonstrates the non-repudiated transport of types of information.

637 Figure 24 shows the decomposition of this class into its constituent components.

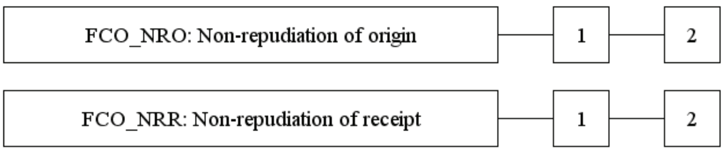



**Figure 24 - FCO: Communication class decomposition**

## D.1 Non-repudiation of origin (FCO_NRO)

### User notes

638 Non-repudiation of origin defines requirements to provide evidence to users/subjects about the identity of the originator of some information. The originator cannot successfully deny having sent the information because evidence of origin (e.g. digital signature) provides evidence of the binding between the originator and the information sent. The recipient or a third party can verify the evidence of origin. This evidence should not be forgeable.

639 If the information or the associated attributes are altered in any way, validation of the evidence of origin might fail. Therefore a PP/ST author should consider including integrity requirements such as FDP_UIT.1 Data exchange integrity in the PP/ST.

640 In non-repudiation there are several different roles involved, each of which could be combined in one or more subjects. The first role is a subject that requests evidence of origin (only in FCO_NRO.1 Selective proof of origin). The second role is the recipient and/or other subjects to which the evidence is

provided (e.g. a notary). The third role is a subject that requests verification of the evidence of origin, for example, a recipient or a third party such as an arbiter.

641 The PP/ST author must specify the conditions that must be met to be able to verify the validity of the evidence. An example of a condition which could be specified is where the verification of evidence must occur within 24 hours. These conditions, therefore, allow the tailoring of the non-repudiation to legal requirements, such as being able to provide evidence for several years.

642 In most cases, the identity of the recipient will be the identity of the user who received the transmission. In some instances, the PP/ST author does not want the user identity to be exported. In that case the PP/ST author must consider whether it is appropriate to include this class, or whether the identity of the transport service provider or the identity of the host should be used.

643 In addition to (or instead of) the user identity, a PP/ST author might be more concerned about the time the information was transmitted. For example, requests for proposals must be transmitted before a certain date in order to be considered. In such instances, these requirements can be customised to provide a timestamp indication (time of origin).

## FCO_NRO.1 Selective proof of origin

### Operations

#### Assignment:

644 In **FCO_NRO.1.1**, the PP/ST author should fill in the types of information subject to the evidence of origin function, for example, electronic mail messages.

#### Selection:

645 In **FCO_NRO.1.1**, the PP/ST author should specify the user/subject who can request evidence of origin.

#### Assignment:

646 In **FCO_NRO.1.1**, the PP/ST author, dependent on the selection, should specify the third parties that can request evidence of origin. A third party could be an arbiter, judge or legal body.

647 In **FCO_NRO.1.2**, the PP/ST author should fill in the list of the attributes that shall be linked to the information; for example, originator identity, time of origin, and location of origin.

648 In **FCO_NRO.1.2**, the PP/ST author should fill in the list of information fields within the information over which the attributes provide evidence of origin, such as the body of a message.

Selection:

649 In **FCO_NRO.1.3**, the PP/ST author should specify the user/subject who can verify the evidence of origin.

Assignment:

650 In **FCO_NRO.1.3**, the PP/ST author should fill in the list of limitations under which the evidence can be verified. For example the evidence can only be verified within a 24 hour time interval. An assignment of “immediate” or “indefinite” is acceptable.

651 In **FCO_NRO.1.3**, the PP/ST author, dependent on the selection, should specify the third parties that can verify the evidence of origin.

### FCO_NRO.2 Enforced proof of origin

Operations

Assignment:

652 In **FCO_NRO.2.1**, the PP/ST author should fill in the types of information subject to the evidence of origin function, for example, electronic mail messages.

653 In **FCO_NRO.2.2**, the PP/ST author should fill in the list of the attributes that shall be linked to the information; for example, originator identity, time of origin, and location of origin.

654 In **FCO_NRO.2.2**, the PP/ST author should fill in the list of information fields within the information over which the attributes provide evidence of origin, such as the body of a message.

Selection:

655 In **FCO_NRO.2.3**, the PP/ST author should specify the user/subject who can verify the evidence of origin.

Assignment:

656 In **FCO_NRO.2.3**, the PP/ST author should fill in the list of limitations under which the evidence can be verified. For example the evidence can only be verified within a 24 hour time interval. An assignment of “immediate” or “indefinite” is acceptable.

657 In **FCO_NRO.2.3**, the PP/ST author, dependent on the selection, should specify the third parties that can verify the evidence of origin. A third party could be an arbiter, judge or legal body.

## D.2 Non-repudiation of receipt (FCO_NRR)

### User notes

658 Non-repudiation of receipt defines requirements to provide evidence to other users/subjects that the information was received by the recipient. The recipient cannot successfully deny having received the information because evidence of receipt (e.g. digital signature) provides evidence of the binding between the recipient attributes and the information. The originator or a third party can verify the evidence of receipt. This evidence should not be forgeable.

659 It should be noted that the provision of evidence that the information was received does not necessarily imply that the information was read or comprehended, but only delivered

660 If the information or the associated attributes are altered in any way, validation of the evidence of receipt with respect to the original information might fail. Therefore a PP/ST author should consider including integrity requirements such as FDP_UIT.1 Data exchange integrity in the PP/ST.

661 In non-repudiation, there are several different roles involved, each of which could be combined in one or more subjects. The first role is a subject that requests evidence of receipt (only in FCO_NRR.1 Selective proof of receipt). The second role is the recipient and/or other subjects to which the evidence is provided, (e.g. a notary). The third role is a subject that requests verification of the evidence of receipt, for example, an originator or a third party such as an arbiter.

662 The PP/ST author must specify the conditions that must be met to be able to verify the validity of the evidence. An example of a condition which could be specified is where the verification of evidence must occur within 24 hours. These conditions, therefore, allow the tailoring of the non-repudiation to legal requirements, such as being able to provide evidence for several years.

663 In most cases, the identity of the recipient will be the identity of the user who received the transmission. In some instances, the PP/ST author does not want the user identity to be exported. In that case, the PP/ST author must consider whether it is appropriate to include this class, or whether the identity of the transport service provider or the identity of the host should be used.

664 In addition to (or instead of) the user identity, a PP/ST author might be more concerned about the time the information was received. For example, when an offer expires at a certain date, orders must be received before a certain date in order to be considered. In such instances, these requirements can be customised to provide a timestamp indication (time of receipt).

### FCO_NRR.1 Selective proof of receipt

Operations

Assignment:

665 In **FCO_NRR.1.1**, the PP/ST author should fill in the types of information subject to the evidence of receipt function, for example, electronic mail messages.

Selection:

666 In **FCO_NRR.1.1**, the PP/ST author should specify the user/subject who can request evidence of receipt.

Assignment:

667 In **FCO_NRR.1.1**, the PP/ST author, dependent on the selection, should specify the third parties that can request evidence of receipt. A third party could be an arbiter, judge or legal body.

668 In **FCO_NRR.1.2**, the PP/ST author should fill in the list of the attributes that shall be linked to the information; for example, recipient identity, time of receipt, and location of receipt.

669 In **FCO_NRR.1.2**, the PP/ST author should fill in the list of information fields with the fields within the information over which the attributes provide evidence of receipt, such as the body a message.

Selection:

670 In **FCO_NRR.1.3**, the PP/ST author should specify the user/subjects who can verify the evidence of receipt.

Assignment:

671 In **FCO_NRR.1.3**, the PP/ST author should fill in the list of limitations under which the evidence can be verified. For example the evidence can only be verified within a 24 hour time interval. An assignment of “immediate” or “indefinite” is acceptable.

672 In **FCO_NRR.1.3**, the PP/ST author, dependent on the selection, should specify the third parties that can verify the evidence of receipt.

## FCO_NRR.2 Enforced proof of receipt

### Operations

#### Assignment:

673 In **FCO_NRR.2.1**, the PP/ST author should fill in the types of information subject to the evidence of receipt function, for example electronic mail messages.

674 In **FCO_NRR.2.2**, the PP/ST author should fill in the list of the attributes that shall be linked to the information; for example, recipient identity, time of receipt, and location of receipt.

675 In **FCO_NRR.2.2**, the PP/ST author should fill in the list of information fields with the fields within the information over which the attributes provide evidence of receipt, such as the body of a message.

#### Selection:

676 In **FCO_NRR.2.3**, the PP/ST author should specify the user/subjects who can verify the evidence of receipt.

#### Assignment:

677 In **FCO_NRR.2.3**, the PP/ST author should fill in the list of limitations under which the evidence can be verified. For example the evidence can only be verified within a 24 hour time interval. An assignment of "immediate" or "indefinite" is acceptable.

678 In **FCO_NRR.2.3**, the PP/ST author, dependent on the selection, should specify the third parties that can verify the evidence of receipt. A third party could be an arbiter, judge or legal body.

# E Class FCS: Cryptographic support (normative)

679 The TSF may employ cryptographic functionality to help satisfy several high-level security objectives. These include (but are not limited to): identification and authentication, non-repudiation, trusted path, trusted channel and data separation. This class is used when the TOE implements cryptographic functions, the implementation of which could be in hardware, firmware and/or software.

680 The FCS: Cryptographic support class is composed of two families: Cryptographic key management (FCS_CKM) and Cryptographic operation (FCS_COP). The Cryptographic key management (FCS_CKM) family addresses the management aspects of cryptographic keys, while the Cryptographic operation (FCS_COP) family is concerned with the operational use of those cryptographic keys.

681 For each cryptographic key generation method implemented by the TOE, if any, the PP/ST author should select the FCS_CKM.1 Cryptographic key generation component.

682 For each cryptographic key distribution method implemented by the TOE, if any, the PP/ST author should select the FCS_CKM.2 Cryptographic key distribution component.

683 For each cryptographic key access method implemented by the TOE, if any, the PP/ST author should select the FCS_CKM.3 Cryptographic key access component.

684 For each cryptographic key destruction method implemented by the TOE, if any, the PP/ST author should select the FCS_CKM.4 Cryptographic key destruction component.

685 For each cryptographic operation (such as digital signature, data encryption, key agreement, secure hash, etc.) performed by the TOE, if any, the PP/ST author should select the FCS_COP.1 Cryptographic operation component.

686 Cryptographic functionality may be used to meet objectives specified in class FCO: Communication, and in families Data authentication (FDP_DAU), Stored data integrity (FDP_SDI), Inter-TSF user data confidentiality transfer protection (FDP_UCT), Inter-TSF user data integrity transfer protection (FDP_UIT), Specification of secrets (FIA_SOS), User authentication (FIA_UAU), to meet a variety of objectives. In the cases where cryptographic functionality is used to meet objectives for other classes, the individual functional components specify the objectives that cryptographic functionality must satisfy. The objectives in class FCS: Cryptographic support should be used when cryptographic functionality of the TOE is sought by consumers.

687 Figure 25 shows the decomposition of this class into its constituent components.

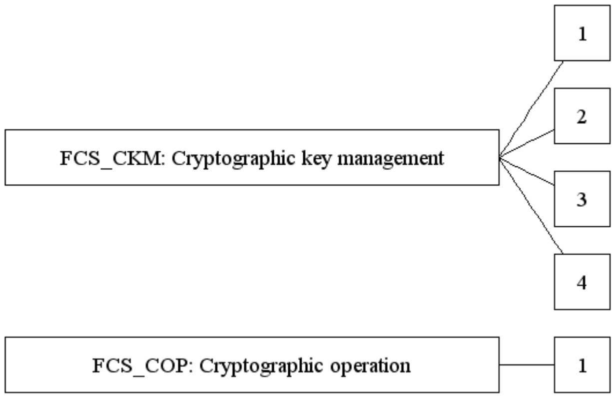



**Figure 25 - FCS: Cryptographic support class decomposition**

## E.1 Cryptographic key management (FCS_CKM)

### User notes

688 Cryptographic keys must be managed throughout their lifetime. The typical events in the lifecycle of a cryptographic key include (but are not limited to): generation, distribution, entry, storage, access (e.g. backup, escrow, archive, recovery) and destruction.

689 The inclusion of other stages is dependent on the key management strategy being implemented, as the TOE need not be involved in all of the key life-cycle (e.g. the TOE may only generate and distribute cryptographic keys).

690 This family is intended to support the cryptographic key lifecycle and consequently defines requirements for the following activities: cryptographic key generation, cryptographic key distribution, cryptographic key access and cryptographic key destruction. This family should be included whenever there are functional requirements for the management of cryptographic keys.

691 If Security audit data generation (FAU_GEN) Security Audit Data Generation is included in the PP/ST then, in the context of the events being audited:

a) The object attributes may include the assigned user for the cryptographic key, the user role, the cryptographic operation that the cryptographic key is to be used for, the cryptographic key identifier and the cryptographic key validity period.

b) The object value may include the values of cryptographic key(s) and parameters excluding any sensitive information (such as secret or private cryptographic keys).

692 Typically, random numbers are used to generate cryptographic keys. If this is the case, then FCS_CKM.1 Cryptographic key generation Cryptographic key

generation should be used instead of the component FIA_SOS.2 TSF Generation of secrets TSF Generation of secrets. In cases where random number generation is required for purposes other than for the generation of cryptographic keys, the component FIA_SOS.2 TSF Generation of secrets TSF Generation of secrets should be used.

## FCS_CKM.1 Cryptographic key generation

### User application notes

693 This component requires the cryptographic key sizes and method used to generate cryptographic keys to be specified, this can be in accordance with an assigned standard. It should be used to specify the cryptographic key sizes and the method (e.g. algorithm) used to generate the cryptographic keys. Only one instance of the component is needed for the same method and multiple key sizes. The key size could be common or different for the various entities, and could be either the input to or the output from the method.

### Operations

#### Assignment:

694 In **FCS_CKM.1.1**, the PP/ST author should specify the cryptographic key generation algorithm to be used.

695 In **FCS_CKM.1.1**, the PP/ST author should specify the cryptographic key sizes to be used. The key sizes specified should be appropriate for the algorithm and its intended use.

696 In **FCS_CKM.1.1**, the PP/ST author should specify the assigned standard that documents the method used to generate cryptographic keys. The assigned standard may comprise none, one or more actual standards publications, for example, from international, national, industry or organisational standards.

## FCS_CKM.2 Cryptographic key distribution

### User application notes

697 This component requires the method used to distribute cryptographic keys to be specified, this can be in accordance with an assigned standard.

### Operations

#### Assignment:

698 In **FCS_CKM.2.1**, the PP/ST author should specify the cryptographic key distribution method to be used.

699 In **FCS_CKM.2.1**, the PP/ST author should specify the assigned standard that documents the method used to distribute cryptographic

keys. The assigned standard may comprise none, one or more actual standards publications, for example, from international, national, industry or organisational standards.

## FCS_CKM.3 Cryptographic key access

### User application notes

700 This component requires the method used to access cryptographic keys be specified, this can be in accordance with an assigned standard.

### Operations

#### Assignment:

701 In **FCS_CKM.3.1**, the PP/ST author should specify the type of cryptographic key access being used. Examples of types of cryptographic key access include (but are not limited to) cryptographic key backup, cryptographic key archival, cryptographic key escrow and cryptographic key recovery.

702 In **FCS_CKM.3.1**, the PP/ST author should specify the cryptographic key access method to be used.

703 In **FCS_CKM.3.1**, the PP/ST author should specify the assigned standard that documents the method used to access cryptographic keys. The assigned standard may comprise none, one or more actual standards publications, for example, from international, national, industry or organisational standards.

## FCS_CKM.4 Cryptographic key destruction

### User application notes

704 This component requires the method used to destroy cryptographic keys be specified, this can be in accordance with an assigned standard.

### Operations

#### Assignment:

705 In **FCS_CKM.4.1**, the PP/ST author should specify the key destruction method to be used to destroy cryptographic keys.

706 In **FCS_CKM.4.1**, the PP/ST author should specify the assigned standard that documents the method used to destroy cryptographic keys. The assigned standard may comprise none, one or more actual standards publications, for example, from international, national, industry or organisational standards.

## E.2 Cryptographic operation (FCS_COP)

### User notes

707 A cryptographic operation may have cryptographic mode(s) of operation associated with it. If this is the case, then the cryptographic mode(s) must be specified. Examples of cryptographic modes of operation are cipher block chaining, output feedback mode, electronic code book mode, and cipher feedback mode.

708 Cryptographic operations may be used to support one or more TOE security services. The Cryptographic operation (FCS_COP) component may need to be iterated more than once depending on:

a) the user application for which the security service is being used.

b) the use of different cryptographic algorithms and/or cryptographic key sizes.

c) the type or sensitivity of the data being operated on.

709 If Security audit data generation (FAU_GEN) Security audit data generation is included in the PP/ST then, in the context of the cryptographic operation events being audited:

a) The types of cryptographic operation may include digital signature generation and/or verification, cryptographic checksum generation for integrity and/or for verification of checksum, secure hash (message digest) computation, data encryption and/or decryption, cryptographic key encryption and/or decryption, cryptographic key agreement and random number generation.

b) The subject attributes may include subject role(s) and user(s) associated with the subject.

c) The object attributes may include the assigned user for the cryptographic key, user role, cryptographic operation the cryptographic key is to be used for, cryptographic key identifier, and the cryptographic key validity period.

## FCS_COP.1 Cryptographic operation

### User application notes

710 This component requires the cryptographic algorithm and key size used to perform specified cryptographic operation(s) which can be based on an assigned standard.

### Operations

#### Assignment:

711 In **FCS_COP.1.1**, the PP/ST author should specify the cryptographic operations being performed. Typical cryptographic operations include digital signature generation and/or verification, cryptographic checksum generation for integrity and/or for verification of checksum, secure hash (message digest) computation, data encryption and/or decryption, cryptographic key encryption and/or decryption, cryptographic key agreement and random number generation. The cryptographic operation may be performed on user data or TSF data.

712 In **FCS_COP.1.1**, the PP/ST author should specify the cryptographic algorithm to be used. Typical cryptographic algorithms include, but are not limited to, DES, RSA and IDEA.

713 In **FCS_COP.1.1**, the PP/ST author should specify the cryptographic key sizes to be used. The key sizes specified should be appropriate for the algorithm and its intended use.

714 In **FCS_COP.1.1**, the PP/ST author should specify the assigned standard that documents how the identified cryptographic operation(s) are performed. The assigned standard may comprise none, one or more actual standards publications, for example, from international, national, industry or organisational standards.

# F Class FDP: User data protection (normative)

715 This class contains families specifying requirements for TOE security functions and TOE security function policies related to protecting user data. This class differs from FIA and FPT in that FDP: User data protection specifies components to protect user data, FIA specifies components to protect attributes associated with the user, and FPT specifies components to protect TSF information.

716 The class does not contain explicit requirements for traditional Mandatory Access Controls (MAC) or traditional Discretionary Access Controls (DAC); however, such requirements may be constructed using components from this class.

717 FDP: User data protection does not explicitly deal with confidentiality, integrity, or availability, as all three are most often intertwined in the policy and mechanisms. However, the TOE security policy must adequately cover these three objectives in the PP/ST.

718 A final aspect of this class is that it specifies access control in terms of "operations". An operation is defined as a specific type of access on a specific object. It depends on the level of abstraction of the PP/ST author whether these operations are described as "read" and/or "write" operations, or as more complex operations such as "update the database".

719 The access control policies are policies that control access to the information container. The attributes represent attributes of the container. Once the information is out of the container, the accessor is free to modify that information, including writing the information into a different container with different attributes. By contrast, an information flow policies controls access to the information, independent of the container. The attributes of the information, which may be associated with the attributes of the container (or may not, as in the case of a multi-level database) stay with the information as it moves. The accessor does not have the ability, in the absence of an explicit authorisation, to change the attributes of the information.

720 This class is not meant to be a complete taxonomy of IT access policies, as others can be imagined. Those policies included here are simply those for which current experience with actual systems provides a basis for specifying requirements. There may be other forms of intent that are not captured in the definitions here.

721 For example, one could imagine a goal of having user-imposed (and user-defined) controls on information flow (e.g. an automated implementation of the NO FOREIGN handling caveat). Such concepts could be handled as refinements of, or extensions to the FDP: User data protection components.

722 Finally, it is important when looking at the components in FDP: User data protection to remember that these components are requirements for functions that may be implemented by a mechanism that also serves or could serve another purpose. For example, it is possible to build an access control policy (Access control policy (FDP_ACC)) that uses labels (FDP_IFF.1 Simple security attributes) as the basis of the access control mechanism.

723 A TOE security policy may encompass many security function policies (SFPs), each to be identified by the two policy oriented components Access control policy (FDP_ACC), and Information flow control policy (FDP_IFC). These policies will typically take confidentiality, integrity, and availability aspects into consideration as required, to satisfy the TOE requirements. Care should be taken to ensure that all objects are covered by at least one SFP and that there are no conflicts arising from implementing the multiple SFPs.

724 When building a PP/ST using components from the FDP: User data protection class, the following information provides guidance on where to look and what to select from the class.

725 The requirements in the FDP: User data protection class are defined in terms of a security function (abbreviated SF) that will implement a SFP. Since a TOE may implement multiple SFPs simultaneously, the PP/ST author must specify the name for each SFP, so it can be referenced in other families. This name will then be used in each component selected to indicate that it is being used as part of the definition of requirements for that function. This allows the author to easily indicate the scope for operations such as objects covered, operations covered, authorised users, etc.

726 Each instantiation of a component can apply to only one SFP. Therefore if an SFP is specified in a component then this SFP will apply to all the elements in this component. The components may be instantiated multiple times within a PP/ST to account for different policies if so desired.

727 The key to selecting components from this family is to have a well defined TOE security policy to enable proper selection of the components from the two policy components; Access control policy (FDP_ACC) and Information flow control policy (FDP_IFC). In Access control policy (FDP_ACC) and Information flow control policy (FDP_IFC) respectively, all access control policies and all information flow control policies are named. Furthermore the scope of control of these components in terms of the subjects, objects and operations covered by this security function. The names of these policies are meant to be used throughout the remainder of the functional components that have an operation that calls for an assignment or selection of an “access control SFP” or an “information flow control SFP”. The rules that define the functionality of the named access control and information flow control SFPs will be defined in the Access control functions (FDP_ACF) and Information flow control functions (FDP_IFF) families (respectively).

728 The following steps are guidance on how this class is applied in the construction of a PP/ST:

a) Identify the policies to be enforced from the Access control policy (FDP_ACC), and Information flow control policy (FDP_IFC) families. These families define scope of control for the policy, granularity of control and may identify some rules to go with the policy.

b) Identify the components and perform any applicable operations in the policy components. The assignment operations may be performed generally (such as with a statement “All files”) or specifically (“The files “A”, “B”, etc.) depending upon the level of detail known.

c) Identify any applicable function components from the Access control functions (FDP_ACF) and Information flow control functions (FDP_IFF) families to address the named policy families from Access control policy (FDP_ACC) and Information flow control policy (FDP_IFC). Perform the operations to make the components define the rules to be enforced by the named policies. This should make the components fit the requirements of the selected function envisioned or to be built.

d) Identify who will have the ability to control and change security attributes under the function, such as only a security administrator, only the owner of the object, etc. Select the appropriate components from FMT: Security management and perform the operations. Refinements may be useful here to identify missing features, such as that some or all changes must be done via trusted path.

e) Identify any appropriate components from the FMT: Security management for initial values for new objects and subjects.

f) Identify any applicable rollback components from the Rollback (FDP_ROL) family.

g) Identify any applicable residual information protection requirements from the Residual information protection (FDP_RIP) family.

h) Identify any applicable import or export components, and how security attributes should be handled during import and export, from the Import from outside TSF control (FDP_ITC) and Export to outside TSF control (FDP_ETC) families.

i) Identify any applicable internal TOE communication components from the Internal TOE transfer (FDP_ITT) family.

j) Identify any requirements for integrity protection of stored information from the Stored data integrity (FDP_SDI).

k) Identify any applicable inter-TSF communication components from the Inter-TSF user data confidentiality transfer protection (FDP_UCT) or Inter-TSF user data integrity transfer protection (FDP_UIT) families.

729 Figure 26 shows the decomposition of this class into its constituent components.

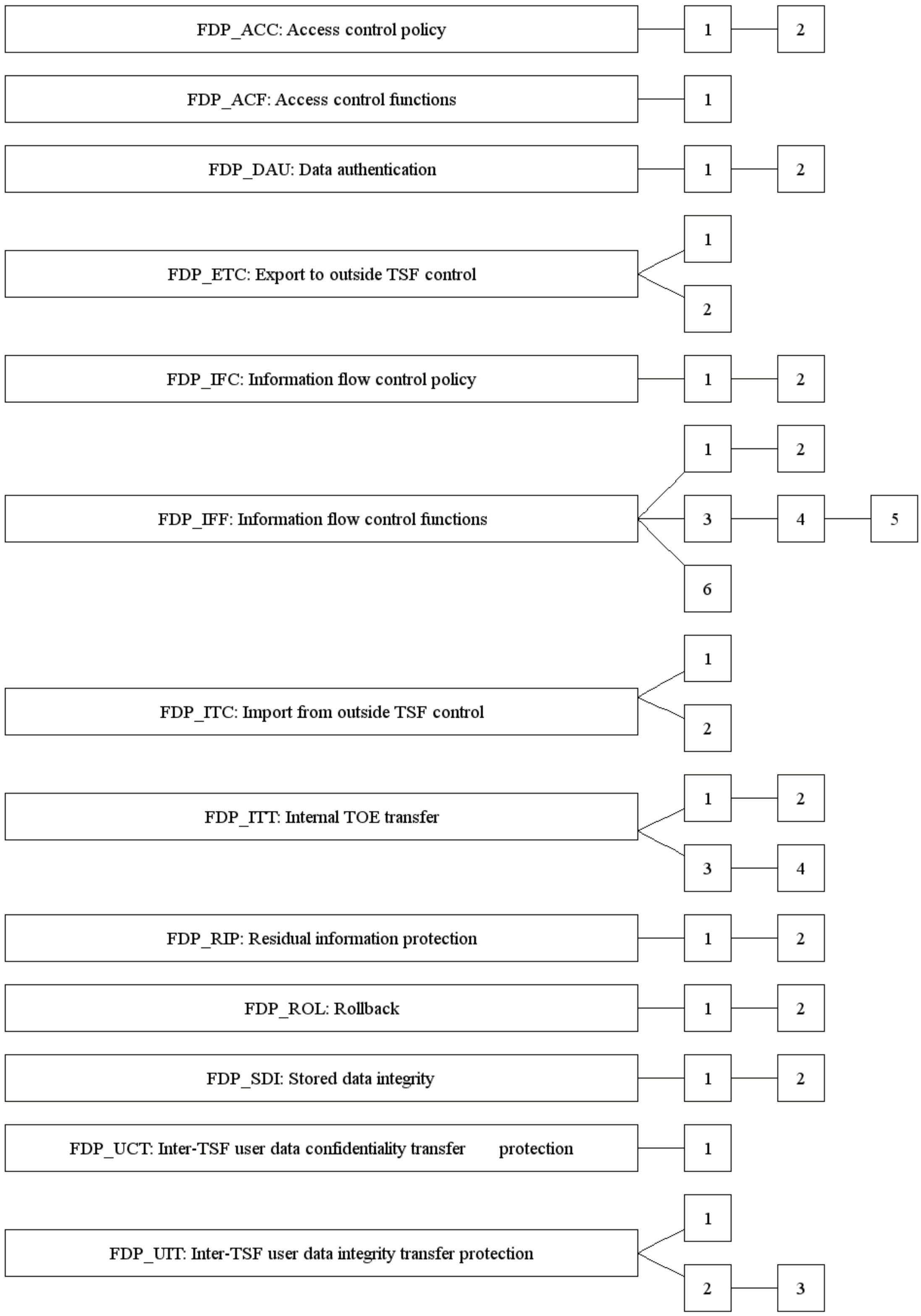



**Figure 26 - FDP: User data protection class decomposition**

## F.1 Access control policy (FDP_ACC)

### User notes

730 This family is based upon the concept of arbitrary controls on the interaction of subjects and objects. The scope and purpose of the controls is based upon the attributes of the accessor (subject), the attributes of the container being accessed (object), the actions (operations) and any associated access control rules.

731 The components in this family are capable of identifying the access control SFPs (by name) to be enforced by the traditional Discretionary Access Control (DAC) mechanisms. It further defines the subjects, objects and operations that are covered by identified access control SFPs. The rules that define the functionality of an access control SFP will be defined by other families, such as Access control functions (FDP_ACF) and Residual information protection (FDP_RIP). The names of the access control SFPs defined in Access control policy (FDP_ACC) are meant to be used throughout the remainder of the functional components that have an operation that calls for an assignment or selection of an "access control SFP."

732 The access control SFP covers a set of triplets: subject, object, and operations. Therefore a subject can be covered by multiple access control SFPs but only with respect to a different operation or a different object. Of course the same applies to objects and operations.

733 A critical aspect of an access control function that enforces an access control SFP is the ability for users to modify the attributes involved in access control decisions. The Access control policy (FDP_ACC) family does not address these aspects. Some of these requirements are left undefined, but can be added as refinements, while others are covered elsewhere in other families and classes such as FMT: Security management.

734 There are no audit requirements in Access control policy (FDP_ACC) as this family specifies access control SFP requirements. Audit requirements will be found in families specifying functions to satisfy the access control SFPs identified in this family.

735 This family provides a PP/ST author the capability to specify several policies, for example, a fixed access control SFP to be applied to one scope of control, and a flexible access control SFP to be defined for a different scope of control. To specify more than one access control policy, the components from this family can be iterated multiple times in a PP/ST to different subsets of operations and objects. This will accommodate TOEs that contain multiple policies, each addressing a particular set of operations and objects. In other words, the PP/ST author should specify the required information in the ACC component for each of the access control SFPs that the TSF will enforce. For example, a TOE incorporating three access control SFPs, each covering only a subset of the objects, subjects, and operations within the TOE, will contain one FDP_ACC.1 Subset access control

component for each of the three access control SFPs, necessitating a total of three FDP_ACC.1 Subset access control components.

## FDP_ACC.1 Subset access control

### User application notes

736 The terms object and subject refer to generic elements in the TOE. For a policy to be implementable, the entities must be clearly identified. For a PP, the objects and operations might be expressed as types such as: named objects, data repositories, observe accesses, etc. For a specific system these generic terms (subject, object) must be refined, e.g. files, registers, ports, daemons, open calls, etc.

737 This component specifies that the policy cover some well-defined set of operations on some subset of the objects. It places no constraints on any operations outside the set - including operations on objects for which other operations are controlled.

### Operations

#### Assignment:

738 In **FDP_ACC.1.1**, the PP/ST author should specify a uniquely named access control SFP to be enforced by the TSF.

739 In **FDP_ACC.1.1**, the PP/ST author should specify the list of subjects, objects, and operations among subjects and objects covered by the SFP.

## FDP_ACC.2 Complete access control

### User application notes

740 This component requires that all possible operations on objects, that are included in the SFP, are covered by an access control SFP.

741 The PP/ST author must demonstrate that each combination of objects and subjects is covered by an access control SFP.

### Operations

#### Assignment:

742 In **FDP_ACC.2.1**, the PP/ST author should specify a uniquely named access control SFP to be enforced by the TSF.

743 In **FDP_ACC.2.1**, the PP/ST author should specify the list of subjects and objects covered by the SFP. All operations among those subjects and objects will be covered by the SFP.

## F.2 Access control functions (FDP_ACF)

### User notes

744 This family describes the rules for the specific functions that can implement an access control policy named in Access control policy (FDP_ACC) which also specifies the scope of control of the policy.

745 This family provides a PP/ST author the capability to describe the rules for access control. This results in a system where the access to objects will not change. An example of such an object is “Message of the Day”, which is readable by all, and changeable only by the authorised administrator. This family also provides the PP/ST author with the ability to describe rules that provide for exceptions to the general access control rules. Such exceptions would either explicitly allow or deny authorisation to access an object.

746 There are no explicit components to specify other possible functions such as two-person control, sequence rules for operations, or exclusion controls. However, these mechanisms, as well as traditional DAC mechanisms, can be represented with the existing components, by careful drafting of the access control rules.

747 A variety of acceptable access control SFs may be specified in this family such as:

- Access control lists (ACLs)
- Time-based access control specifications
- Origin-based access control specifications
- Owner-controlled access control attributes

### FDP_ACF.1 Security attribute based access control

#### User application notes

748 This component provides requirements for a mechanism that mediates access control based on security attributes associated with subjects and objects. Each object and subject has a set of associated attributes, such as location, time of creation, access rights (e.g., Access Control Lists (ACLs)). This component allows the PP/ST author to specify the attributes that will be used for the access control mediation. This component allows access control rules, using these attributes, to be specified.

749 Examples of the attributes that a PP/ST author might assign are presented in the following paragraphs.

750 An identity attribute may be associated with users, subjects, or objects to be used for mediation. Examples of such attributes might be the name of the

program image used in the creation of the subject, or a security attribute assigned to the program image.

751 A time attribute can be used to specify that access will be authorised during certain times of the day, during certain days of the week, or during a certain calendar year.

752 A location attribute could specify whether the location is the location of the request for the operation, the location where the operation will be carried out, or both. It could be based upon internal tables to translate the logical interfaces of the TSF into locations such as through terminal locations, CPU locations, etc.

753 A grouping attribute allows a single group of users to be associated with an operation for the purposes of access control. If required, the refinement operation should be used to specify the maximum number of definable groups, the maximum membership of a group, and the maximum number of groups to which a user can concurrently be associated.

754 This component also provides requirements for the access control security functions to be able to explicitly authorise or deny access to an object based upon security attributes. This could be used to provide privilege, access rights, or access authorisations within the TOE. Such privileges, rights, or authorisations could apply to users, subjects (representing users or applications), and objects.

### Operations

#### Assignment:

755 In **FDP_ACF.1.1**, the PP/ST author should specify an access control SFP name that the TSF is to enforce. The name of the access control SFP, and the scope of control for that policy are defined in components from Access control policy (FDP_ACC).

756 In **FDP_ACF.1.1**, the PP/ST author should specify, for each controlled subject and object, the security attributes and/or named groups of security attributes that the function will use in the specification of the rules. For example, such attributes may be things such as the user identity, subject identity, role, time of day, location, ACLs, or any other attribute specified by the PP/ST author. Named groups of security attributes can be specified to provide a convenient means to refer to multiple security attributes. Named groups could provide a useful way to associate “roles” defined in Security management roles (FMT_SMR), and all of their relevant attributes, with subjects. In other words, each role could relate to a named group of attributes.

757 In **FDP_ACF.1.2**, the PP/ST author should specify the SFP rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects. These rules specify when access is granted or denied. It can specify general access

control functions (e.g. typical permission bits) or granular access control functions (e.g. ACLs).

758 In **FDP_ACF.1.3**, the PP/ST author should specify the rules, based on security attributes, that explicitly authorise access of subjects to objects that will be used to explicitly authorise access. These rules are in addition to those specified in **FDP_ACF.1.1**. They are included in **FDP_ACF.1.3** as they are intended to contain exceptions to the rules in **FDP_ACF.1.1**. An example of rules to explicitly authorise access is based on a privilege vector associated with a subject that always grants access to objects covered by the access control SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify "none".

759 In **FDP_ACF.1.4**, the PP/ST author should specify the rules, based on security attributes, that explicitly deny access of subjects to objects. These rules are in addition to those specified in **FDP_ACF.1.1**. They are included in **FDP_ACF.1.4** as they are intended to contain exceptions to the rules in **FDP_ACF.1.1**. An example of rules to explicitly deny access is based on a privilege vector associated with a subject that always denies access to objects covered by the access control SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify "none".

## F.3 Data authentication (FDP_DAU)

### User notes

760 This family describes specific functions that can be used to authenticate "static" data.

761 Components in this family are to be used when there is a requirement for "static" data authentication, i.e. where data is to be signed but not transmitted. (Note that the Non-repudiation of origin (FCO_NRO) family provides for non-repudiation of origin of information received during a data exchange.)

### FDP_DAU.1 Basic Data Authentication

#### User application notes

762 This component may be satisfied by one-way hash functions (cryptographic checksum, fingerprint, message digest), to generate a hash value for a definitive document that may be used as verification of the validity or authenticity of its information content.

#### Operations

##### Assignment:

763 In **FDP_DAU.1.1**, the PP/ST author should specify the list of objects or information types for which the TSF shall be capable of generating data authentication evidence.

764 In **FDP_DAU.1.2**, the PP/ST author should specify the list of subjects that will have the ability to verify data authentication evidence for the objects identified in the previous element. The list of subjects could be very specific, if the subjects are known, or it could be more generic and refer to a "type" of subject such as an identified role.

### FDP_DAU.2 Data Authentication with Identity of Guarantor

#### User application notes

765 This component additionally requires the ability to verify the identity of the user that provided the guarantee of authenticity (e.g. a trusted third party).

#### Operations

##### Assignment:

766 In **FDP_DAU.2.1**, the PP/ST author should specify the list of objects or information types for which the TSF shall be capable of generating data authentication evidence.

767 In **FDP_DAU.2.2**, the PP/ST author should specify the list of subjects that will have the ability to verify data authentication evidence for the objects identified in the previous element as well as the identity of the user that created the data authentication evidence.

## F.4 Export to outside TSF control (FDP_ETC)

#### User notes

768 This family defines functions for exporting user data from the TOE such that its security attributes either can be explicitly preserved or can be ignored once it has been exported. Consistency of these security attributes are addressed by Inter-TSF TSF data consistency (FPT_TDC).

769 Export to outside TSF control (FDP_ETC) is concerned with limitations on export and association of security attributes with the exported user data.

770 This family, and the corresponding Import family Import from outside TSF control (FDP_ITC), address how the TOE deals with user data transferred into and outside its control. In principle this family is concerned with the export of user data and its related security attributes.

771 A variety of activities might be involved here:

a) exporting of user data without any security attributes;

b) exporting user data including security attributes where the two are associated with one another and the security attributes unambiguously represent the exported user data.

772 If there are multiple SFPs (access control and/or information flow control) then it may be appropriate to iterate these components once for each named SFP.

## FDP_ETC.1 Export of user data without security attributes

### User application notes

773 This component is used to specify the export of user data without the export of its security attributes.

### Operations

#### Assignment:

774 In **FDP_ETC.1.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced when exporting user data. The user data that this function exports is scoped by the assignment of these SFPs.

## FDP_ETC.2 Export of user data with security attributes

### User application notes

775 The user data is exported together with its security attributes. The security attributes are unambiguously associated with the user data. There are several ways of achieving this association. One way that this can be achieved is by physically collocating the user data and the security attributes (e.g. the same floppy), or by using cryptographic techniques such as secure signatures to associate the attributes and the user data. Inter-TSF trusted channel (FTP_ITC) could be used to assure that the attributes are correctly received at the other trusted IT product while Inter-TSF TSF data consistency (FPT_TDC) can be used to make sure that those attributes are properly interpreted. Furthermore, Trusted path (FTP_TRP) could be used to make sure that the export is being initiated by the proper user.

### Operations

#### Assignment:

776 In **FDP_ETC.2.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced when exporting user data. The user data that this function exports is scoped by the assignment of these SFPs.

777 In **FDP_ETC.2.4**, the PP/ST author should specify any additional exportation control rules or “none” if there are no additional exportation control rules. These rules will be enforced by the TSF in addition to the access control SFPs and/or information flow control SFPs selected in **FDP_ETC.2.1**.

## F.5 Information flow control policy (FDP_IFC)

### User notes

778 This family covers the identification of information flow control SFPs; and, for each, specifies the scope of control of the SFP.

779 Examples of security policies that might satisfy this objective are:

- Bell and La Padula Security model [BL];
- Biba Integrity model [BIBA];
- Non-Interference [GOGU1], [GOGU2].

780 The components in this family are capable of identifying the information flow control SFPs to be enforced by the traditional Mandatory Access Control mechanisms that would be found in a TOE. However, they go beyond just the traditional MAC mechanisms and can be used to identify and describe non-interference policies and state-transitions. It further defines the subjects under control of the policy, the information under control of the policy, and operations which cause controlled information to flow to and from controlled subjects for each information flow control SFP in the TOE. The functionality that defines the rules of an information flow control SFP will be defined by other families such as Information flow control functions (FDP_IFF) and Residual information protection (FDP_RIP). The information flow control SFPs named here in Information flow control policy (FDP_IFC) are meant to be used throughout the remainder of the functional components that have an operation that calls for an assignment or selection of an “information flow control SFP.”

781 These components are quite flexible. They allow the domain of flow control to be specified and there is no requirement that the mechanism be based upon labels. The different elements of the information flow control components also permit different degrees of exception to the policy.

782 Each SFP covers a set of triplets: subject, information, and operations that cause information to flow to and from subjects. Some information flow control policies may be at a very low level of detail and explicitly describe subjects in terms of processes within an operating system. Other information flow control policies may be at a high level and describe subjects in the generic sense of users or input/output channels. If the information flow control policy is at too high a level of detail, it may not clearly define the desired IT security functions. In such cases, it is more appropriate to include such descriptions of information flow control policies as objectives. Then the

desired IT security functions can be specified as supportive of those objectives.

783 In the second component (FDP_IFC.2 Complete information flow control), each information flow control SFP will cover all possible operations that cause information covered by that SFP to flow to and from subjects covered by that SFP. Furthermore, all information flows will need to be covered by a SFP. Therefore for each action that causes information to flow, there will be a set of rules that define whether the action is allowed. If there are multiple SFPs that are applicable for a given information flow, all involved SFPs must allow this flow before it is permitted to take place.

784 An information flow control SFP covers a well-defined set of operations. The SFPs coverage may be “complete” with respect to some information flows, or it may address only some of the operations that affect the information flow.

785 An access control SFP controls access to the objects that contain information. An information flow control SFP controls access to the information, independent of its container. The attributes of the information, which may be associated with the attributes of the container (or may not, as in the case of a multi-level database) stay with the information as it flows. The accessor does not have the ability, in the absence of an explicit authorisation, to change the attributes of the information.

786 Information flows and operations can be expressed at multiple levels. In the case of a ST, the information flows and operations might be specified at a system-specific level: TCP/IP packets flowing through a firewall based upon known IP addresses. For a PP, the information flows and operations might be expressed as types: email, data repositories, observe accesses, etc.

787 The components in this family can be applied multiple times in a PP/ST to different subsets of operations and objects. This will accommodate TOEs that contain multiple policies, each addressing a particular set of objects, subjects, and operations.

## FDP_IFC.1 Subset information flow control

### User application notes

788 This component requires that an information flow control policy apply to a subset of the possible operations in the TOE.

### Operations

#### Assignment:

789 In **FDP_IFC.1.1**, the PP/ST author should specify a uniquely named information flow control SFP to be enforced by the TSF.

790 In **FDP_IFC.1.1**, the PP/ST author should specify the list of subjects, information, and operations which cause controlled information to

flow to and from controlled subjects covered by the SFP. As mentioned above, the list of subjects could be at various levels of detail depending on the needs of the PP/ST author. It could specify users, machines, or processes for example. Information could refer to data such as email or network protocols, or more specific objects similar to those specified under an access control policy. If the information that is specified is contained within an object that is subject to an access control policy, then both the access control policy and information flow control policy must be enforced before the specified information could flow to or from the object.

### FDP_IFC.2 Complete information flow control

#### User application notes

791 This component requires that all possible operations that cause information to flow to and from subjects included in the SFP, are covered by an information flow control SFP.

792 The PP/ST author must demonstrate that each combination of information flows and subjects is covered by an information flow control SFP.

#### Operations

##### Assignment:

793 In **FDP_IFC.2.1**, the PP/ST author should specify a uniquely named information flow control SFP to be enforced by the TSF.

794 In **FDP_IFC.2.1**, the PP/ST author should specify the list of subjects and information that will be covered by the SFP. All operations that cause that information to flow to and from subjects will be covered by the SFP. As mentioned above, the list of subjects could be at various levels of detail depending on the needs of the PP/ST author. It could specify users, machines, or processes for example. Information could refer to data such as email or network protocols, or more specific objects similar to those specified under an access control policy. If the information that is specified is contained within an object that is subject to an access control policy, then both the access control policy and information flow control policy must be enforced before the specified information could flow to or from the object.

## F.6 Information flow control functions (FDP_IFF)

#### User notes

795 This family describes the rules for the specific functions that can implement the information flow control SFPs named in Information flow control policy (FDP_IFC), which also specifies the scope of control of the policies. It consists of two “trees:” one addressing the common information flow control function issues, and a second addressing illicit information flows (i.e. covert

channels) with respect to one or more information flow control SFPs. This division arises because the issues concerning illicit information flows are, in some sense, orthogonal to the rest of an SFP. Illicit information flows are flows in violation of policy; thus they are not a policy issue.

796 In order to implement strong protection against disclosure or modification in the face of untrusted software, controls on information flow are required. Access controls alone are not sufficient because they only control access to containers, allowing the information they contain to flow, without controls, throughout a system.

797 In this family, the phrase “types of illicit information flows” is used. This phrase may be used to refer to the categorisation of flows as “Storage Channels” or “Timing Channels”, or it can refer to improved categorisations reflective of the needs of a PP/ST author.

798 The flexibility of these components allows the definition of a privilege policy within FDP_IFF.1 Simple security attributes and FDP_IFF.2 Hierarchical security attributes to allow the controlled bypass of all or part of a particular SFP. If there is a need for a predefined approach to SFP bypass, the PP/ST author should consider incorporating a privilege policy.

## FDP_IFF.1 Simple security attributes

### User application notes

799 This component requires security attributes on information, and on subjects that cause that information to flow and subjects that act as recipients of that information. The attributes of the containers of the information should also be considered if it is desired that they should play a part in information flow control decisions or if they are covered by an access control policy. This component specifies the key rules that are enforced, and describes how security attributes are derived. For example, this component should be used when at least one of the information flow control SFPs in the TSP is based on labels as defined in the Bell and LaPadula security policy model [BL], but these security attributes do not form a hierarchy.

800 This component does not specify the details of how a security attribute is assigned (i.e. user versus process). Flexibility in policy is provided by having assignments that allow specification of additional policy and function requirements, as necessary.

801 This component also provides requirements for the information flow control functions to be able to explicitly authorise and deny an information flow based upon security attributes. This could be used to implement a privilege policy that covers exceptions to the basic policy defined in this component.

### Operations

#### Assignment:

802 In **FDP_IFF.1.1**, the PP/ST author should specify the information flow control SFPs enforced by the TSF. The name of the information flow control SFP, and the scope of control for that policy are defined in components from Information flow control policy (FDP_IFC).

803 In **FDP_IFF.1.1**, the PP/ST author should specify, for each type of controlled subject and information, the security attributes that are relevant to the specification of the SFP rules. For example, such security attributes may be things such the subject identifier, subject sensitivity label, subject clearance label, information sensitivity label, etc. The types of security attributes should be sufficient to support the environmental needs.

804 In **FDP_IFF.1.2**, the PP/ST author should specify for each operation, the security attribute-based relationship that must hold between subject and information security attributes that the TSF will enforce.

805 In **FDP_IFF.1.3**, the PP/ST author should s pecify any additional information flow control SFP rules that the TSF is to enforce. If there are no additional rules then the PP/ST author should specify “none”.

806 In **FDP_IFF.1.4**, the PP/ST author should specify any additional SFP capabilities that the TSF is to provide. If there are no additional capabilities then the PP/ST author should specify “none”.

807 In **FDP_IFF.1.5**, the PP/ST author should specify the rules, based on security attributes, that explicitly authorise information flows. These rules are in addition to those specified in the preceding elements. They are included in **FDP_IFF.1.5** as they are intended to contain exceptions to the rules in the preceding elements. An example of rules to explicitly authorise information flows is based on a privilege vector associated with a subject that always grants the subject the ability to cause an information flow for information that is covered by the SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify “none”.

808 In **FDP_IFF.1.6**, the PP/ST author should specify the rules, based on security attributes, that explicitly deny information flows. These rules are in addition to those specified in the preceding elements. They are included in **FDP_IFF.1.6** as they are intended to contain exceptions to the rules in the preceding elements. An example of rules to explicitly authorise information flows is based on a privilege vector associated with a subject that always denies the subject the ability to cause an information flow for information that is covered by the SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify “none”.

## FDP_IFF.2 Hierarchical security attributes

### User application notes

809 This component requires that all information flow control SFPs in the TSP use hierarchical security attributes that form a lattice.

810 For example, it should be used when at least one of the information flow control SFPs in the TSP is based on labels as defined in the Bell and LaPadula security policy model [BL] and form a hierarchy.

811 It is important to note that the hierarchical relationship requirements identified in **FDP_IFF.2.5** need only apply to the information flow control security attributes for the information flow control SFPs that have been identified in **FDP_IFF.2.1**. This component is not meant to apply to other SFPs such as access control SFPs.

812 Like the preceding component, this component could also be used to implement a privilege policy that covers rules that allow for the explicit authorisation or denial of information flows.

813 If it is the case that multiple information flow control SFPs are to be specified, and that each of these SFPs will have their own security attributes that are not related to one another, then the PP/ST author should iterate this component once for each of those SFPs. Otherwise a conflict might arise with the sub-items of **FDP_IFF.2.5** since the required relationships will not exist.

### Operations

#### Assignment:

814 In **FDP_IFF.2.1**, the PP/ST author should specify the information flow control SFPs enforced by the TSF. The name of the information flow control SFP, and the scope of control for that policy are defined in components from Information flow control policy (FDP_IFC).

815 In **FDP_IFF.2.1**, the PP/ST author should specify, for each type of controlled subject and information, the security attributes that are relevant to the specification of the SFP rules. For example, such security attributes may be things such the subject identifier, subject sensitivity label, subject clearance label, information sensitivity label, etc. The types of security attributes should be sufficient to support the environmental needs.

816 In **FDP_IFF.2.2**, the PP/ST author should specify for each operation, the security attribute-based relationship that must hold between subject and information security attributes that the TSF will enforce. These relationships should be based upon the ordering relationships between the security attributes.

817 In **FDP_IFF.2.3**, the PP/ST author should specify any additional information flow control SFP rules that the TSF is to enforce. If there are no additional rules then the PP/ST author should specify “none”.

818 In **FDP_IFF.2.4**, the PP/ST author should specify any additional SFP capabilities that the TSF is to enforce. If there are no additional rules then the PP/ST author should specify “none”.

819 In **FDP_IFF.2.5**, the PP/ST author should specify the rules, based on security attributes, that explicitly authorise information flows. These rules are in addition to those specified in the preceding elements. They are included in **FDP_IFF.2.5** as they are intended to contain exceptions to the rules in the preceding elements. An example of rules to explicitly authorise information flows is based on a privilege vector associated with a subject that always grants the subject the ability to cause an information flow for information that is covered by the SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify “none”.

820 In **FDP_IFF.2.6**, the PP/ST author should specify the rules, based on security attributes, that explicitly deny information flows. These rules are in addition to those specified in the preceding elements. They are included in **FDP_IFF.2.6** as they are intended to contain exceptions to the rules in the preceding elements. An example of rules to explicitly authorise information flows is based on a privilege vector associated with a subject that always denies the subject the ability to cause an information flow for information that is covered by the SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify “none”.

## FDP_IFF.3 Limited illicit information flows

### User application notes

821 This component should be used when at least one of the SFPs that requires control of illicit information flows does not require elimination of flows.

822 For the specified illicit information flows, certain maximum capacities should be provided. In addition a PP/ST author has the ability to specify whether the illicit information flows must be audited.

### Operations

#### Assignment:

823 In **FDP_IFF.3.1**, the PP/ST author should specify the information flow control SFPs enforced by the TSF. The name of the information flow control SFP, and the scope of control for that policy are defined in components from Information flow control policy (FDP_IFC).

824 In **FDP_IFF.3.1**, the PP/ST author should specify the types of illicit information flows that are subject to a maximum capacity limitation.

825 In **FDP_IFF.3.1**, the PP/ST author should specify the maximum capacity permitted for any identified illicit information flows.

## FDP_IFF.4 Partial elimination of illicit information flows

### User application notes

826 This component should be used when all the SFPs that requires control of illicit information flows require elimination of some (but not necessarily all) illicit information flows.

### Operations

#### Assignment:

827 In **FDP_IFF.4.1**, the PP/ST author should specify the information flow control SFPs enforced by the TSF. The name of the information flow control SFP, and the scope of control for that policy are defined in components from Information flow control policy (FDP_IFC).

828 In **FDP_IFF.4.1**, the PP/ST author should specify the types of illicit information flows which are subject to a maximum capacity limitation.

829 In **FDP_IFF.4.1**, the PP/ST author should specify the maximum capacity permitted for any identified illicit information flows.

830 In **FDP_IFF.4.2**, the PP/ST author should specify the types of illicit information flows to be eliminated. This list may not be empty as this component requires that some illicit information flows are to be eliminated.

## FDP_IFF.5 No illicit information flows

### User application notes

831 This component should be used when the SFPs that require control of illicit information flows require elimination of all illicit information flows. However, the PP/ST author should carefully consider the potential impact that eliminating all illicit information flows might have on the normal functional operation of the TOE. Many practical applications have shown that there is an indirect relationship between illicit information flows and normal functionality within a TOE and eliminating all illicit information flows may result in less than desired functionality.

#### Operations

##### Assignment:

832 In **FDP_IFF.5.1**, the PP/ST author should specify the information flow control SFP for which illicit information flows are to be eliminated. The name of the information flow control SFP, and the scope of control for that policy are defined in components from Information flow control policy (FDP_IFC).

### FDP_IFF.6 Illicit information flow monitoring

#### User application notes

833 This component should be used when it is desired that the TSF provide the ability to monitor the use of illicit information flows that exceed a specified capacity. If it is desired that such flows be audited, then this component could serve as the source of audit events to be used by components from the Security audit data generation (FAU_GEN) family.

#### Operations

##### Assignment:

834 In **FDP_IFF.6.1**, the PP/ST author should specify the information flow control SFPs enforced by the TSF. The name of the information flow control SFP, and the scope of control for that policy are defined in components from Information flow control policy (FDP_IFC).

835 In **FDP_IFF.6.1**, the PP/ST author should specify the types of illicit information flows that will be monitored for exceeding a maximum capacity.

836 In **FDP_IFF.6.1**, the PP/ST author should specify the maximum capacity above which illicit information flows will be monitored by the TSF.

## F.7 Import from outside TSF control (FDP_ITC)

#### User notes

837 This family defines mechanisms for importing user data from outside the TSC into the TOE such that the user data security attributes can be preserved. Consistency of these security attributes are addressed by Inter-TSF TSF data consistency (FPT_TDC).

838 Import from outside TSF control (FDP_ITC) is concerned with limitations on import, user specification of security attributes, and association of security attributes with the user data.

839 This family, and the corresponding export family Export to outside TSF control (FDP_ETC), address how the TOE deals with user data outside its

control. This family is concerned with assigning and abstraction of the user data security attributes.

840 A variety of activities might be involved here:

a) importing user data from an unformatted medium (e.g. floppy disk, tape, scanner, video or audit signal), without including any security attributes, and physically marking the medium to indicate its contents;

b) importing user data, including security attributes, from a medium and verifying that the object security attributes are appropriate;

c) importing user data, including security attributes, from a medium using a cryptographic sealing technique to protect the association of user data and security attributes.

841 This family is not concerned with the determination of whether the user data may be imported. It is concerned with the values of the security attributes to associate with the imported user data.

842 There are two possibilities for the import of user data: either the user data is unambiguously associated with reliable object security attributes (values and meaning of the security attributes is not modified), or no reliable security attributes (or no security attributes at all) are available from the import source. This family addresses both cases.

843 If there are reliable security attributes available, they may have been associated with the user data by physical means (the security attributes are on the same media), or by logical means (the security attributes are distributed differently, but include unique object identification, e.g. cryptographic checksum).

844 This family is concerned with importing user data and maintaining the association of security attributes as required by the SFP. Other families are concerned with other import aspects such as consistency, trusted channels, and integrity that are beyond the scope of this family. Furthermore, Import from outside TSF control (FDP_ITC) is only concerned with the interface to the import medium. Export to outside TSF control (FDP_ETC) is responsible for the other end point of the medium (the source).

845 Some of the well known import requirements are:

a) importing of user data without any security attributes;

b) importing of user data including security attributes where the two are associated with one another and the security attributes unambiguously represent the information being imported.

846 These import requirements may be handled by the TSF with or without human intervention, depending on the IT limitations and the organisational

security policy. For example, if user data is received on a "confidential" channel, the security attributes of the objects will be set to "confidential".

847 If there are multiple SFPs (access control and/or information flow control) then it may be appropriate to iterate these components once for each named SFP.

## FDP_ITC.1 Import of user data without security attributes

### User application notes

848 This component is used to specify the import of user data that does not have reliable (or any) security attributes associated with it. This function requires that the security attributes for the imported user data be initialised within the TSF. It could also be the case that the PP/ST author specifies the rules for import. It may be appropriate, in some environments, to require that these attributes be supplied via a trusted path or a trusted channel mechanism.

### Operations

#### Assignment:

849 In **FDP_ITC.1.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced when importing user data from outside of the TSC. The user data that this function imports is scoped by the assignment of these SFPs.

850 In **FDP_ITC.1.3**, the PP/ST author should specify any additional importation control rules or "none" if there are no additional importation control rules. These rules will be enforced by the TSF in addition to the access control SFPs and/or information flow control SFPs selected in **FDP_ITC.1.1**.

## FDP_ITC.2 Import of user data with security attributes

### User application notes

851 This component is used to specify the import of user data that has reliable security attributes associated with it. This function relies upon the security attributes that are accurately and unambiguously associated with the objects on the import medium. Once imported, those objects will have those same attributes. This requires Inter-TSF TSF data consistency (FPT_TDC) to ensure the consistency of the data. It could also be the case that the PP/ST author specifies the rules for import.

### Operations

#### Assignment:

852 In **FDP_ITC.2.1**, the PP/ST author should specify the access control SFP and/or information flow control SFP that will be enforced when

importing user data from outside of the TSC. The user data that this function imports is scoped by the assignment of these SFPs.

853 In **FDP_ITC.2.5**, the PP/ST author should specify any additional importation control rules or "none" if there are no additional importation control rules. These rules will be enforced by the TSF in addition to the access control SFPs and/or information flow control SFPs selected in **FDP_ITC.2.1**.

## F.8 Internal TOE transfer (FDP_ITT)

### User notes

854 This family provides requirements that address protection of user data when it is transferred between parts of a TOE across an internal channel. This may be contrasted with the Inter-TSF user data confidentiality transfer protection (FDP_UCT) and Inter-TSF user data integrity transfer protection (FDP_UIT) family, which provide protection for user data when it is transferred between distinct TSFs across an external channel, and Export to outside TSF control (FDP_ETC) and Import from outside TSF control (FDP_ITC), which address transfer of data to or from outside the TSF's Control.

855 The requirements in this family allow a PP/ST author to specify the desired security for user data while in transit within the TOE. This security could be protection against disclosure, modification, or loss of availability.

856 The determination of the degree of physical separation above which this family should apply depends on the intended environment of use. In a hostile environment, there may be risks arising from transfers between parts of the TOE separated by only a system bus. In more benign environments, the transfers may be across more traditional network media.

857 If there are multiple SFPs (access control and/or information flow control) then it may be appropriate to iterate these components once for each named SFP.

### FDP_ITT.1 Basic internal transfer protection

#### Operations

##### Assignment:

858 In **FDP_ITT.1.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) covering the information being transferred.

##### Selection:

859 In **FDP_ITT.1.1**, the PP/ST author should specify the types of transmission errors that the TSF should prevent occuring for user data while in transport. The options are disclosure, modification, loss of use.

## FDP_ITT.2 Transmission separation by attribute

### User application notes

860 This component could, for example, be used to provide different forms of protection to information with different clearance levels.

861 One of the ways to achieve separation of data when it is transmitted is through the use of separate logical or physical channels.

### Operations

#### Assignment:

862 In **FDP_ITT.2.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) covering the information being transferred.

#### Selection:

863 In **FDP_ITT.2.1**, the PP/ST author should specify the types of transmission errors that the TSF should prevent occuring for user data while in transport. The options are disclosure, modification, loss of use.

#### Assignment:

864 In **FDP_ITT.2.2**, the PP/ST author should specify the security attributes, the values of which the TSF will use to determine when to separate data that is being trasmitted between physically-separated parts of the TOE. An example is that user data associated with the identity of one owner is transmitted separately from the user data associated with the identify of a different owner. In this case, the value of the identity of the owner of the data is what is used to determine when to separate the data for transmission.

## FDP_ITT.3 Integrity monitoring

### User application notes

865 This component is used in combination with either FDP_ITT.1 Basic internal transfer protection or FDP_ITT.2 Transmission separation by attribute. It ensures that the TSF checks received user data (and their attributes) for integrity. FDP_ITT.1 Basic internal transfer protection or FDP_ITT.2 Transmission separation by attribute will provide the data in a manner such that it is protected from modification (so that FDP_ITT.3 Integrity monitoring can detect any modifications).

866 The PP/ST author has to specify the types of errors that must be detected. The PP/ST author should consider: modification of data, substitution of data, unrecoverable ordering change of data, replay of data, incomplete data, in addition to other integrity errors.

867 The PP/ST author must specify the actions that the TSF should take on detection of a failure. For example: ignore the user data, request the data again, inform the authorised administrator, reroute traffic for other lines.

#### Operations

##### Assignment:

868 In **FDP_ITT.3.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) covering the information being transferred and monitored for integrity errors.

869 In **FDP_ITT.3.1**, the PP/ST author should specify the type of possible integrity errors to be monitored during transmission of the user data.

870 In **FDP_ITT.3.2**, the PP/ST author should specify the action to be taken by the TSF when an integrity error is encountered. An example might be that the TSF should request the resubmission of the user data. The SFP(s) specified in **FDP_ITT.3.1** will be enforced as the actions are taken by the TSF.

### FDP_ITT.4 Attribute-based integrity monitoring

#### User application notes

871 This component is used in combination with FDP_ITT.2 Transmission separation by attribute. It ensures that the TSF checks received user data, that has been transmitted by separate channels (based on values of specified security attributes), for integrity. It allows the PP/ST author to specify actions to be taken upon detection of an integrity error.

872 For example, this component could be used to provide different integrity error detection and action for information at different integrity levels.

873 The PP/ST author has to specify the types of errors that must be detected. The PP/ST author should consider: modification of data, substitution of data, unrecoverable ordering change of data, replay of data, incomplete data, in addition to other integrity errors.

874 The PP/ST author should specify the attributes (and associated transmission channels) that necessitate integrity error monitoring

875 The PP/ST author must specify the actions that the TSF should take on detection of a failure. For example: ignore the user data, request the data again, inform the authorised administrator, reroute traffic for other lines.

### Operations

#### Assignment:

876 In **FDP_ITT.4.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) covering the information being transferred and monitored for integrity errors.

877 In **FDP_ITT.4.1**, the PP/ST author should specify the type of possible integrity errors to be monitored during transmission of the user data.

878 In **FDP_ITT.4.1**, the PP/ST author should specify a list of security attributes that require separate transmission channels. This list is used to determine which user data to monitor for integrity errors., based on its security attributes and its transmission channel. This element is directly related to FDP_ITT.2 Transmission separation by attribute.

879 In **FDP_ITT.4.2**, the PP/ST author should specify the action to be taken by the TSF when an integrity error is encountered. An example might be that the TSF should request the resubmission of the user data. The SFP(s) specified in **FDP_ITT.3.1** will be enforced as the actions are taken by the TSF.

## F.9 Residual information protection (FDP_RIP)

### User notes

880 This family addresses the need to ensure that deleted information is no longer accessible, and that newly-created objects do not contain information from previously used objects within the TOE. This family does not address objects stored off-line.

881 This family requires protection for information that has been logically deleted or released (not available to the user but still within the system and may be recoverable). In particular, this includes information that is contained in an object, as part of the TSF reusable resources, where destruction of the object does not necessarily equate to destruction of the resource or any contents of the resource.

882 It also applies to resources that are serially reused by different subjects within the system. For example, most operating systems typically rely upon hardware registers (resources) to support processes within the system. As processes are swapped from a “run” state to a “sleep” state (and vice versa), these registers are serially reused by different subjects. While this “swapping” action may not be considered an allocation or deallocation of a resource, Residual information protection (FDP_RIP) could apply to such events and resources.

883 Residual information protection (FDP_RIP) typically controls access to information that is not part of any currently defined or accessible object; however, in certain cases this may not be true. For example, object “A” is a

file and object “B” is the disk upon which that file resides. If object “A” is deleted, the information from object “A” is under the control of Residual information protection (FDP_RIP) even though it is still part of object “B”.

884 It is important to note that Residual information protection (FDP_RIP) applies only to on-line objects and not off-line objects such as those backed-up on tapes. For example, if a file is deleted in the TOE, Residual information protection (FDP_RIP) can be instantiated to require that no residual information exists upon deallocation; however, the TSF cannot extend this enforcement to that same file that exists on the off-line back-up. Therefore that same file is still available. If this is a concern, then the PP/ST author should make sure that the proper environmental objectives are in place to support administrative guidance to address off-line objects.

885 Residual information protection (FDP_RIP) and Rollback (FDP_ROL) can conflict when Residual information protection (FDP_RIP) is instantiated to require that residual information be cleared at the time the application releases the object to the TSF (i.e. upon deallocation). Therefore, the Residual information protection (FDP_RIP) selection of “deallocation” should not be used with Rollback (FDP_ROL) since there would be no information to roll back. The other selection, “unavailability upon allocation”, may be used with Rollback (FDP_ROL), but there is the risk that the resource which held the information has been allocated to a new object before the roll back took place. If that were to occur, then the roll back would not be possible.

886 There are no audit requirements in Residual information protection (FDP_RIP) because this is not a user-invokable function. Auditing of allocated or deallocated resources would be auditable as part of the access control SFP or the information flow control SFP operations.

887 This family should apply to the objects specified in the access control SFP(s) or the information flow control SFP(s) as specified by the PP/ST author.

## FDP_RIP.1 Subset residual information protection

### User application notes

888 This component requires that, for a subset of the objects in the TOE, the TSF will ensure that there is no available residual information contained in a resource allocated to those objects or deallocated from those objects.

### Operations

#### Selection:

889 In **FDP_RIP.1.1**, the PP/ST author should specify the event, allocation of the resource to or deallocation of the resource from, that invokes the residual information protection function.

#### Assignment:

890 In **FDP_RIP.1.1**, the PP/ST author should specify the list of objects subject to residual information protection.

### FDP_RIP.2 Full residual information protection

#### User application notes

891 This component requires that for all objects in the TOE, the TSF will ensure that there is no available residual information contained in a resource allocated to those objects or deallocated from those objects.

#### Operations

#### Selection:

892 In **FDP_RIP.2.1**, the PP/ST author should specify the event, allocation of the resource to or deallocation of the resource from, that invokes the residual information protection function.

## F.10 Rollback (FDP_ROL)

#### User notes

893 This family addresses the need to return to a well defined valid state, such as the need of a user to undo modifications to a file or to undo transactions in case of an incomplete series of transaction as in the case of databases.

894 This family is intended to assist a user in returning to a well defined valid state after the user undoes the last set of actions, or, in distributed databases, the return of all of the distributed copies of the databases to the state before an operation failed.

895 Residual information protection (FDP_RIP) and Rollback (FDP_ROL) conflict when Residual information protection (FDP_RIP) enforces that the contents will be made unavailable at the time that a resource is deallocated from an object. Therefore, this use of Residual information protection (FDP_RIP) cannot be combined with Rollback (FDP_ROL) as there would be no information to roll back. Residual information protection (FDP_RIP) can be used only with Rollback (FDP_ROL) when it enforces that the contents will be unavailable at the time that a resource is allocated to an object. This is because the Rollback (FDP_ROL) mechanism will have an opportunity to access the previous information that may still be present in the TOE in order to successfully roll back the operation.

896 The rollback requirement is bounded by certain limits. For example a text editor typically only allows you roll back up to a certain number of commands. Another example would be backups. If backup tapes are rotated, after a tape is reused, the information can no longer be retrieved. This also poses a bound on the rollback requirement.

## FDP_ROL.1 Basic rollback

### User application notes

897 This component allows a user or subject to undo a set of operations on a predefined set of objects. The undo is only possible within certain limits, for example up to a number of characters or up to a time limit.

### Operations

#### Assignment:

898 In **FDP_ROL.1.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced when performing rollback operations. This is necessary to make sure that roll back is not used to circumvent the specified SFPs.

899 In **FDP_ROL.1.1**, the PP/ST author should specify the list of operations that can be rolled back.

900 In **FDP_ROL.1.1**, the PP/ST author should specify the information and/or list of objects that are subjected to the rollback policy.

901 In **FDP_ROL.1.2**, the PP/ST author should specify the boundary limit to which rollback operations may be performed. The boundary may be specified as a predefined period of time, for example, operations may be undone which were performed within the past two minutes. Other possible boundaries may be defined as the maximum number of operations allowable or the size of a buffer.

## FDP_ROL.2 Advanced rollback

### User application notes

902 This component enforces that the TSF provide the capability to rollback all operations; however, the user can choose to rollback only a part of them.

### Operations

#### Assignment:

903 In **FDP_ROL.2.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced when performing rollback operations. This is necessary to make sure that roll back is not used to circumvent the specified SFPs.

904 In **FDP_ROL.2.1**, the PP/ST author should specify the list of objects that are subjected to the rollback policy.

905 In **FDP_ROL.2.2**, the PP/ST author should specify the boundary limit to which rollback operations may be performed. The boundary may be specified as a predefined period of time, for example, operations may

be undone which were performed within the past two minutes. Other possible boundaries may be defined as the maximum number of operations allowable or the size of a buffer.

## F.11 Stored data integrity (FDP_SDI)

### User notes

906 This family provides requirements that address protection of user data while it is stored within the TSC.

907 Hardware glitches or errors may affect data stored in memory. This family provides requirements to detect these unintentional errors. The integrity of user data while stored on storage devices within the TSC are also addressed by this family.

908 To prevent a subject from modifying the data, the Information flow control functions (FDP_IFF) or Access control functions (FDP_ACF) families are required (rather than this family).

909 This family differs from Internal TOE transfer (FDP_ITT) that protects the user data from integrity errors while being transferred within the TOE.

### FDP_SDI.1 Stored data integrity monitoring

#### User application notes

910 This component monitors data stored on media for integrity errors. The PP/ST author can specify different kinds of user data attributes that will be used as the basis for monitoring.

#### Operations

##### Assignment:

911 In **FDP_SDI.1.1**, the PP/ST author should specify the integrity errors that the TSF will detect.

912 In **FDP_SDI.1.1**, the PP/ST author should specify the user data attributes that will be used as the basis for the monitoring.

### FDP_SDI.2 Stored data integrity monitoring and action

#### User application notes

913 This component monitors data stored on media for integrity errors. The PP/ST author can specify which action should be taken in case an integrity error is detected.

Operations

Assignment:

914 In **FDP_SDI.2.1**, the PP/ST author should specify the integrity errors that the TSF will detect.

915 In **FDP_SDI.2.1**, the PP/ST author should specify the user data attributes that will be used as the basis for the monitoring.

916 In **FDP_SDI.2.2**, the PP/ST author should specify the actions to be taken in case an integrity error is detected.

## F.12 Inter-TSF user data confidentiality transfer protection (FDP_UCT)

User notes

917 This family defines the requirements for ensuring the confidentiality of user data when it is transferred using an external channel between the TOE and another trusted IT product. Confidentiality is enforced by preventing unauthorised disclosure of user data in transit between the two end points. The end points may be a TSF or a user.

918 This family provides a requirement for the protection of user data during transit. In contrast, Confidentiality of exported TSF data (FPT_ITC) handles TSF data.

### FDP_UCT.1 Basic data exchange confidentiality

User application notes

919 The TSF has the ability to protect from disclosure some user data which is exchanged.

Operations

Assignment:

920 In **FDP_UCT.1.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced when exchanging user data. The specified policies will be enforced to make decisions about who can exchange data and which data can be exchanged.

Selection:

921 In **FDP_UCT.1.1**, the PP/ST author should specify whether this element applies to a mechanism that transmits or receives user data.

## F.13 Inter-TSF user data integrity transfer protection (FDP_UIT)

### User notes

922 This family defines the requirements for providing integrity for user data in transit between the TSF and another trusted IT product and recovering from detectable errors. At a minimum, this family monitors the integrity of user data for modifications. Furthermore, this family supports different ways of correcting detected integrity errors.

923 This family defines the requirements for providing integrity for user data in transit; while Integrity of exported TSF data (FPT_ITI) handles TSF data.

924 Inter-TSF user data integrity transfer protection (FDP_UIT) and Inter-TSF user data confidentiality transfer protection (FDP_UCT) are duals of each other, as Inter-TSF user data confidentiality transfer protection (FDP_UCT) addresses user data confidentiality. Therefore, the same mechanism that implements Inter-TSF user data integrity transfer protection (FDP_UIT) could possibly be used to implement other families such as Inter-TSF user data confidentiality transfer protection (FDP_UCT) and Import from outside TSF control (FDP_ITC).

### FDP_UIT.1 Data exchange integrity

#### User application notes

925 The TSF has a basic ability to send or receive user data in a manner such that modification of the user data can be detected. There is no requirement for a TSF mechanism to attempt to recover from the modification.

#### Operations

##### Assignment:

926 In **FDP_UIT.1.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced on the transmitted data or on the received data. The specified policies will be enforced to make decisions about who can transmit or who can receive data, and which data can be transmitted or received.

##### Selection:

927 In **FDP_UIT.1.1**, the PP/ST author should specify whether this element applies to a TSF that is transmitting or receiving objects.

928 In **FDP_UIT.1.1**, the PP/ST author should specify whether the data should be protected from modification, deletion, insertion or replay.

929 In **FDP_UIT.1.2**, the PP/ST author should specify whether the errors of the type: modification, deletion, insertion or replay are detected.

## FDP_UIT.2 Source data exchange recovery

### User application notes

930 This component provides the ability to recover from a set of identified transmission errors, if required, with the help of the other trusted IT product. As the other trusted IT product is outside the TSC, the TSF cannot control its behaviour. However, it can provide functions that have the ability to cooperate with the other trusted IT product for the purposes of recovery. For example, the TSF could include functions that depend upon the source trusted IT product to re-send the data in the event that an error is detected. This component deals with the ability of the TSF to handle such an error recovery.

### Operations

#### Assignment:

931 In **FDP_UIT.2.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced when recovering user data. The specified policies will be enforced to make decisions about which data can be recovered and how it can be recovered.

932 In **FDP_UIT.2.1**, the PP/ST author should specify the list of integrity errors from which the TSF, with the help of the source trusted IT product, is be able to recover the original user data.

## FDP_UIT.3 Destination data exchange recovery

### User application notes

933 This component provides the ability to recover from a set of identified transmission errors. It accomplishes this task without help from the source trusted IT product. For example, if certain errors are detected, the transmission protocol must be robust enough to allow the TSF to recover from the error based on checksums and other information available within that protocol.

### Operations

#### Assignment:

934 In **FDP_UIT.3.1**, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced when recovering user data. The specified policies will be enforced to make decisions about which data can be recovered and how it can be recovered.

935 In **FDP_UIT.3.1**, the PP/ST author should specify the list of integrity errors from which the receiving TSF, alone, is able to recover the original user data.

# G Class FIA: Identification and authentication (normative)

936 A common security requirement is to unambiguously identify the person and/or entity performing functions in a TOE. This involves not only establishing the claimed identity of each user, but also verifying that each user is indeed who he/she claims to be. This is achieved by requiring users to provide the TSF with some information that is known by the TSF to be associated with the user in question.

937 Families in this class address the requirements for functions to establish and verify a claimed user identity. Identification and Authentication is required to ensure that users are associated with the proper security attributes (e.g. identity, groups, roles, security or integrity levels).

938 The unambiguous identification of authorised users and the correct association of security attributes with users and subjects is critical to the enforcement of the security policies.

939 The User identification (FIA_UID) family addresses determining the identity of a user.

940 The User authentication (FIA_UAU) family addresses verifying the identity of a user.

941 The Authentication failures (FIA_AFL) family addresses defining limits on repeated unsuccessful authentication attempts.

942 The User attribute definition (FIA_ATD) family address the definition of user attributes that are used in the enforcement of the TSP.

943 The User-subject binding (FIA_USB) family addresses the correct association of security attributes for each authorised user.

944 The Specification of secrets (FIA_SOS) family addresses the generation and verification of secrets that satisfy a defined metric.

945 Figure 27 shows the decomposition of this class into its constituent components.

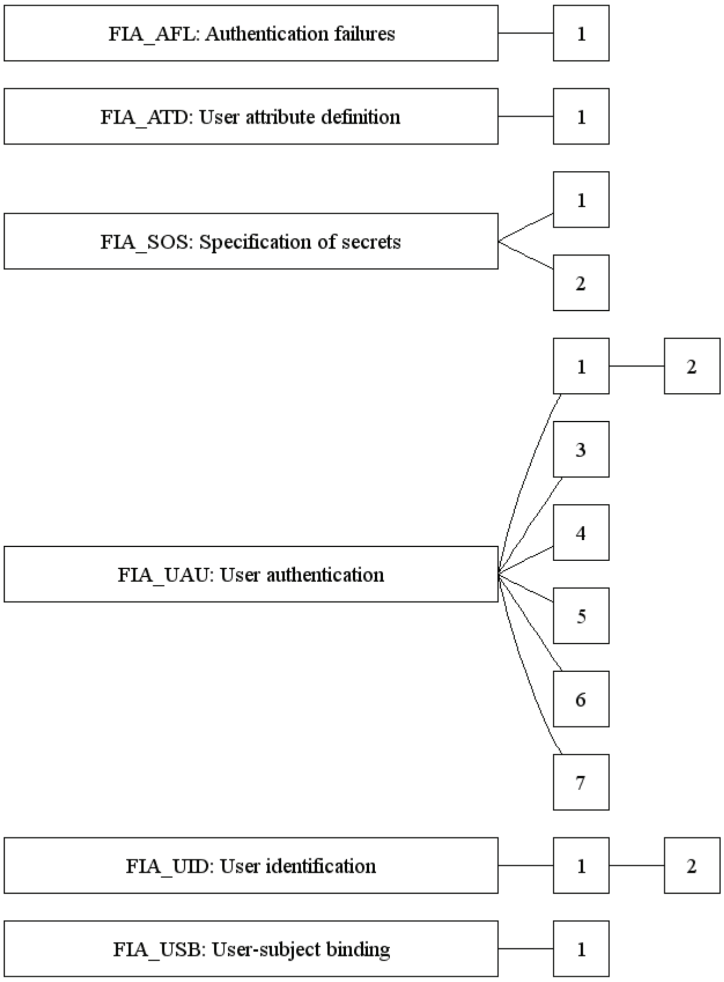



**Figure 27 - FIA: Identification and authentication class decomposition**

# G.1 Authentication failures (FIA_AFL)

## User notes

946 This family addresses requirements for defining values for authentication attempts and TSF actions in cases of authentication attempt failure. Parameters include, but are not limited to, the number of attempts and time thresholds.

947 The session establishment process is the interaction with the user to perform the session establishment independent of the actual implementation. If the number of unsuccessful authentication attempts exceeds the indicated threshold, either the user account or the terminal (or both) will be locked. If the user account is disabled, the user cannot log-on to the system. If the terminal is disabled, the terminal (or the address that the terminal has) cannot be used for any log-on. Both of these situations continue until the condition for re-establishment is satisfied.

## FIA_AFL.1 Authentication failure handling

### User application notes

948 The PP/ST author may define the number of unsuccessful authentication attempts or may choose to let the TOE developer or the authorised user to define this number. The unsuccessful authentication attempts need not be consecutive, but rather related to an authentication event. Such an authentication event could be the count from the last successful session establishment at a given terminal.

949 The PP/ST author could specify a list of actions that the TSF shall take in the case of authentication failure. An authorised administrator could also be allowed to manage the events, if deemed opportune by the PP/ST author. These actions could be, among other things, terminal deactivation, user account deactivation, or administrator alarm. The conditions under which the situation will be restored to normal must be specified on the action.

950 In order to prevent denial of service, TOEs usually ensure that there is at least one user account that cannot be disabled.

951 Further actions for the TSF can be stated by the PP/ST author, including rules for re-enabling the user session establishment process, or sending an alarm to the administrator. Examples of these actions are: until a specified time has lapsed, until the authorised administrator re-enables the terminal/account, a time related to failed previous attempts (every time the attempt fails, the disabling time is doubled).

### Operations

#### Selection:

952 In **FIA_AFL.1.1**, the PP/ST author should select either the assignment of a positive integer, or the phrase “an administrator configurable positive integer” specifying the range of acceptable values.

#### Assignment:

953 In **FIA_AFL.1.1**, the PP/ST author should specify the authentication events. Examples of these authentication events are: the unsuccessful authentication attempts since the last successful authentication for the indicated user identity, the unsuccessful authentication attempts since the last successful authentication for the current terminal, the number of unsuccessful authentication attempts in the last 10 minutes. At least one authentication event must be specified.

954 In **FIA_AFL.1.1**, if the assignment of a positive integer is selected, the PP/ST author should specify the default number (positive integer) of unsuccessful authentication attempts that, when met or surpassed, will trigger the events.

955 In **FIA_AFL.1.1**, if an administrator configurable positive integer is selected, the PP/ST author should specify the range of acceptable values from which the administrator of the TOE may configure the number of unsuccessful authentication attempts. The number of authentication attempts should be less than or equal to the upper bound and greater or equal to the lower bound values.

956 In **FIA_AFL.1.2**, the PP/ST author should specify the actions to be taken in case the threshold is met or surpassed. These actions could be disabling of an account for 5 minutes, disabling the terminal for an increasing amount of time (2 to the power of the number of unsuccessful attempts in seconds), or disabling of the account until unlocked by the administrator and simultaneously informing the administrator. The actions should specify the measures and if applicable the duration of the measure (or the conditions under which the measure will be ended).

## G.2 User attribute definition (FIA_ATD)

### User notes

957 All authorised users may have a set of security attributes, other than the user's identity, that are used to enforce the TSP. This family defines the requirements for associating user security attributes with users as needed to support the TSP.

958 There are dependencies on the individual security policy definitions. These individual definitions should contain the listing of attributes that are necessary for policy enforcement.

### FIA_ATD.1 User attribute definition

#### User application notes

959 This component specifies the security attributes that should be maintained at the level of the user. This means that the security attributes listed are assigned to and can be changed at the level of the user. In other words, changing a security attribute in this list associated with a user should have no impact on the security attributes of any other user.

960 In case security attributes belong to a group of users (such as Capability List for a group), the user will need to have a reference (as security attribute) to the relevant group.

#### Operations

##### Assignment:

961 In **FIA_ATD.1.1**, the PP/ST author should specify the security attributes that are associated to an individual user. An example of such a list is {“clearance”, “group identifier”, “rights”}.

## G.3 Specification of secrets (FIA_SOS)

### User notes

962 This family defines requirements for mechanisms that enforce defined quality metrics on provided secrets, and generate secrets to satisfy the defined metric. Examples of such mechanisms may include automated checking of user supplied passwords, or automated password generation.

963 A secret can be generated outside the TOE (e.g. selected by the user and introduced in the system). In such cases, the FIA_SOS.1 Verification of secrets component can be used to ensure that the external generated secret adheres to certain standards, for example a minimum size, not present in a dictionary, and/or not previously used.

964 Secrets can also be generated by the TOE. In those cases, the FIA_SOS.2 TSF Generation of secrets component can be used to require the TOE to ensure that the secrets that will adhere to some specified metrics.

965 Secrets contain the authentication data provided by the user for an authentication mechanism that is based on knowledge the user possesses. When cryptographic keys are employed, the class FCS: Cryptographic support should be used instead of this family.

### FIA_SOS.1 Verification of secrets

#### User application notes

966 Secrets can be generated by the user. This component ensures that those user generated secrets can be verified to meet a certain quality metric.

#### Operations

##### Assignment:

967 In FIA_SOS.1.1, the PP/ST author should provide a defined quality metric. The quality metric specification can be as simple as a description of the quality checks to be performed, or as formal as a reference to a government published standard that defines the quality metrics that secrets must meet. Examples of quality metrics could include a description of the alphanumeric structure of acceptable secrets and/or the space size that acceptable secrets must meet.

### FIA_SOS.2 TSF Generation of secrets

#### User application notes

968 This component allows the TSF to generate secrets for specific functions such as authentication by means of passwords.

969 When a pseudo-random number generator is used in a secret generation algorithm, it should accept as input random data that would provide output

that has a high degree of unpredictability. This random data (seed) can be derived from a number of available parameters such as a system clock, system registers, date, time, etc. The parameters should be selected to ensure that the number of unique seeds that can be generated from these inputs should be at least equal to the minimum number of secrets that must be generated.

#### Operations

##### Assignment:

970 In **FIA_SOS.2.1**, the PP/ST author should provide a defined quality metric. The quality metric specification can be as simple as a description of the quality checks to be performed or as formal as a reference to a government published standard that defines the quality metrics that secrets must meet. Examples of quality metrics could include a description of the alphanumeric structure of acceptable secrets and/or the space size that acceptable secrets must meet.

971 In **FIA_SOS.2.2**, the PP/ST author should provide a list of TSF functions for which the TSF generated secrets must be used. An example of such a function could include a password based authentication mechanism.

## G.4 User authentication (FIA_UAU)

#### User notes

972 This family defines the types of user authentication mechanisms supported by the TSF. This family defines the required attributes on which the user authentication mechanisms must be based.

### FIA_UAU.1 Timing of authentication

#### User application notes

973 This component requires that the PP/ST author define the TSF-mediated actions that can be performed by the TSF on behalf of the user before the claimed identity of the user is authenticated. The TSF-mediated actions should have no security concerns with users incorrectly identifying themselves prior to being authenticated. For all other TSF-mediated actions not in the list, the user must be authenticated before the action can be performed by the TSF on behalf of the user.

974 This component cannot control whether the actions can also be performed before the identification took place. This requires the use of either FIA_UID.1 Timing of identification and FIA_UID.2 User identification before any action with the appropriate assignments.

#### Operations

##### Assignment:

975 In **FIA_UAU.1.1**, the PP/ST author should specify a list of TSF-mediated actions that can be performed by the TSF on behalf of a user before the claimed identity of the user is authenticated. This list cannot be empty. If no actions are appropriate, component FIA_UAU.2 User authentication before any action should be used instead. An example of such an action might include the request for help on the login procedure.

### FIA_UAU.2 User authentication before any action

#### User application notes

976 This component requires that a user is authenticated before any TSF-mediated action can take place on behalf of that user.

### FIA_UAU.3 Unforgeable authentication

#### User application notes

977 This component addresses requirements for mechanisms that provide protection of authentication data. Authentication data that is copied from another user, or is in some way constructed should be detected and/or rejected. These mechanisms provide confidence that users authenticated by the TSF are actually who they claim to be.

978 This component may be useful only with authentication mechanisms that are based on authentication data that cannot be shared (e.g. biometrics). It is impossible for a TSF to detect or prevent the sharing of passwords outside the control of the TSF.

#### Operations

##### Selection:

979 In **FIA_UAU.3.1**, the PP/ST author should specify whether the TSF will detect, prevent, or detect and prevent forging of authentication data.

980 In **FIA_UAU.3.2**, the PP/ST author should specify whether the TSF will detect, prevent, or detect and prevent copying of authentication data.

### FIA_UAU.4 Single-use authentication mechanisms

#### User application notes

981 This component addresses requirements for authentication mechanisms based on single-use authentication data. Single-use authentication data can be something the user has or knows, but not something the user is. Examples

of single-use authentication data include single-use passwords, encrypted time-stamps, and/or random numbers from a secret lookup table.

982 The PP/ST author can specify to which authentication mechanism(s) this requirement applies.

### Operations

#### Assignment:

983 In **FIA_UAU.4.1**, the PP/ST author should specify the list of authentication mechanisms to which this requirement applies. This assignment can be "all authentication mechanisms". An example of this assignment could be "the authentication mechanism employed to authenticate people on the external network".

## FIA_UAU.5 Multiple authentication mechanisms

### User application notes

984 The use of this component allows specification of requirements for more than one authentication mechanism to be used within a TOE. For each distinct mechanism, applicable requirements must be chosen from the FIA: Identification and authentication class to be applied to each mechanism. It is possible that the same component could be selected multiple times in order to reflect different requirements for the different use of the authentication mechanism.

985 The management functions in the class FMT may provide maintenance capabilities for the set of authentication mechanisms, as well as the rules that determine whether the authentication was successful.

986 To allow anonymous users to be on the system, a "none" authentication mechanism can be incorporated. The use of such access should be clearly explained in the rules of **FIA_UAU.5.2**.

### Operations

#### Assignment:

987 In **FIA_UAU.5.1**, the PP/ST author should define the available authentication mechanisms. An example of such a list could be: "none, password mechanism, biometric (retinal scan), S/key mechanism".

988 In **FIA_UAU.5.2**, the PP/ST author should specify the rules that describe how the authentication mechanisms provide authentication and when each is to be used. This means that for each situation the set of mechanisms that might be used for authenticating the user must be described. An example of a list of such rules is: "if the user has special privileges a password mechanism and a biometric mechanism

both shall be used, with success only if both succeed; for all other users a password mechanism shall be used."

989 The PP/ST author might give the boundaries within which the authorised administrator may specify specific rules. An example of a rule is: "the user shall always be authenticated by means of a token; the administrator might specify additional authentication mechanisms that also must be used." The PP/ST author also might choose not to specify any boundaries but leave the authentication mechanisms and their rules completely up to the authorised administrator.

## FIA_UAU.6 Re-authenticating

### User application notes

990 This component addresses potential needs to re-authenticate users at defined points in time. These may include user requests for the TSF to perform security relevant actions, as well as requests from non-TSF entities for re-authentication (e.g. a server application requesting that the TSF re-authenticate the client it is serving).

### Operations

#### Assignment:

991 In **FIA_UAU.6.1**, the PP/ST author should specify the list of conditions requiring re-authentication. This list could include a specified user inactivity period that has elapsed, the user requesting a change in active security attributes, or the user requesting the TSF to perform some security critical function.

992 The PP/ST author might give the boundaries within which the reauthentication should occur and leave the specifics to the authorised administrator. An example of such a rule is: "the user shall always be re-authenticated at least once a day; the administrator might specify that the re-authentication should happen more often but not more often than once every 10 minutes."

## FIA_UAU.7 Protected authentication feedback

### User application notes

993 This component addresses the feedback on the authentication process that will be provided to the user. In some systems the feedback consists of indicating how many characters have been typed but not showing the characters themselves, in other systems even this information might not be appropriate.

994 This component requires that the authentication data is not provided as-is back to the user. In a workstation environment, it could display a "dummy" (e.g. star) for each password character provided, and not the original character.

Operations

Assignment:

995 In **FIA_UAU.7.1**, the PP/ST author should specify the feedback related to the authentication process that will be provided to the user. An example of a feedback assignment is “the number of characters typed”, another type of feedback is “the authentication mechanism that failed the authentication”.

## G.5 User identification (FIA_UID)

User notes

996 This family defines the conditions under which users are required to identify themselves before performing any other actions that are to be mediated by the TSF and that require user identification.

### FIA_UID.1 Timing of identification

User application notes

997 This component poses requirements for the user to be identified. The PP/ST author can indicate specific actions that can be performed before the identification takes place.

998 If FIA_UID.1 Timing of identification is used, the TSF-mediated actions mentioned in FIA_UID.1 Timing of identification should also appear in this FIA_UAU.1 Timing of authentication.

Operations

Assignment:

999 In **FIA_UID.1.1**, the PP/ST author should specify a list of TSF-mediated actions that can be performed by the TSF on behalf of a user before the user has to identify itself. If no actions are appropriate, component FIA_UID.2 User identification before any action should be used instead. An example of such an action might include the request for help on the login procedure.

### FIA_UID.2 User identification before any action

User application notes

1000 In this component users will be identified. A user is not allowed by the TSF to perform any action before being identified.

## G.6 User-subject binding (FIA_USB)

### User notes

1001 An authenticated user, in order to use the TOE, typically activates a subject. The user's security attributes are associated (totally or partially) with this subject. This family defines requirements to create and maintain the association of the user's security attributes to a subject acting on the user's behalf.

### FIA_USB.1 User-subject binding

#### User application notes

1002 The phrase “acting on behalf of” has proven to be a contentious issue in source criteria. It is intended that a subject is acting on behalf of the user who caused the subject to come into being or to be activated to perform a certain task.

1003 Therefore, when a subject is created, that subject is acting on behalf of the user who initiated the creation. In cases where anonymity is used, the subject is still acting on behalf of a user, but the identity of that user is unknown. A special category of subjects are those subjects that serve multiple users (e.g. a server process). In such cases the user that created this subject is assumed to be the “owner”.

#### Operations

##### Assignment:

1004 In **FIA_USB.1.1**, the PP/ST author should specify a list of the user security attributes that are to be bound to subjects.

1005 In **FIA_USB.1.2**, the PP/ST author should specify any rules that are to apply upon initial association of attributes with subjects, or “none”.

1006 In **FIA_USB.1.3**, the PP/ST author should specify any rules that are to apply when changes are made to the user security attributes associated with subjects acting on behalf of users, or “none”.

# H Class FMT: Security management (normative)

1007 This class specifies the management of several aspects of the TSF: security attributes, TSF data and functions in the TSF. The different management roles and their interaction, such as separation of capability, can also be specified

1008 In an environment where the TOE is made up of multiple physically separated parts that form a distributed system, the timing issues with respect to propagation of security attributes, TSF data, and function modification become very complex, especially if the information is required to be replicated across the parts of the TOE. This should be considered when selecting components such as FMT_REV.1 Revocation, or FMT_SAE.1 Time-limited authorisation, where the behaviour might be impaired. In such situations, use of components from Internal TOE TSF data replication consistency (FPT_TRC) is advisable.

1009 Figure 28 shows the decomposition of this class into its constituent components.

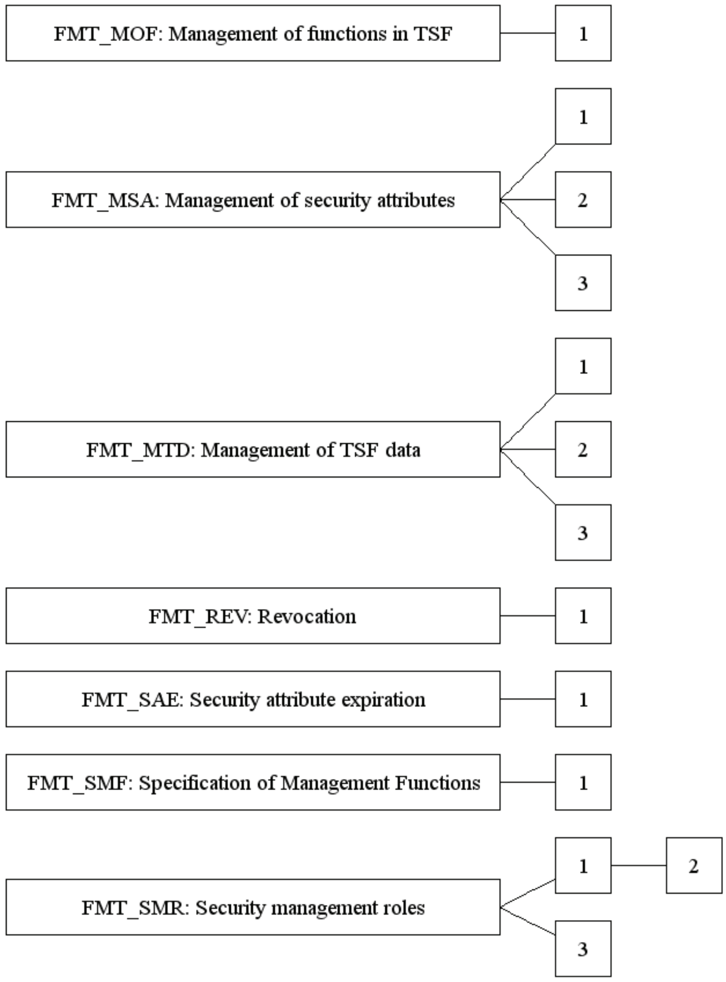



**Figure 28 - FMT: Security management class decomposition**

## H.1 Management of functions in TSF (FMT_MOF)

### User notes

1010 The TSF management functions enable authorised users to set up and control the secure operation of the TOE. These administrative functions typically fall into a number of different categories:

a) Management functions that relate to access control, accountability and authentication controls enforced by the TOE. For example, definition and update of user security characteristics (e.g. unique identifiers associated with user names, user accounts, system entry parameters) or definition and update of auditing system controls (e.g. selection of audit events, management of audit trails, audit trail analysis, and audit report generation), definition and update of per-user policy attributes (such as user clearance), definition of known system access control labels, and control and management of user groups.

b) Management functions that relate to controls over availability. For example, definition and update of availability parameters or resource quotas.

c) Management functions that relate to general installation and configuration. For example, TOE configuration, manual recovery, installation of TOE security fixes (if any), repair and reinstallation of hardware.

d) Management functions that relate to routine control and maintenance of TOE resources. For example, enabling and disabling peripheral devices, mounting of removable storage media, backup and recovery of user and system objects.

1011 Note that these functions need to be present in a TOE based on the families included in the PP or ST. It is the responsibility of the PP/ST author to ensure that adequate functions will be provided to manage the system in a secure fashion.

1012 The TSF might contain functions that can be controlled by an administrator. For example, the auditing functions could be switched off, the time synchronisation could be switchable, and/or the authentication mechanism could be modifiable.

## FMT_MOF.1 Management of security functions behaviour

### User application notes

1013 This component allows identified roles to manage the security functions of the TSF. This might entail obtaining the current status of a security function, disabling or enabling the security function, or modifying the behaviour of the security function. An example of modifying the behaviour of the security functions is changing of authentication mechanisms.

### Operations

#### Selection:

1014 In **FMT_MOF.1.1**, the PP/ST author should select whether the role can determine the behaviour of, disable, enable, and/or modify the behaviour of the security functions.

#### Assignment:

1015 In **FMT_MOF.1.1**, the PP/ST author should specify the functions that can be modified by the identified roles. Examples include auditing and time determination.

1016 In **FMT_MOF.1.1**, the PP/ST author should specify the roles that are allowed to modify the functions in the TSF. The possible roles are specified in FMT_SMR.1 Security roles.

## H.2 Management of security attributes (FMT_MSA)

User notes

1017 This family defines the requirements on the management of security attributes.

1018 Users, subjects and objects have associated security attributes that will affect the behaviour of the TSF. Examples of such security attributes are the groups to which a user belongs, the roles he/she might assume, the priority of a process (subject), and the rights belonging to a role or a user. These security attributes might need to be managed by the user, a subject or a specific authorised user (a user with explicitly given rights for this management).

1019 It is noted that the right to assign rights to users is itself a security attribute and/or potentially subject to management by FMT_MSA.1 Management of security attributes.

1020 FMT_MSA.2 Secure security attributes can be used to ensure that any accepted combination of security attributes is within a secure state. The definition of what "secure" means is left to the TOE guidance and the TSP model. If the developer provided a clear definition of the secure values and the reason why they should be considered secure, the dependency from FMT_MSA.2 Secure security attributes to ADV_SPM.1 Informal TOE security policy model can be argued away.

1021 In some instances subjects, objects or user accounts are created. If no explicit values for the related security attributes are given, default values need to be used. FMT_MSA.1 Management of security attributes can be used to specify that these default values can be managed.

### FMT_MSA.1 Management of security attributes

User application notes

1022 This component allows users acting in certain roles to manage identified security attributes. The users are assigned to a role within the component FMT_SMR.1 Security roles.

1023 The default value of a parameter is the value the parameter takes when it is instantiated without specifically assigned values. An initial value is provided during the instantiation (creation) of a parameter, and overrides the default value.

Operations

Assignment:

1024 In **FMT_MSA.1.1**, the PP/ST author should list the access control SFP or the information flow control SFP for which the security attributes are applicable.

### Selection:

1025 In **FMT_MSA.1.1**, the PP/ST author should specify the operations that can be applied to the identified security attributes. The PP/ST author can specify that the role can modify the default value (change_default), query, modify the security attribute, delete the security attributes entirely or define their own operation.

### Assignment:

1026 In **FMT_MSA.1.1**, the PP/ST author should specify the security attributes that can be operated on by the identified roles. It is possible for the PP/ST author to specify that the default value such as default access-rights can be managed. Examples of these security attributes are user-clearance, priority of service level, access control list, default access rights.

1027 In **FMT_MSA.1.1**, the PP/ST author should specify the roles that are allowed to operate on the security attributes. The possible roles are specified in FMT_SMR.1 Security roles.

1028 In **FMT_MSA.1.1**, if selected, the PP/ST author should specify which other operations the role could perform. An example of such an operation could be “create”.

## FMT_MSA.2 Secure security attributes

### User application notes

1029 This component contains requirements on the values that can be assigned to security attributes. The assigned values should be such that the TOE will remain in a secure state.

1030 The definition of what “secure” means is not answered in this component but is left to the development of the TOE (specifically ADV_SPM.1 Informal TOE security policy model) and the resulting information in the guidance. An example could be that if a user account is created, it should have a non-trivial password.

## FMT_MSA.3 Static attribute initialisation

### User application notes

1031 This component requires that the TSF provide default values for relevant object security attributes, which can be overridden by an initial value. It may still be possible for a new object to have different security attributes at creation, if a mechanism exists to specify the permissions at time of creation.

#### Operations

##### Assignment:

1032 In **FMT_MSA.3.1**, the PP/ST author should list the access control SFP or the information flow control SFP for which the security attributes are applicable.

##### Selection:

1033 In **FMT_MSA.3.1**, the PP/ST author should select whether the default property of the access control attribute will be restrictive, permissive, or another property. In case of another property, the PP/ST author should refine this to a specific property. Only one of these options may be chosen.

##### Assignment:

1034 In **FMT_MSA.3.1**, if the PP/ST author selects another property, the PP/ST author should specify the desired characteristics of the default values.

1035 In **FMT_MSA.3.2**, the PP/ST author should specify the roles that are allowed to modify the values of the security attributes. The possible roles are specified in FMT_SMR.1 Security roles.

## H.3 Management of TSF data (FMT_MTD)

#### User notes

1036 This component imposes requirements on the management of TSF data. Examples of TSF data are the current time and the audit trail. So, for example, this family allows the specification of whom can read, delete or create the audit trail.

### FMT_MTD.1 Management of TSF data

#### User application notes

1037 This component allows users with a certain role to manage values of TSF data. The users are assigned to a role within the component FMT_SMR.1 Security roles.

1038 The default value of a parameter is the values the parameter takes when it is instantiated without specifically assigned values. An initial value is provided during the instantiation (creation) of a parameter and overrides the default value.

### Operations

#### Selection:

1039 In **FMT_MTD.1.1**, the PP/ST author should specify the operations that can be applied to the identified TSF data. The PP/ST author can specify that the role can modify the default value (change_default), clear, query or modify the TSF data, or delete the TSF data entirely. If so desired the PP/ST author could specify any type of operation. To clarify "clear TSF data" means that the content of the TSF data is removed, but that the entity itself remains in the system.

#### Assignment:

1040 In **FMT_MTD.1.1**, the PP/ST author should specify the TSF data that can be operated on by the identified roles. It is possible for the PP/ST author to specify that the default value can be managed.

1041 In **FMT_MTD.1.1**, the PP/ST author should specify the roles that are allowed to operate on the TSF data. The possible roles are specified in FMT_SMR.1 Security roles.

1042 In **FMT_MTD.1.1**, if selected, the PP/ST author should specify which other operations the role could perform. An example could be "create".

## FMT_MTD.2 Management of limits on TSF data

### User application notes

1043 This component specifies limits on TSF data, and actions to be taken if these limits are exceeded. This component, for example, will allow limits on the size of the audit trail to be defined, and specification of the actions to be taken when these limits are exceeded.

### Operations

#### Assignment:

1044 In **FMT_MTD.2.1**, the PP/ST author should specify the TSF data that can have limits, and the value of those limits. An example of such TSF data is the number of users logged-in.

1045 In **FMT_MTD.2.1**, the PP/ST author should specify the roles that are allowed to modify the limits on the TSF data and the actions to be taken. The possible roles are specified in FMT_SMR.1 Security roles.

1046 In **FMT_MTD.2.2**, the PP/ST author should specify the actions to be taken if the specified limit on the specified TSF data is exceeded. An example of such TSF action is that the authorised user is informed and an audit record is generated.

### FMT_MTD.3 Secure TSF data

User application notes

1047 This component covers requirements on the values that can be assigned to TSF data. The assigned values should be such that the TOE will remain in a secure state.

1048 The definition of what "secure" means is not answered in this component but is left to the development of the TOE (specifically ADV_SPM.1 Informal TOE security policy model) and the resulting information in the guidance. If the developer provided a clear definition of the secure values and the reason why they should be considered secure, the dependency from FMT_MSA.3 Static attribute initialisation to ADV_SPM.1 Informal TOE security policy model can be argued away.

## H.4 Revocation (FMT_REV)

User notes

1049 This family addresses revocation of security attributes for a variety of entities within a TOE.

### FMT_REV.1 Revocation

User application notes

1050 This component specifies requirements on the revocation of rights. It requires the specification of the revocation rules. Examples are:

a) Revocation will take place on the next login of the user;

b) Revocation will take place on the next attempt to open the file;

c) Revocation will take place within a fixed time. This might mean that all open connections are re-evaluated every x minutes.

Operations

Selection:

1051 In **FMT_REV.1.1**, the PP/ST author should specify whether the ability to revoke security attributes from users, subjects, objects, or any additional resources shall be provided by the TSF.

Assignment:

1052 In **FMT_REV.1.1**, the PP/ST author should specify the roles that are allowed to modify the functions in the TSF. The possible roles are specified in FMT_SMR.1 Security roles.

1053 In **FMT_REV.1.1**, the PP/ST author should, if additional resources is selected, specify whether the ability to revoke their security attributes shall be provided by the TSF.

1054 In **FMT_REV.1.2**, the PP/ST author should specify the revocation rules. Examples of these rules could include: “prior to the next operation on the associated resource”, or “for all new subject creations”.

## H.5 Security attribute expiration (FMT_SAE)

User notes

1055 This family addresses the capability to enforce time limits for the validity of security attributes. This family can be applied to specify expiration requirements for access control attributes, identification and authentication attributes, certificates (key certificates such as ANSI X509 for example), audit attributes, etc.

### FMT_SAE.1 Time-limited authorisation

Operations

Assignment:

1056 In **FMT_SAE.1.1**, the PP/ST author should provide the list of security attributes for which expiration is to be supported. An example of such an attribute might be a user's security clearance.

1057 In **FMT_SAE.1.1**, the PP/ST author should specify the roles that are allowed to modify the security attributes in the TSF. The possible roles are specified in FMT_SMR.1 Security roles.

1058 In **FMT_SAE.1.2**, the PP/ST author should provide a list of actions to be taken for each security attribute when it expires. An example might be that the user's security clearance, when it expires, is set to the lowest allowable clearance on the TOE. If immediate revocation is desired by the PP/ST, the action “immediate revocation” should be specified.

## H.6 Specification of Management Functions (FMT_SMF)

User notes

1059 This family allows the specification of the management functions to be provided by the TOE. Each security management function that is listed in fulfilling the assignment is either security attribute management, TSF data management, or security function management.

### FMT_SMF.1 Specification of Management Functions

#### User application notes

1060 This component specifies the management functions to be provided.

1061 PP/ST authors should consult the "Management" sections for components included in their PP/ST to provide a basis for the management functions to be listed via this component.

#### Operations

##### Assignment:

1062 In **FMT_SMF.1.1**, the PP/ST author should specify the management functions to be provided by the TSF, either security attribute management, TSF data management, or security function management.

## H.7 Security management roles (FMT_SMR)

#### User notes

1063 This family reduces the likelihood of damage resulting from users abusing their authority by taking actions outside their assigned functional responsibilities. It also addresses the threat that inadequate mechanisms have been provided to securely administer the TSF.

1064 This family requires that information be maintained to identify whether a user is authorised to use a particular security-relevant administrative function.

1065 Some management actions can be performed by users, others only by designated people within the organisation. This family allows the definition of different roles, such as owner, auditor, administrator, daily-management.

1066 The roles as used in this family are security related roles. Each role can encompass an extensive set of capabilities (e.g. root in UNIX), or can be a single right (e.g. right to read a single object such as the helpfile). This family defines the roles. The capabilities of the role are defined in Management of functions in TSF (FMT_MOF), Management of security attributes (FMT_MSA) and Management of TSF data (FMT_MTD).

1067 Some type of roles might be mutually exclusive. For example the daily-management might be able to define and activate users, but might not be able to remove users (which is reserved for the administrator (role)). This class will allow policies such as two-person control to be specified.

## FMT_SMR.1 Security roles

### User application notes

1068 This component specifies the different roles that the TSF should recognise. Often the system distinguishes between the owner of an entity, an administrator and other users.

### Operations

#### Assignment:

1069 In **FMT_SMR.1.1**, the PP/ST author should specify the roles that are recognised by the system. These are the roles that users could occupy with respect to security. Examples are: owner, auditor and administrator.

## FMT_SMR.2 Restrictions on security roles

### User application notes

1070 This component specifies the different roles that the TSF should recognise, and conditions on how those roles could be managed. Often the system distinguishes between the owner of an entity, an administrator and other users.

1071 The conditions on those roles specify the interrelationship between the different roles, as well as restrictions on when the role can be assumed by a user.

### Operations

#### Assignment:

1072 In **FMT_SMR.2.1**, the PP/ST author should specify the roles that are recognised by the system. These are the roles that users could occupy with respect to security. Examples are: owner, auditor, administrator.

1073 In **FMT_SMR.2.3**, the PP/ST author should specify the conditions that govern role assignment. Examples of these conditions are: “an account cannot have both the auditor and administrator role” or “a user with the assistant role must also have the owner role”.

## FMT_SMR.3 Assuming roles

### User application notes

1074 This component specifies that an explicit request must be given to assume the specific role.

### Operations

#### Assignment:

1075 In **FMT_SMR.3.1**, the PP/ST author should specify the roles that require an explicit request to be assumed. Examples are: auditor and administrator.

# I Class FPR: Privacy (normative)

1076 This class describes the requirements that could be levied to satisfy the users' privacy needs, while still allowing the system flexibility as far as possible to maintain sufficient control over the operation of the system.

1077 In the components of this class there is flexibility as to whether or not authorised users are covered by the required security functions. For example, a PP/ST author might consider it appropriate not to require protection of the privacy of users against a suitably authorised user.

1078 This class, together with other classes (such as those concerned with audit, access control, trusted path, and non-repudiation) provides the flexibility to specify the desired privacy behaviour. On the other hand, the requirements in this class might impose limitations on the use of the components of other classes, such as FIA: Identification and authentication or FAU: Security audit. For example, if authorised users are not allowed to see the user identity (e.g. Anonymity or Pseudonymity), it will obviously not be possible to hold individual users accountable for any security relevant actions they perform that are covered by the privacy requirements. However, it may still be possible to include audit requirements in a PP/ST, where the fact that a particular security relevant event has occurred is more important than knowing who was responsible for it.

1079 Additional information is provided in the application notes for class FAU: Security audit, where it is explained that the definition of "identity" in the context of auditing can also be an alias or other information that could identify a user.

1080 This class describes four families: Anonymity, Pseudonymity, Unlinkability and Unobservability. Anonymity, Pseudonymity and Unlinkability have a complex interrelationship. When choosing a family, the choice should depend on the threats identified. For some types of privacy threats, pseudonymity will be more appropriate than anonymity (e.g. if there is a requirement for auditing). In addition, some types of privacy threats are best countered by a combination of components from several families.

1081 All families assume that a user does not explicitly perform an action that discloses the user's own identity. For example, the TSF is not expected to screen the user name in electronic messages or databases.

1082 All families in this class have components that can be scoped through operations. These operations allow the PP/ST author to state the cooperating users/subjects to which the TSF must be resistant. An example of an instantiation of anonymity could be: " The TSF shall ensure that the users and/or subjects are unable to determine the user identity bound to the teleconsulting application".

1083 It is noted that the TSF should not only provide this protection against individual users, but also against users cooperating to obtain the information. The strength of the protection provided by this class should be described as strength of function as specified in annexes A and B of CC Part 1.

1084 Figure 29 shows the decomposition of this class into its constituent components.

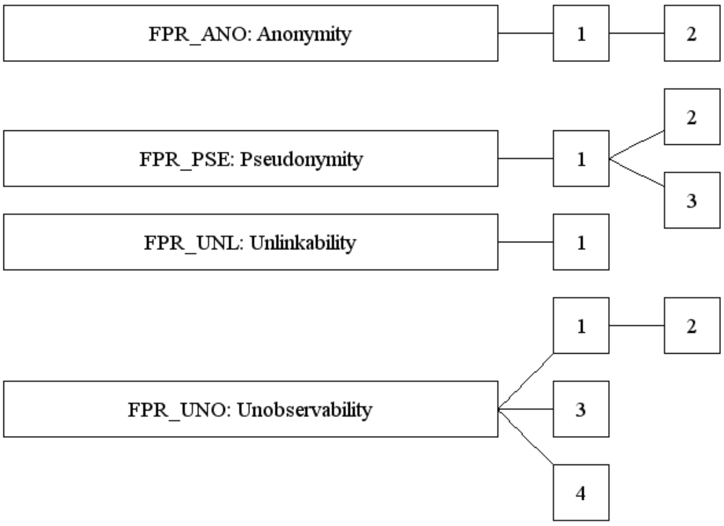



**Figure 29 - FPR: Privacy class decomposition**

## I.1 Anonymity (FPR_ANO)

### User notes

1085 Anonymity ensures that a subject may use a resource or service without disclosing its user identity.

1086 The intention of this family is to specify that a user or subject might take action without releasing its user identity to others such as users, subjects, or objects. The family provides the PP/ST author with a means to identify the set of users that cannot see the identity of someone performing certain actions.

1087 Therefore if a subject, using anonymity, performs an action, another subject will not be able to determine either the identity or even a reference to the identity of the user employing the subject. The focus of the anonymity is on the protection of the users identity, not on the protection of the subject identity; hence, the identity of the subject is not protected from disclosure.

1088 Although the identity of the subject is not released to other subjects or users, the TSF is not explicitly prohibited from obtaining the users identity. In case the TSF is not allowed to know the identity of the user, FPR_ANO.2 Anonymity without soliciting information could be invoked. In that case the TSF should not request the user information.

1089 The interpretation of "determine" should be taken in the broadest sense of the word. The PP/ST author might want to use a Strength of Function to indicate how much rigour should be applied.

1090 The component levelling distinguishes between the users and an authorised user. An authorised user is often excluded from the component, and therefore allowed to retrieve a user's identity. However, there is no specific requirement that an authorised user must be able to have the capability to determine the user's identity. For ultimate privacy the components would be used to say that no user or authorised user can see the identity of anyone performing any action.

1091 Although some systems will provide anonymity for all services that are provided, other systems provide anonymity for certain subjects/operations. To provide this flexibility, an operation is included where the scope of the requirement is defined. If the PP/ST author wants to address all subjects/operations, the words "all subjects and all operations" could be provided.

1092 Possible applications include the ability to make enquiries of a confidential nature to public databases, respond to electronic polls, or make anonymous payments or donations.

1093 Examples of potential hostile users or subjects are providers, system operators, communication partners and users, who smuggle malicious parts (e.g. Trojan Horses) into systems. All of these users can investigate usage patterns (e.g. which users used which services) and misuse this information.

## FPR_ANO.1 Anonymity

### User application notes

1094 This component ensures that the identity of a user is protected from disclosure. There may be instances, however, that a given authorised user can determine who performed certain actions. This component gives the flexibility to capture either a limited or total privacy policy.

### Operations

#### Assignment:

1095 In **FPR_ANO.1.1**, the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

1096 In **FPR_ANO.1.1**, the PP/ST author should identify the list of subjects and/or operations and/or objects where the real user name of the subject should be protected, for example, “the voting application”.

### FPR_ANO.2 Anonymity without soliciting information

#### User application notes

1097 This component is used to ensure that the TSF is not allowed to know the identity of the user.

#### Operations

##### Assignment:

1098 In **FPR_ANO.2.1**, the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

1099 In **FPR_ANO.2.1**, the PP/ST author should identify the list of subjects and/or operations and/or objects where the real user name of the subject should be protected, for example, “the voting application”.

1100 In **FPR_ANO.2.2**, the PP/ST author should identify the list of services which are subject to the anonymity requirement, for example, “the accessing of job descriptions”.

1101 In **FPR_ANO.2.2**, the PP/ST author should identify the list of subjects from which the real user name of the subject should be protected when the specified services are provided.

## I.2 Pseudonymity (FPR_PSE)

#### User notes

1102 Pseudonymity ensures that a user may use a resource or service without disclosing its identity, but can still be accountable for that use. The user can be accountable by directly being related to a reference (alias) held by the TSF, or by providing an alias that will be used for processing purposes, such as an account number.

1103 In several respects, pseudonymity resembles anonymity. Both pseudonymity and anonymity protect the identity of the user, but in pseudonymity a reference to the user's identity is maintained for accountability or other purposes.

1104 The component FPR_PSE.1 Pseudonymity does not specify the requirements on the reference to the user's identity. For the purpose of specifying requirements on this reference two sets of requirements are presented: FPR_PSE.2 Reversible pseudonymity and FPR_PSE.3 Alias pseudonymity.

1105 A way to use the reference is by being able to obtain the original user identifier. For example, in a digital cash environment it would be advantageous to be able to trace the user's identity when a check has been issued multiple times (i.e. fraud). In general, the user's identity needs to be retrieved under specific conditions. The PP/ST author might want to incorporate FPR_PSE.2 Reversible pseudonymity to describe those services.

1106 Another usage of the reference is as an alias for a user. For example, a user who does not wish to be identified, can provide an account to which the resource utilisation should be charged. In such cases, the reference to the user identity is an alias for the user, where other users or subjects can use the alias for performing their functions without ever obtaining the user's identity (for example, statistical operations on use of the system). In this case, the PP/ST author might wish to incorporate FPR_PSE.3 Alias pseudonymity to specify the rules to which the reference must conform.

1107 Using these constructs above, digital money can be created using FPR_PSE.2 Reversible pseudonymity specifying that the user identity will be protected and, if so specified in the condition, that there be a requirement to trace the user identity if the digital money is spent twice. When the user is honest, the user identity is protected; if the user tries to cheat, the user identity can be traced.

1108 A different kind of system could be a digital credit card, where the user will provide a pseudonym that indicates an account from which the cash can be subtracted. In such cases, for example, FPR_PSE.3 Alias pseudonymity could be used. This component would specify that the user identity will be protected and, furthermore, that the same user will only get assigned values for which he/she has provided money (if so specified in the conditions).

1109 It should be realised that the more stringent components potentially cannot be combined with other requirements, such as identification and authentication or audit. The interpretation of “determine the identity” should be taken in the broadest sense of the word. The information is not provided by the TSF during the operation, nor can the entity determine the subject or the owner of the subject that invoked the operation, nor will the TSF record information, available to the users or subjects, which might release the user identity in the future.

1110 The intent is that the TSF not reveal any information that would compromise the identity of the user, e.g. the identity of subjects acting on the user's behalf. The information that is considered to be sensitive depends on the effort an attacker is capable of spending. Therefore, the Pseudonymity (FPR_PSE) family is subject to Strength of Function requirements.

1111 Possible applications include the ability to charge a caller for premium rate telephone services without disclosing his or her identity, or to be charged for the anonymous use of an electronic payment system.

1112 Examples of potential hostile users are providers, system operators, communication partners and users, who smuggle malicious parts (e.g. Trojan Horses) into systems. All of these attackers can investigate which users used which services and misuse this information. Additionally to Anonymity services, Pseudonymity Services contains methods for authorisation without identification, especially for anonymous payment ("Digital Cash"). This helps providers to obtain their payment in a secure way while maintaining customer anonymity.

## FPR_PSE.1 Pseudonymity

### User application notes

1113 This component provides the user protection against disclosure of identity to other users. The user will remain accountable for its actions.

### Operations

#### Assignment:

1114 In **FPR_PSE.1.1**, the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

1115 In **FPR_PSE.1.1**, the PP/ST author should identify the list of subjects and/or operations and/or objects where the real user name of the subject should be protected, for example, "the accessing of job offers". Note that "objects" includes any other attributes that might enable another user or subject to derive the actual identity of the user.

1116 In **FPR_PSE.1.2**, the PP/ST author should identify the (one or more) number of aliases the TSF is able to provide.

1117 In **FPR_PSE.1.2**, the PP/ST author should identify the list of subjects to whom the TSF is able to provide an alias.

#### Selection:

1118 In **FPR_PSE.1.3**, the PP/ST author should specify whether the user alias is generated by the TSF, or supplied by the user. Only one of these options may be chosen.

#### Assignment:

1119 In **FPR_PSE.1.3**, the PP/ST author should identify the metric to which the TSF-generated or user-generated alias should conform.

## FPR_PSE.2 Reversible pseudonymity

### User application notes

1120 In this component, the TSF shall ensure that under specified conditions the user identity related to a provided reference can be determined.

1121 In FPR_PSE.1 Pseudonymity the TSF shall provide an alias instead of the user identity. When the specified conditions are satisfied, the user identity to which the alias belong can be determined. An example of such a condition in an electronic cash environment is: “ The TSF shall provide the notary a capability to determine the user identity based on the provided alias only under the conditions that a check has been issued twice.”.

### Operations

#### Assignment:

1122 In **FPR_PSE.2.1**, the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

1123 In **FPR_PSE.2.1**, the PP/ST author should identify the list of subjects and/or operations and/or objects where the real user name of the subject should be protected, for example, “the accessing of job offers”. Note that “objects” includes any other attributes that might enable another user or subject to derive the actual identity of the user.

1124 In **FPR_PSE.2.2**, the PP/ST author should identify the (one or more) number of aliases the TSF, is able to provide.

1125 In **FPR_PSE.2.2**, the PP/ST author should identify the list of subjects to whom the TSF is able to provide an alias.

#### Selection:

1126 In **FPR_PSE.2.3**, the PP/ST author should specify whether the user alias is generated by the TSF or supplied by the user. Only one of these options may be chosen.

#### Assignment:

1127 In **FPR_PSE.2.3**, the PP/ST author should identify the metric to which the TSF-generated or user-generated alias should conform.

#### Selection:

1128 In **FPR_PSE.2.4**, the PP/ST author should select whether the authorised user and/or trusted subjects can determine the real user name.

#### Assignment:

1129 In **FPR_PSE.2.4**, the PP/ST author should identify the list of conditions under which the trusted subjects and authorised user can determine the real user name based on the provided reference. These conditions can be conditions such as time of day, or they can be administrative such as on a court order.

1130 In **FPR_PSE.2.4**, the PP/ST author should identify the list of trusted subjects that can obtain the real user name under a specified condition, for example, a notary or special authorised user.

## FPR_PSE.3 Alias pseudonymity

### User application notes

1131 In this component, the TSF shall ensure that the provided reference meets certain construction rules, and thereby can be used in a secure way by potentially insecure subjects.

1132 If a user wants to use disk resources without disclosing its identity, pseudonymity can be used. However, every time the user accesses the system, the same alias must be used. Such conditions can be specified in this component.

### Operations

#### Assignment:

1133 In **FPR_PSE.3.1**, the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

1134 In **FPR_PSE.3.1**, the PP/ST author should identify the list of subjects and/or operations and/or objects where the real user name of the subject should be protected, for example, "the accessing of job offers". Note that "objects" includes any other attributes which might enable another user or subject to derive the actual identity of the user.

1135 In **FPR_PSE.3.2**, the PP/ST author should identify the (one or more) number of aliases the TSF is able to provide.

1136 In **FPR_PSE.3.2**, the PP/ST author should identify the list of subjects to whom the TSF is able to provide an alias.

Selection:

1137 In **FPR_PSE.3.3**, the PP/ST author should specify whether the user alias is generated by the TSF, or supplied by the user. Only one of these options may be chosen.

Assignment:

1138 In **FPR_PSE.3.3**, the PP/ST author should identify the metric to which the TSF-generated or user-generated alias should conform.

1139 In **FPR_PSE.3.4**, the PP/ST author should identify the list of conditions that indicate when the used reference for the real user name shall be identical and when it shall be different, for example, "when the user logs on to the same host" it will use a unique alias.

## I.3 Unlinkability (FPR_UNL)

### User notes

1140 Unlinkability ensures that a user may make multiple uses of resources or services without others being able to link these uses together. Unlinkability differs from pseudonymity that, although in pseudonymity the user is also not known, relations between different actions can be provided.

1141 The requirements for unlinkability are intended to protect the user identity against the use of profiling of the operations. For example, when a telephone smart card is employed with a unique number, the telephone company can determine the behaviour of the user of this telephone card. When a telephone profile of the users is known, the card can be linked to a specific user. Hiding the relationship between different invocations of a service or access of a resource will prevent this kind of information gathering.

1142 As a result, a requirement for unlinkability could imply that the subject and user identity of an operation must be protected. Otherwise this information might be used to link operations together.

1143 Unlinkability requires that different operations cannot be related. This relationship can take several forms. For example, the user associated with the operation, or the terminal which initiated the action, or the time the action was executed. The PP/ST author can specify what kind of relationships are present that must be countered.

1144 Possible applications include the ability to make multiple use of a pseudonym without creating a usage pattern that might disclose the user's identity.

1145 Examples for potential hostile subjects and users are providers, system operators, communication partners and users, who smuggle malicious parts, (e.g. Trojan Horses) into systems, they do not operate but want to get information about. All of these attackers can investigate (e.g. which users used which services) and misuse this information. Unlinkability protects users from linkages, which could be drawn between several actions of a customer. An example is a series of phone calls made by an anonymous customer to different partners, where the combination of the partner's identities might disclose the identity of the customer.

## FPR_UNL.1 Unlinkability

### User application notes

1146 This component ensures that users cannot link different operations in the system and thereby obtain information.

### Operations

#### Assignment:

1147 In **FPR_UNL.1.1**, the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

1148 In **FPR_UNL.1.1**, the PP/ST author should identify the list of operations which should be subjected to the unlinkability requirement, for example, “sending email”.

#### Selection:

1149 In **FPR_UNL.1.1**, the PP/ST author should select the relationships that should be obscured. The selection allows either the user identity or an assignment of relations to be specified.

#### Assignment:

1150 In **FPR_UNL.1.1**, the PP/ST author should identify the list of relations which should be protected against, for example, “originate from the same terminal”.

# I.4 Unobservability (FPR_UNO)

## User notes

1151 Unobservability ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used.

1152 Unobservability approaches the user identity from a different direction than the previous families Anonymity, Pseudonymity and Unlinkability. In this case, the intent is to hide the use of a resource or service, rather than to hide the user's identity.

1153 A number of techniques can be applied to implement unobservability. Examples of techniques to provide unobservability are:

a) Allocation of information impacting unobservability: Unobservability relevant information (e.g. information that describes that an operation occurred) can be allocated in several locations within the TOE. The information might be allocated to a single randomly chosen part of the TOE such that an attacker does not know which part of the TOE should be attacked. An alternative system might distribute the information such that no single part of the TOE has sufficient information that, if circumvented, the privacy of the user would be compromised. This technique is explicitly addressed in FPR_UNO.2 Allocation of information impacting unobservability.

b) Broadcast: When information is broadcast (e.g. ethernet, radio), users cannot determine who actually received and used that information. This technique is especially useful when information should reach receivers which have to fear a stigma for being interested in that information (e.g. sensitive medical information).

c) Cryptographic protection and message padding: People observing a message stream might obtain information from the fact that a message is transferred and from attributes on that message. By traffic padding, message padding and encrypting the message stream, the transmission of a message and its attributes can be protected.

1154 Sometimes, users should not see the use of a resource, but an authorised user must be allowed to see the use of the resource in order to perform his duties. In such cases, the FPR_UNO.4 Authorised user observability could be used, which provides the capability for one or more authorised users to see the usage.

1155 This family makes use of the concept “parts of the TOE”. This is considered any part of the TOE that is either physically or logically separated from other parts of the TOE. In the case of logical separation Domain separation (FPT_SEP) may be relevant.

1156 Unobservability of communications may be an important factor in many areas, such as the enforcement of constitutional rights, organisational policies, or in defence related applications.

## FPR_UNO.1 Unobservability

### User application notes

1157 This component requires that the use of a function or resource cannot be observed by unauthorised users. In addition to this component, a PP/ST author might want to incorporate Covert Channel Analysis.

### Operations

#### Assignment:

1158 In **FPR_UNO.1.1**, the PP/ST author should specify the list of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

1159 In **FPR_UNO.1.1**, the PP/ST author should identify the list of operations that are subjected to the unobservability requirement. Other users/subjects will then not be able to observe the operations on a covered object in the specified list (e.g. reading and writing to the object).

1160 In **FPR_UNO.1.1**, the PP/ST author should identify the list of objects which are covered by the unobservability requirement. An example could be a specific mail server or ftp site.

1161 In **FPR_UNO.1.1**, the PP/ST author should specify the set of protected users and/or subjects whose unobservability information will be protected. An example could be: “users accessing the system through the internet”.

## FPR_UNO.2 Allocation of information impacting unobservability

### User application notes

1162 This component requires that the use of a function or resource cannot be observed by specified users or subjects. Furthermore this component specifies that information related to the privacy of the user is distributed within the TOE such that attackers might not know which part of the TOE to target, or they need to attack multiple parts of the TOE.

1163 An example of the use of this component is the use of a randomly allocated node to provide a function. In such a case the component might require that the privacy related information shall only be available to one identified part of the TOE, and will not be communicated outside this part of the TOE.

1164 A more complex example can be found in some “voting algorithms”. Several parts of the TOE will be involved in the service, but no individual part of the TOE will be able to violate the policy. So a person may cast a vote (or not) without the TOE being able to determine whether a vote has been cast and what the vote happened to be (unless the vote was unanimous).

1165 In addition to this component, a PP/ST author might want to incorporate Covert Channel Analysis.

### Operations

#### Assignment:

1166 In **FPR_UNO.2.1**, the PP/ST author should specify the list of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

1167 In **FPR_UNO.2.1**, the PP/ST author should identify the list of operations that are subjected to the unobservability requirement. Other users/subjects will then not be able to observe the operations on a covered object in the specified list (e.g. reading and writing to the object).

1168 In **FPR_UNO.2.1**, the PP/ST author should identify the list of objects which are covered by the unobservability requirement. An example could be a specific mail server or ftp site.

1169 In **FPR_UNO.2.1**, the PP/ST author should specify the set of protected users and/or subjects whose unobservability information will be protected. An example could be: “users accessing the system through the internet”.

1170 In **FPR_UNO.2.2**, the PP/ST author should identify which privacy related information should be distributed in a controlled manner. Examples of this information could be: IP address of subject, IP address of object, time, used encryption keys.

1171 In **FPR_UNO.2.2**, the PP/ST author should specify the conditions to which the dissemination of the information should adhere. These conditions should be maintained throughout the lifetime of the privacy related information of each instance. Examples of these conditions could be: “the information shall only be present at a single separated part of the TOE and shall not be communicated outside this part of the TOE.", "the information shall only reside in a single separated part of the TOE, but shall be moved to another part of the TOE periodically", "the information shall be distributed between the different parts of the TOE such that compromise of any 5 separated parts of the TOE will not compromise the security policy".

## FPR_UNO.3 Unobservability without soliciting information

### User application notes

1172 This component is used to require that the TSF does not try to obtain information that might compromise unobservability when provided specific services. Therefore the TSF will not solicit (i.e. try to obtain from other entities) any information that might be used to compromise unobservability.

### Operations

#### Assignment:

1173 In **FPR_UNO.3.1**, the PP/ST author should identify the list of services which are subject to the unobservability requirement, for example, "the accessing of job descriptions".

1174 In **FPR_UNO.3.1**, the PP/ST author should identify the list of subjects from which privacy related information should be protected when the specified services are provided.

1175 In **FPR_UNO.3.1**, the PP/ST author should specify the privacy related information that will be protected from the specified subjects. Examples include the identity of the subject that used a service and the quantity of a service that has been used such as memory resource utilisation.

## FPR_UNO.4 Authorised user observability

### User application notes

1176 This component is used to require that there will be one or more authorised users with the rights to view the resource utilisation. Without this component, this review is allowed, but not mandated.

### Operations

#### Assignment:

1177 In **FPR_UNO.4.1**, the PP/ST author should specify the set of authorised users for which the TSF must provide the capability to observe the resource utilisation. A set of authorised users, for example, could be a group of authorised users which can operate under the same role or can all use the same process(es).

1178 In **FPR_UNO.4.1**, the PP/ST author should specify the set of resources and/or services that the authorised user must be able to observe.

# J Class FPT: Protection of the TSF (normative)

1179 This class contains families of functional requirements that relate to the integrity and management of the mechanisms that provide the TSF (independent of TSP-specifics) and to the integrity of TSF data (independent of the specific contents of the TSP data). In some sense, families in this class may appear to duplicate components in the FDP: User data protection class; they may even be implemented using the same mechanisms. However, FDP: User data protection focuses on user data protection, while FPT: Protection of the TSF focuses on TSF data protection. In fact, components from the FPT: Protection of the TSF class are necessary to provide requirements that the SFPs in the TOE cannot be tampered with or bypassed.

1180 From the point of view of this class, there are three significant portions that make up the TSF:

a) The TSF's abstract machine, which is the virtual or physical machine upon which the specific TSF implementation under evaluation executes.

b) The TSF's implementation, which executes on the abstract machine and implements the mechanisms that enforce the TSP.

c) The TSF's data, which are the administrative databases that guide the enforcement of the TSP.

1181 All of the families in the FPT: Protection of the TSF class can be related to these areas, and fall into the following groupings:

a) TSF physical protection (FPT_PHP), which provides an authorised user with the ability to detect external attacks on the parts of the TOE that comprise the TSF.

b) Underlying abstract machine test (FPT_AMT) and TSF self test (FPT_TST), which provide an authorised user with the ability to verify the correct operation of the underlying abstract machine and the TSF as well as the integrity of the TSF data and executable code.

c) Domain separation (FPT_SEP) and Reference mediation (FPT_RVM), which protect the TSF during execution and ensure that the TSF cannot be bypassed. When appropriate components from these families are combined with the appropriate components from TSF internals (ADV_INT), the TOE can be said to have what has been traditionally called a “Reference Monitor.”

d) Trusted recovery (FPT_RCV), Fail secure (FPT_FLS), and Internal TOE TSF data replication consistency (FPT_TRC), which address the behaviour of the TSF when failure occurs and immediately after.

e) Availability of exported TSF data (FPT_ITA), Confidentiality of exported TSF data (FPT_ITC), Integrity of exported TSF data (FPT_ITI), which address the protection and availability of TSF data between the TSF and a remote trusted IT product.

f) Internal TOE TSF data transfer (FPT_ITT), which addresses protection of TSF data when it is transmitted between physically-separated parts of the TOE.

g) Replay detection (FPT_RPL), which addresses the replay of various types of information and/or operations.

h) State synchrony protocol (FPT_SSP), which addresses the synchronisation of states, based upon TSF data, between different parts of a distributed TSF.

i) Time stamps (FPT_STM), which addresses reliable timing.

j) Inter-TSF TSF data consistency (FPT_TDC), which addresses the consistency of TSF data shared between the TSF and a remote trusted IT product.

1182 Figure 30 shows the decomposition of this class into its constituent components.

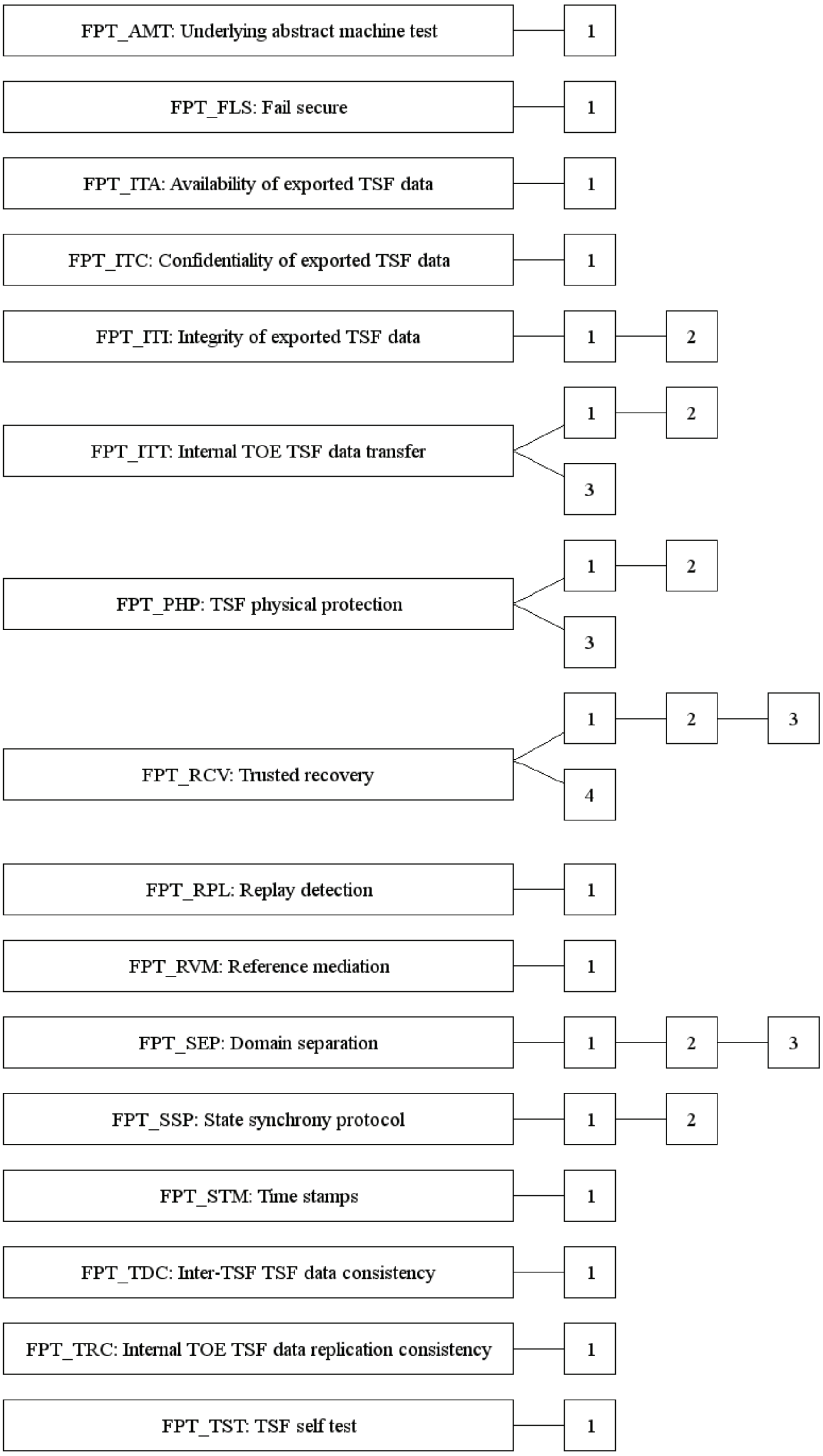



**Figure 30 - FPT: Protection of the TSF class decomposition**

# J.1 Underlying abstract machine test (FPT_AMT)

### User notes

1183 This family defines the requirements for the TSF's testing of security assumptions made about the underlying abstract machine upon which the TSF relies. This “abstract” machine could be a hardware/firmware platform, or it could be some known and assessed hardware/software combination acting as a virtual machine. Examples of such testing could be testing hardware page protection, sending sample packets across a network to ensure receipt, and verifying the behaviour of the virtual machine interface. These tests can be carried out either in some maintenance state, at start-up, on-line, or continuously. The actions to be taken by the TOE as the result of testing are defined in Trusted recovery (FPT_RCV).

1184 The term “underlying abstract machine” typically refers to the hardware components upon which the TSF has been implemented. However, the phrase can also be used to refer to an underlying, previously evaluated hardware and software combination behaving as a virtual machine upon which the TSF relies.

1185 The tests of the abstract machine may take various forms:

a) Power-On Tests. These are tests that ensure the correct operation of the underlying platform. For hardware and firmware, this might include tests of elements such as memory boards, data paths, buses, control logic, processor registers, communication ports, console interfaces, speakers, and peripherals. For software elements (virtual machine), this would include verification of correct initialisation and behaviour.

b) Loadable Tests. These are tests that might be loaded and executed by an authorised user or be activated by specific conditions. This might include processor component stress tests (logic units, calculation units, etc.) and control memory.

### Evaluator notes

1186 The tests of the underlying abstract machine should be sufficient to test all of the characteristics of the underlying abstract machine upon which the TSF relies.

## FPT_AMT.1 Abstract machine testing

### User application notes

1187 This component provides support for the periodic testing of the security assumptions of the underlying abstract machine upon which the TSF's operation depends, by requiring the ability to periodically invoke testing functions.

1188 The PP/ST author may refine the requirement to state whether the function should be available in off-line, on-line or maintenance mode.

### Evaluator notes

1189 It is acceptable for the functions for periodic testing to be available only in an off-line or maintenance mode. Controls should be in place to limit access, during maintenance, to authorised users.

### Operations

#### Selection:

1190 In **FPT_AMT.1.1**, the PP/ST author should specify when the TSF will execute the abstract machine testing, during initial start-up, periodically during normal operation, at the request of an authorised user, or under other conditions. In the case of the latter option, the PP/ST author should refine what those conditions are. The PP/ST author, through this selection, has the ability to indicate the frequency with which the self tests will be run. If the tests are run often, then the end users should have more confidence that the TOE is operating correctly than if the tests are run less frequently. However, this need for confidence that the TOE is operating correctly must be balanced with the potential impact on the availability of the TOE, as often times, self tests may delay the normal operation of a TOE.

#### Assignment:

1191 In **FPT_AMT.1.1**, the PP/ST author should, if other conditions is selected, specify the frequency with which the self tests will be run.

## J.2 Fail secure (FPT_FLS)

### User notes

1192 The requirements of this family ensure that the TOE will not violate its TSP in the event of certain types of failures in the TSF.

### FPT_FLS.1 Failure with preservation of secure state

### User application notes

1193 The term “secure state” refers to a state in which the TSF data are consistent and the TSF continues correct enforcement of the TSP. The “secure state” is defined in the TSP model. If the developer provided a clear definition of the secure state and the reason why it should be considered secure, the dependency from FPT_FLS.1 Failure with preservation of secure state to ADV_SPM.1 Informal TOE security policy model can be argued away.

1194 Although it is desirable to audit situations in which failure with preservation of secure state occurs, it is not possible in all situations. The PP/ST author should specify those situations in which audit is desired and feasible.

1195 Failures in the TSF may include “hard” failures, which indicate an equipment malfunction and which may require maintenance, service or repair of the TSF. Failures in the TSF may also include recoverable “soft” failures, which may only require initialisation or resetting of the TSF.

### Operations

#### Assignment:

1196 In **FPT_FLS.1.1**, the PP/ST author should list the types of failures in the TSF for which the TSF should “fail secure,” that is, should preserve a secure state and continue to correctly enforce the TSP.

## J.3 Availability of exported TSF data (FPT_ITA)

### User notes

1197 This family defines the rules for the prevention of loss of availability of TSF data moving between the TSF and a remote trusted IT product. This data could be TSF critical data such as passwords, keys, audit data, or TSF executable code.

1198 This family is used in a distributed system context where the TSF is providing TSF data to a remote trusted IT product. The TSF can only take the measures at its site and cannot be held responsible for the TSF at the other trusted IT product.

1199 If there are different availability metrics for different types of TSF data, then this component should be iterated for each unique pairing of metrics and types of TSF data.

### FPT_ITA.1 Inter-TSF availability within a defined availability metric

### Operations

#### Assignment:

1200 In **FPT_ITA.1.1**, the PP/ST author should specify the types of TSF data that are subject to the availability metric.

1201 In **FPT_ITA.1.1**, the PP/ST should specify the availability metric for the applicable TSF data.

1202 In **FPT_ITA.1.1**, the PP/ST author should specify the conditions under which availability must be ensured. For example: there must be a connection between the TOE and the remote trusted IT product.

## J.4 Confidentiality of exported TSF data (FPT_ITC)

User notes

1203 This family defines the rules for the protection from unauthorised disclosure of TSF data moving between the TSF and a remote trusted IT product. Examples of this data are TSF critical data such as passwords, keys, audit data, or TSF executable code.

1204 This family is used in a distributed system context where the TSF is providing TSF data to a remote trusted IT product. The TSF can only take the measures at its site and cannot be held responsible for the behaviour of the other trusted IT product.

### FPT_ITC.1 Inter-TSF confidentiality during transmission

Evaluator notes

1205 Confidentiality of TSF Data during transmission is necessary to protect such information from disclosure. Some possible implementations that could provide confidentiality include the use of cryptographic algorithms as well as spread spectrum techniques.

## J.5 Integrity of exported TSF data (FPT_ITI)

User notes

1206 This family defines the rules for the protection, from unauthorised modification, of TSF data during transmission between the TSF and a remote trusted IT product. Examples of this data are TSF critical data such as passwords, keys, audit data, or TSF executable code.

1207 This family is used in a distributed system context where the TSF is exchanging TSF data with a remote trusted IT product. Note that a requirement that addresses modification, detection, or recovery at the remote trusted IT product cannot be specified, as the mechanisms that a remote trusted IT product will use to protect its data cannot be determined in advance. For this reason, these requirements are expressed in terms of the “TSF providing a capability” which the remote trusted IT product can use.

### FPT_ITI.1 Inter-TSF detection of modification

User application notes

1208 This component should be used in situations where it is sufficient to detect when data have been modified. An example of such a situation is one in which the remote trusted IT product can request the TOE's TSF to retransmit data when modification has been detected, or respond to such types of request.

1209 The desired strength of modification detection is based upon a specified modification metric that is a function of the algorithm used, which may range from a weak checksum and parity mechanisms that may fail to detect multiple bit changes, to more complicated cryptographic checksum approaches.

Operations

Assignment:

1210 In **FPT_ITI.1.1**, the PP/ST should specify the modification metric that the detection mechanism must satisfy. This modification metric shall specify the desired strength of the modification detection.

1211 In **FPT_ITI.1.2**, the PP/ST should specify the actions to be taken if a modification of TSF data has been detected. An example of an action is: “ignore the TSF data, and request the originating trusted product to send the TSF data again”.

## FPT_ITI.2 Inter-TSF detection and correction of modification

User application notes

1212 This component should be used in situations where it is necessary to detect or correct modifications of TSF critical data.

1213 The desired strength of modification detection is based upon a specified modification metric that is a function of the algorithm used, which may range from a checksum and parity mechanisms that may fail to detect multiple bit changes, to more complicated cryptographic checksum approaches. The metric that needs to be defined can either refer to the attacks it will resist (e.g. only 1 in a 1000 random messages will be accepted), or to mechanisms that are well known in the public literature (e.g. the strength must be conformant to the strength offered by Secure Hash Algorithm).

1214 The approach taken to correct modification might be done through some form of error correcting checksum.

Evaluator notes

1215 Some possible means of satisfying this requirement involves the use of cryptographic functions or some form of checksum.

Operations

Assignment:

1216 In **FPT_ITI.2.1**, the PP/ST should specify the modification metric that the detection mechanism must satisfy. This modification metric shall specify the desired strength of the modification detection.

1217 In **FPT_ITI.2.2**, the PP/ST should specify the actions to be taken if a modification of TSF data has been detected. An example of an action is: "ignore the TSF data, and request the originating trusted product to send the TSF data again".

1218 In **FPT_ITI.2.3**, the PP/ST author should define the types of modification from which the TSF should be capable of recovering.

## J.6 Internal TOE TSF data transfer (FPT_ITT)

### User notes

1219 This family provides requirements that address protection of TSF data when it is transferred between separate parts of a TOE across an internal channel.

1220 The determination of the degree of separation (i.e., physical or logical) that would make application of this family useful depends on the intended environment of use. In a hostile environment, there may be risks arising from transfers between parts of the TOE separated by only a system bus or an inter-process communications channel. In more benign environments, the transfers may be across more traditional network media.

### Evaluator notes

1221 One practical mechanism available to a TSF to provide this protection is cryptographically-based.

### FPT_ITT.1 Basic internal TSF data transfer protection

#### Operations

##### Selection:

1222 In **FPT_ITT.1.1**, the PP/ST author should specify the desired type of protection to be provided from the choices: disclosure, modification.

### FPT_ITT.2 TSF data transfer separation

#### User application notes

1223 One of the ways to achieve separation of TSF data based on SFP-relevant attributes is through the use of separate logical or physical channels.

#### Operations

##### Selection:

1224 In **FPT_ITT.2.1**, the PP/ST author should specify the desired type of protection to be provided from the choices: disclosure, modification.

### FPT_ITT.3 TSF data integrity monitoring

Operations

Selection:

1225 In **FPT_ITT.3.1**, the PP/ST author should specify the desired type of modification that the TSF shall be able to detect. The PP/ST author should select from: modification of data, substitution of data, re-ordering of data, deletion of data, or any other integrity errors.

Assignment:

1226 In **FPT_ITT.3.1**, if the PP/ST author chooses the latter selection noted in the preceding paragraph, then the author should also specify what those other integrity errors are that the TSF should be capable of detecting.

1227 In **FPT_ITT.3.2**, the PP/ST author should specify the action to be taken when an integrity error is identified.

## J.7 TSF physical protection (FPT_PHP)

User notes

1228 TSF physical protection components refer to restrictions on unauthorised physical access to the TSF, and to the deterrence of, and resistance to, unauthorised physical modification, or substitution of the TSF.

1229 The requirements in this family ensure that the TSF is protected from physical tampering and interference. Satisfying the requirements of these components results in the TSF being packaged and used in such a manner that physical tampering is detectable, or resistance to physical tampering is measurable based on defined work factors. Without these components, the protection functions of a TSF lose their effectiveness in environments where physical damage cannot be prevented. This component also provides requirements regarding how the TSF must respond to physical tampering attempts.

1230 Examples of physical tampering scenarios include mechanical attack, radiation, changing the temperature.

1231 It is acceptable for the functions that are available to an authorised user for detecting physical tampering to be available only in an off-line or maintenance mode. Controls should be in place to limit access during such modes to authorised users. As the TSF may not be "operational" during those modes, it may not be able to provide normal enforcement for authorised user access. The physical implementation of a TOE might consist of several structures: for example an outer shielding, cards, and chips. This set of "elements" as a whole must protect (protect, notify and resist) the TSF from physical tampering. This does not mean that all devices must provide these features, but the complete physical construct as a whole should.

1232 Although there is only minimal auditing associating with these components, this is solely because there is the potential that the detection and alarm mechanisms may be implemented completely in hardware, below the level of interaction with an audit subsystem (for example, a hardware-based detection system based on breaking a circuit and lighting a light emitting diode (LED) if the circuit is broken when a button is pressed by the authorised user). Nevertheless, a PP/ST author may determine that for a particular anticipated threat environment, there is a need to audit physical tampering. If this is the case, the PP/ST author should include appropriate requirements in the list of audit events. Note that inclusion of these requirements may have implications on the hardware design and its interface to the software.

## FPT_PHP.1 Passive detection of physical attack

### User application notes

1233 FPT_PHP.1 Passive detection of physical attack should be used when threats from unauthorised physical tampering with parts of the TOE are not countered by procedural methods. It addresses the threat of undetected physical tampering with the TSF. Typically, an authorised user would be given the function to verify whether tampering took place. As written, this component simply provides a TSF capability to detect tampering. Specification of management functions in FMT_MOF.1 Management of security functions behaviour should be considered to specify who can make use of that capability, and how they can make use of that capability. If this function is realised by non-IT mechanisms (e.g. physical inspection) management functions are not required.

## FPT_PHP.2 Notification of physical attack

### User application notes

1234 FPT_PHP.2 Notification of physical attack should be used when threats from unauthorised physical tampering with parts of the TOE are not countered by procedural methods, and it is required that designated individuals be notified of physical tampering. It addresses the threat that physical tampering with TSF elements, although detected, may not be noticed.

### Operations

#### Assignment:

1235 In **FPT_PHP.2.3**, the PP/ST author should provide a list of TSF devices/elements for which active detection of physical tampering is required.

1236 In **FPT_PHP.2.3**, the PP/ST author should designate a user or role that is to be notified when tampering is detected. The type of user or role may vary depending on the particular security administration component (from the FMT_MOF.1 Management of security functions behaviour family) included in the PP/ST.

### FPT_PHP.3 Resistance to physical attack

#### User application notes

1237 For some forms of tampering, it is necessary that the TSF not only detects the tampering, but actually resists it or delays the attacker.

1238 This component should be used when TSF devices and TSF elements are expected to operate in an environment where a physical tampering (e.g. observation, analysis, or modification) of the internals of a TSF device or TSF element itself is a threat.

#### Operations

##### Assignment:

1239 In **FPT_PHP.3.1**, the PP/ST author should specify tampering scenarios to a list of TSF devices/elements for which the TSF should resist physical tampering. This list may be applied to a defined subset of the TSF physical devices and elements based on considerations such as technology limitations and relative physical exposure of the device. Such subsetting should be clearly defined and justified. Furthermore, the TSF should automatically respond to physical tampering. The automatic response should be such that the policy of the device is preserved; for example, with a confidentiality policy, it would be acceptable to physically disable the device so that the protected information may not be retrieved.

1240 In **FPT_PHP.3.1**, the PP/ST author should specify the list of TSF devices/elements for which the TSF should resist physical tampering in the scenarios that have been identified.

## J.8 Trusted recovery (FPT_RCV)

#### User notes

1241 The requirements of this family ensure that the TSF can determine that the TOE is started-up without protection compromise and can recover without protection compromise after discontinuity of operations. This family is important because the start-up state of the TSF determines the protection of subsequent states.

1242 Recovery components reconstruct the TSF secure states, or prevent transitions to insecure states, as a direct response to occurrences of expected failures, discontinuity of operation or start-up. Failures that must be generally anticipated include the following:

a) Unmaskable action failures that always result in a system crash (e.g. persistent inconsistency of critical system tables, uncontrolled transfers within the TSF code caused by transient failures of hardware or firmware, power failures, processor failures, communication failures).

b) Media failures causing part or all of the media representing the TSF objects to become inaccessible or corrupt (e.g. parity errors, disk head crash, persistent read/write failure caused by misaligned disk heads, worn-out magnetic coating, dust on the disk surface).

c) Discontinuity of operation caused by erroneous administrative action or lack of timely administrative action (e.g. unexpected shutdowns by turning off power, ignoring the exhaustion of critical resources, inadequate installed configuration).

1243 Note that recovery may be from either a complete or partial failure scenario. Although a complete failure might occur in a monolithic operating system, it is less likely to occur in a distributed environment. In such environments, subsystems may fail, but other portions remain operational. Further, critical components may be redundant (disk mirroring, alternative routes), and checkpoints may be available. Thus, recovery is expressed in terms of recovery to a secure state.

1244 There are different interactions between Trusted recovery (FPT_RCV) and TSF self test (FPT_TST) components to be considered when selecting Trusted recovery (FPT_RCV):

a) The need for trusted recovery may be indicated through the results of TSF self-testing, where the results of the self-tests indicate that the TSF is in an insecure state and return to a secure state or entrance in maintenance mode is required.

b) A failure, as discussed above, may be identified by an administrator. Either the administrator may perform the actions to return the TOE to a secure state and then invoke TSF self-tests to confirm that the secure state has been achieved. Or, the TSF self-tests may be invoked to complete the recovery process.

c) A combination of a. and b. above, where the need for trusted recovery is indicated through the results of TSF self-testing, the administrator performs the actions to return the TOE to a secure state and then invokes TSF self-tests to confirm that the secure state has been achieved.

d) Self tests detect a failure/service discontinuity, then either automated recovery or entrance to a maintenance mode.

1245 This family identifies a maintenance mode. In this maintenance mode normal operation might be impossible or severely restricted, as otherwise insecure situations might occur. Typically, only authorised users should be allowed access to this mode but the real details of who can access this mode is a function of FMT: Security management. If FMT: Security management does not put any controls on who can access this mode, then it may be acceptable to allow any user to restore the system if the TOE enters such a state. However, in practice, this is probably not desirable as the user restoring the system has an opportunity to configure the TOE in such a way as to violate the TSP.

1246 Mechanisms designed to detect exceptional conditions during operation fall under TSF self test (FPT_TST), Fail secure (FPT_FLS), and other areas that address the concept of "Software Safety." It is likely that the use of one of these families will be required to support the adoption of Trusted recovery (FPT_RCV). This is to ensure that the TOE will be able to detect when recovery is required.

1247 Throughout this family, the phrase "secure state" is used. This refers to some state in which the TOE has consistent TSF data and a TSF that can correctly enforce the policy. This state may be the initial "boot" of a clean system, or it might be some checkpointed state. The "secure state" is defined in the TSP model. If the developer provided a clear definition of the secure state and the reason why it should be considered secure, the dependency from each of the components in Trusted recovery (FPT_RCV) to ADV_SPM.1 Informal TOE security policy model can be argued away.

1248 Following recovery, it may be necessary to confirm that the secure state has been achieved through self-testing of the TSF. However, if the recovery is performed in a manner such that only a secure state can be achieved, else recovery fails, then the dependency to the FPT_TST.1 TSF testing TSF self-test component may be argued away.

## FPT_RCV.1 Manual recovery

### User application notes

1249 In the hierarchy of the trusted recovery family, recovery that requires only manual intervention is the least desirable, for it precludes the use of the system in an unattended fashion.

1250 This component is intended for use in TOEs that do not require unattended recovery to a secure state. The requirements of this component reduce the threat of protection compromise resulting from an attended TOE returning to an insecure state after recovery from a failure or other discontinuity.

### Evaluator notes

1251 It is acceptable for the functions that are available to an authorised user for trusted recovery to be available only in a maintenance mode. Controls should be in place to limit access during maintenance to authorised users.

#### Operations

##### Assignment:

1252 In **FPT_RCV.1.1**, the PP/ST author should specify the list of failures or service discontinuities (e.g. power failure, audit storage exhaustion, any failure or discontinuity) following which the TOE will enter a maintenance mode.

### FPT_RCV.2 Automated recovery

#### User application notes

1253 Automated recovery is considered to be more useful than manual recovery, as it allows the machine to operate in an unattended fashion.

1254 The component FPT_RCV.2 Automated recovery extends the feature coverage of FPT_RCV.1 Manual recovery by requiring that there be at least one automated method of recovery from failure or service discontinuity. It addresses the threat of protection compromise resulting from an unattended TOE returning to an insecure state after recovery from a failure or other discontinuity.

#### Evaluator notes

1255 It is acceptable for the functions that are available to an authorised user for trusted recovery to be available only in a maintenance mode. Controls should be in place to limit access during maintenance to authorised users.

1256 For **FPT_RCV.2.1**, it is the responsibility of the developer of the TSF to determine the set of recoverable failures and service discontinuities.

1257 It is assumed that the robustness of the automated recovery mechanisms will be verified.

#### Operations

##### Assignment:

1258 In **FPT_RCV.2.1**, the PP/ST author should specify the list of failures or service discontinuities (e.g. power failure, audit storage exhaustion) following which the TOE will need to enter a maintenance mode.

1259 In **FPT_RCV.2.2**, the PP/ST author should specify the list of failures or other discontinuities for which automated recovery must be possible.

## FPT_RCV.3 Automated recovery without undue loss

### User application notes

1260 Automated recovery is considered to be more useful than manual recovery, but it runs the risk of losing a substantial number of objects. Preventing undue loss of objects provides additional utility to the recovery effort.

1261 The component FPT_RCV.3 Automated recovery without undue loss extends the feature coverage of FPT_RCV.2 Automated recovery by requiring that there not be undue loss of TSF data or objects within the TSC. At FPT_RCV.2 Automated recovery, the automated recovery mechanisms could conceivably recover by deleting all objects and returning the TSF to a known secure state. This type of drastic automated recovery is precluded in FPT_RCV.3 Automated recovery without undue loss.

1262 This component addresses the threat of protection compromise resulting from an unattended TOE returning to an insecure state after recovery from a failure or other discontinuity with a large loss of TSF data or objects within the TSC.

### Evaluator notes

1263 It is acceptable for the functions that are available to an authorised user for trusted recovery to be available only in a maintenance mode. Controls should be in place to limit access during maintenance to authorised users.

1264 It is assumed that the evaluators will verify the robustness of the automated recovery mechanisms.

### Operations

#### Assignment:

1265 In **FPT_RCV.3.1**, the PP/ST author should specify the list of failures or service discontinuities (e.g. power failure, audit storage exhaustion) following which the TOE will need to enter a maintenance mode.

1266 In **FPT_RCV.3.2**, the PP/ST author should specify the list of failures or other discontinuities for which automated recovery must be possible.

1267 In **FPT_RCV.3.3**, the PP/ST author should provide a quantification for the amount of loss of TSF data or objects that is acceptable.

## FPT_RCV.4 Function recovery

### User application notes

1268 Function recovery requires that if there should be some failure in the TSF, that certain SFs in the TSF should either complete successfully or recover to a secure state.

#### Operations

##### Assignment:

1269 In **FPT_RCV.4.1**, the PP/ST author should specify a list the SFs and failure scenarios. In the event that any of the identified failure scenarios happen, the SFs that have been specified must either complete successfully or recover to a consistent and secure state.

## J.9 Replay detection (FPT_RPL)

#### User notes

1270 This family addresses detection of replay for various types of entities and subsequent actions to correct.

### FPT_RPL.1 Replay detection

#### User application notes

1271 The entities included here are, for example, messages, service requests, service responses, or sessions.

#### Operations

##### Assignment:

1272 In **FPT_RPL.1.1**, the PP/ST author should provide a list of identified entities for which detection of replay should be possible. Examples of such entities might include: messages, service requests, service responses, and user sessions.

1273 In **FPT_RPL.1.2**, the PP/ST author should specify the list of actions to be taken by the TSF when replay is detected. The potential set of actions that can be taken includes: ignoring the replayed entity, requesting confirmation of the entity from the identified source, and terminating the subject from which the re-played entity originated.

## J.10 Reference mediation (FPT_RVM)

#### User notes

1274 The components of this family address the “always invoked” aspect of a traditional reference monitor. The goal of these components is to ensure, with respect to the TSC, that all actions requiring policy enforcement invoked by subjects untrusted with respect to any or all of that SFP to objects controlled by that SFP are validated by the TSF against the SFP. If the portion of the TSF that enforces the SFP also meets the requirements of appropriate components from Domain separation (FPT_SEP) and TSF internals (ADV_INT), than that portion of the TSF provides a “reference monitor” for that SFP.

1275 The Reference Monitor is that portion of the TSF responsible for the enforcement of the TSP; it has the following three characteristics:

a) Untrusted subjects cannot interfere with its operation; i.e. it is tamperproof. This is addressed by the components in the Domain separation (FPT_SEP) family.

b) Untrusted subjects cannot bypass its checks; i.e. it is always invoked. This is addressed by the components in the Reference mediation (FPT_RVM) family.

c) It is simple enough to be analysed and its behaviour understood (i.e. its design is conceptually simple.) This is addressed by the components in the TSF internals (ADV_INT) family.

1276 This component states that, “the TSF shall ensure that TSP enforcement functions are invoked and succeed before each and every function within the TSC is allowed to proceed.” In any system (distributed or otherwise) there are a finite number of functions responsible for enforcing the TSP. There is nothing in this requirement that mandates or prescribes that a single function is invoked to handle security. Rather, it allows multiple functions to fill the role of reference monitor, and the collection of them responsible for enforcing the TSP are simply called, collectively, the reference monitor. However, this must be balanced by the goal of keeping the “reference monitor” simple.

1277 A TSF that implements a SFP provides effective protection against unauthorised functions if and only if all enforceable actions (e.g. accesses to objects) requested by subjects untrusted with respect to any or all of that SFP are validated by the TSF before succeeding, If the enforceable action is incorrectly enforced or bypassed, the overall enforcement of the SFP has been compromised. “Untrusted” subjects could then bypass the SFP in a variety of unauthorised ways (e.g. circumvent access checks for some subjects or objects, bypass checks for objects whose protection was assumed by applications, retain access rights beyond their intended lifetime, bypass auditing of audited actions, or bypass authentication). Note that the term “untrusted subjects” refers to subjects untrusted with respect to any or all of the specific SFPs being enforced; a subject may be trusted with respect to one SFP and untrusted with respect to a different SFP.

## FPT_RVM.1 Non-bypassability of the TSP

### User application notes

1278 In order to obtain the equivalent of a reference monitor, this component must be used with either FPT_SEP.2 SFP domain separation or FPT_SEP.3 Complete reference monitor, and ADV_INT.3 Minimisation of complexity. Further, if complete reference mediation is required, the components from FDP: User data protection must cover all objects.

## J.11 Domain separation (FPT_SEP)

### User notes

1279 The components of this family ensure that at least one security domain is available for the TSF's own execution, and that the TSF is protected from external interference and tampering (e.g. by modification of TSF code or data structures) by untrusted subjects. Satisfying the requirements of this family makes the TSF self-protecting, meaning that an untrusted subject cannot modify or damage the TSF.

1280 This family requires the following:

a) The resources of the TSF's security domain (“protected domain”) and those of subjects and unconstrained entities external to the domain are separated such that the entities external to the protected domain cannot observe or modify data structures or code internal to the protected domain.

b) The transfer of subjects between domains are controlled such that arbitrary entry to, or return from, the protected domain is not possible.

c) The user or application parameters passed to the protected domain by addresses are validated with respect to the protected domain's address space, and those passed by value are validated with respect to the values expected by the protected domain.

d) The security domains of subjects are distinct except for controlled sharing via the TSF.

1281 This family is needed whenever confidence is required that the TSF has not been subverted.

1282 In order to obtain the equivalent of a reference monitor, the components FPT_SEP.2 SFP domain separation or FPT_SEP.3 Complete reference monitor from this family must be used in conjunction with FPT_RVM.1 Non-bypassability of the TSP, and ADV_INT.3 Minimisation of complexity. Further, if complete reference mediation is required, the components from FDP: User data protection must cover all objects.

### FPT_SEP.1 TSF domain separation

#### User application notes

1283 Without a separate protected domain for the TSF, there can be no assurance that the TSF has not been subjected to any tampering attacks by untrusted subjects. Such attacks may involve modification of the TSF code and/or TSF data structures.

## FPT_SEP.2 SFP domain separation

### User application notes

1284 The most important function provided by a TSF is the enforcement of its SFPs. In order to simplify the design and increase the likelihood that those significant SFPs exhibit the characteristics of a reference monitor (RM), in particular, being tamperproof, they must be in a domain distinct from the remainder of the TSF.

### Evaluator notes

1285 It is possible that a reference monitor in a layered design may provide functions beyond those of the SFPs. This arises out of the practical nature of layered software design. The goal should be to minimise the non-SFP related functions.

1286 Note that it is acceptable for the reference monitors for all included SFPs to be in a single distinct reference monitor domain, as well as having multiple reference monitor domains (each enforcing one or more SFPs). If multiple reference monitor domains for SFPs are present, it is acceptable for them to be either peers or in a hierarchical relationship.

1287 For **FPT_SEP.2.1**, the phrase "unisolated portion of the TSF" refers to that portion of the TSF consisting of those functions in the TSF not covered by **FPT_SEP.2.3**.

### Operations

#### Assignment:

1288 In **FPT_SEP.2.3**, the PP/ST author should specify the access control and/or information flow control SFPs in the TSP that should have a separate domain.

## FPT_SEP.3 Complete reference monitor

### User application notes

1289 The most important function provided by a TSF is the enforcement of its SFPs. This component builds upon the intentions of the previous component by requiring that all access control and/or information flow control FSPs be enforced in a domain distinct from the remainder of the TSF. This further simplifies the design and increases the likelihood that the characteristics of a reference monitor (RM), in particular, being tamperproof, are found in the TSF.

### Evaluator notes

1290 It is possible that a reference monitor in a layered design may provide functions beyond those of the SFPs. This arises out of the practical nature of layered software design. The goal should be to minimise the non-SFP related functions.

1291 Note that it is acceptable for the reference monitors for all included SFPs to be in a single distinct reference monitor domain, as well as having multiple reference monitor domains (each enforcing one or more SFPs). If multiple reference monitor domains for SFPs are present, it is acceptable for them to be either peers or in a hierarchical relationship.

## J.12 State synchrony protocol (FPT_SSP)

### User notes

1292 Distributed systems may give rise to greater complexity than monolithic systems through the potential for differences in state between parts of the system, and through delays in communication. In most cases, synchronisation of state between distributed functions involves an exchange protocol, not a simple action. When malice exists in the distributed environment of these protocols, more complex defensive protocols are required.

1293 State synchrony protocol (FPT_SSP) establishes the requirement for certain critical security functions of the TSF to use a trusted protocol. State synchrony protocol (FPT_SSP) ensures that two distributed parts of the TOE (e.g. hosts) have synchronised their states after a security-relevant action.

1294 Some states may never be synchronised, or the transaction cost may be too high for practical use; encryption key revocation is an example, where knowing the state after the revocation action is initiated can never be known. Either the action was taken and acknowledgment cannot be sent, or the message was ignored by hostile communication partners and the revocation never occurred. Indeterminacy is unique to distributed systems. Indeterminacy and state synchrony are related, and the same solution may apply. It is futile to design for indeterminate states; the PP/ST author should express other requirements in such cases (e.g. raise an alarm, audit the event).

### FPT_SSP.1 Simple trusted acknowledgement

#### User application notes

1295 In this component, the TSF must supply an acknowledgement to another part of the TSF when requested. This acknowledgement should indicate that one part of a distributed TOE successfully received an unmodified transmission from a different part of the distributed TOE.

### FPT_SSP.2 Mutual trusted acknowledgement

User application notes

1296 In this component, in addition to the TSF being able to provide an acknowledgement for the receipt of a data transmission, the TSF must comply with a request from another part of the TSF for an acknowledgement to the acknowledgement.

1297 For example, the local TSF transmits some data to a remote part of the TSF. The remote part of the TSF acknowledges the successful receipt of the data and requests that the sending TSF confirm that it receives the acknowledgement. This mechanism provides additional confidence that both parts of the TSF involved in the data transmission know that the transmission completed successfully.

## J.13 Time stamps (FPT_STM)

User notes

1298 This family addresses requirements for a reliable time stamp function within a TOE.

1299 It is the responsibility of the PP/ST author to clarify the meaning of the phrase “reliable time stamp”, and to indicate where the responsibility lies in determining the acceptance of trust.

### FPT_STM.1 Reliable time stamps

User application notes

1300 Some possible uses of this component include providing reliable time stamps for the purposes of audit as well as for security attribute expiration.

## J.14 Inter-TSF TSF data consistency (FPT_TDC)

User notes

1301 In a distributed or composite system environment, a TOE may need to exchange TSF data (e.g. the SFP-attributes associated with data, audit information, identification information) with another trusted IT Product, This family defines the requirements for sharing and consistent interpretation of these attributes between the TSF of the TOE and that of a different trusted IT Product.

1302 The components in this family are intended to provide requirements for automated support for TSF data consistency when such data is transmitted between the TSF of the TOE and another trusted IT Product. It is also possible that wholly procedural means could be used to produce security attribute consistency, but they are not provided for here.

1303 This family is different from FDP_ETC and FDP_ITC, as those two families are concerned only with resolving the security attributes between the TSF and its import/export medium.

1304 If the integrity of the TSF data is of concern, requirements should be chosen from the Integrity of exported TSF data (FPT_ITI) family. These components specify requirements for the TSF to be able to detect or detect and correct modifications to TSF data in transit.

### FPT_TDC.1 Inter-TSF basic TSF data consistency

#### User application notes

1305 The TSF is responsible for maintaining the consistency of TSF data used by or associated with the specified function and that are common between two or more trusted systems. For example, the TSF data of two different systems may have different conventions internally. For the TSF data to be used properly (e.g. to afford the user data the same protection as within the TOE) by the receiving trusted IT product, the TOE and the other trusted IT product must use a pre-established protocol to exchange TSF data.

#### Operations

##### Assignment:

1306 In **FPT_TDC.1.1**, the PP/ST author should define the list of TSF data types, for which the TSF shall provide the capability to consistently interpret, when shared between the TSF and another trusted IT product.

1307 In **FPT_TDC.1.2**, the PP/ST should assign the list of interpretation rules to be applied by the TSF,

## J.15 Internal TOE TSF data replication consistency (FPT_TRC)

#### User notes

1308 The requirements of this family are needed to ensure the consistency of TSF data when such data is replicated internal to the TOE. Such data may become inconsistent if an internal channel between parts of the TOE becomes inoperative. If the TOE is internally structured as a network of parts of the TOE, this can occur when parts become disabled, network connections are broken, and so on.

1309 The method of ensuring consistency is not specified in this component. It could be attained through a form of transaction logging (where appropriate transactions are "rolled back" to a site upon reconnection); it could be updating the replicated data through a synchronisation protocol. If a particular protocol is necessary for a PP/ST, it can be specified through refinement.

1310 It may be impossible to synchronise some states, or the cost of such synchronisation may be too high. Examples of this situation are communication channel and encryption key revocations. Indeterminate states may also occur; if a specific behaviour is desired, it should be specified via refinement.

### FPT_TRC.1 Internal TSF consistency

#### Operations

##### Assignment:

1311 In **FPT_TRC.1.2**, the PP/ST author should specify the list of SFs dependent on TSF data replication consistency.

## J.16 TSF self test (FPT_TST)

#### User notes

1312 The family defines the requirements for the self-testing of the TSF with respect to some expected correct operation. Examples are interfaces to enforcement functions, and sample arithmetical operations on critical parts of the TOE. These tests can be carried out at start-up, periodically, at the request of an authorised user, or when other conditions are met. The actions to be taken by the TOE as the result of self testing are defined in other families.

1313 The requirements of this family are also needed to detect the corruption of TSF executable code (i.e. TSF software) and TSF data by various failures that do not necessarily stop the TOE's operation (which would be handled by other families). These checks must be performed because these failures may not necessarily be prevented. Such failures can occur either because of unforeseen failure modes or associated oversights in the design of hardware, firmware, or software, or because of malicious corruption of the TSF due to inadequate logical and/or physical protection.

1314 In addition, use of this component may, with appropriate conditions, help to prevent inappropriate or damaging TSF changes being applied to an operational TOE as the result of maintenance activities.

1315 The term “correct operation of the TSF” refers primarily to the operation of the TSF software and the integrity of the TSF data. The abstract machine upon which the TSF software is implemented is tested via dependency on Underlying abstract machine test (FPT_AMT).

### FPT_TST.1 TSF testing

#### User application notes

1316 This component provides support for the testing of the critical functions of the TSF's operation by requiring the ability to invoke testing functions and check the integrity of TSF data and executable code.

### Evaluator notes

1317 It is acceptable for the functions that are available to the authorised user for periodic testing to be available only in an off-line or maintenance mode. Controls should be in place to limit access during these modes to authorised users.

### Operations

#### Selection:

1318 In **FPT_TST.1.1**, the PP/ST author should specify when the TSF will execute the TSF test; during initial start-up, periodically during normal operation, at the request of an authorised user, at other conditions. In the case of the latter option, the PP/ST author should also assign what those conditions are via the following assignment.

1319 In **FPT_TST.1.1**, the PP/ST author should specify whether the self tests are to be carried out to demonstrate the correct operation of the entire TSF, or of only specified parts of TSF.

#### Assignment:

1320 In **FPT_TST.1.1**, the PP/ST author should, if selected, specify the conditions under which the self test should take place.

1321 In **FPT_TST.1.1**, the PP/ST author should, if selected, specify the list of parts of the TSF that will be subject to TSF self-testing.

#### Selection:

1322 In **FPT_TST.1.2**, the PP/ST author should specify whether data integrity is to be verified for all TSF data, or only for selected data.

#### Assignment:

1323 In **FPT_TST.1.2**, the PP/ST author should, if selected, specify the list of TSF data that will be verified for integrity.

# K Class FRU: Resource utilisation (normative)

1324 This class provides three families that support the availability of required resources such as processing capability and/or storage capacity. The family Fault Tolerance provides protection against unavailability of capabilities caused by failure of the TOE. The family Priority of Service ensures that the resources will be allocated to the more important or time-critical tasks, and cannot be monopolised by lower priority tasks. The family Resource Allocation provides limits on the use of available resources, therefore preventing users from monopolising the resources.

1325 Figure 31 shows the decomposition of this class into its constituent components.

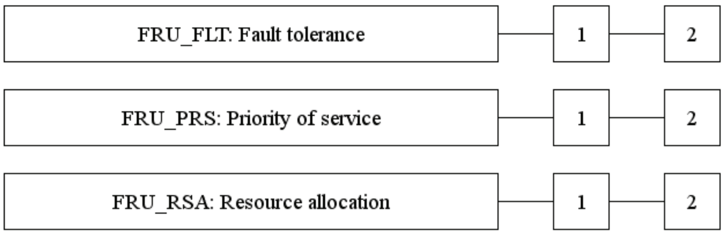



**Figure 31 - FRU: Resource utilisation class decomposition**

## K.1 Fault tolerance (FRU_FLT)

### User notes

1326 This family provides requirements for the availability of capabilities even in the case of failures. Examples of such failures are power failure, hardware failure, or software error. In case of these errors, if so specified, the TOE will maintain the specified capabilities. The PP/ST author could specify, for example, that a TOE used in a nuclear plant will continue the operation of the shut-down procedure in the case of power-failure or communication-failure.

1327 Because the TOE can only continue its correct operation if the TSP is enforced, there is a requirement that the system must remain in a secure state after a failure. This capability is provided by FPT_FLS.1 Failure with preservation of secure state.

1328 The mechanisms to provide fault tolerance could be active or passive. In case of an active mechanism, specific functions are in place that are activated in case the error occurs. For example, a fire alarm is an active mechanism: the TSF will detect the fire and can take action such as switching operation to a backup. In a passive scheme, the architecture of the TOE is capable of handling the error. For example, the use of a majority voting scheme with multiple processors is a passive solution; failure of one processor will not disrupt the operation of the TOE (although it needs to be detected to allow correction).

1329 For this family, it does not matter whether the failure has been initiated accidentally (such as flooding or unplugging the wrong device) or intentionally (such as monopolising).

## FRU_FLT.1 Degraded fault tolerance

### User application notes

1330 This component is intended to specify which capabilities the TOE will still provide after a failure of the system. Since it would be difficult to describe all specific failures, categories of failures may be specified. Examples of general failures are flooding of the computer room, short term power interruption, breakdown of a CPU or host, software failure, or buffer overflow.

### Operations

#### Assignment:

1331 In **FRU_FLT.1.1**, the PP/ST author should specify the list of TOE capabilities the TOE will maintain during and after a specified failure.

1332 In **FRU_FLT.1.1**, the PP/ST author should specify the list of type of failures against which the TOE has to be explicitly protected. If a failure in this list occurs, the TOE will be able to continue its operation.

## FRU_FLT.2 Limited fault tolerance

### User application notes

1333 This component is intended to specify against what type of failures the TOE must be resistant. Since it would be difficult to describe all specific failures, categories of failures may be specified. Examples of general failures are flooding of the computer room, short term power interruption, breakdown of a CPU or host, software failure, or overflow of buffer.

### Operations

#### Assignment:

1334 In **FRU_FLT.2.1**, the PP/ST author should specify the list of type of failures against which the TOE has to be explicitly protected. If a failure in this list occurs, the TOE will be able to continue its operation.

## K.2 Priority of service (FRU_PRS)

### User notes

1335 The requirements of this family allow the TSF to control the use of resources within the TSC by users and subjects such that high priority activities within the TSC will always be accomplished without interference or delay due to low priority activities. In other words, time critical tasks will not be delayed by tasks that are less time critical.

1336 This family could be applicable to several types of resources, for example, processing capacity, and communication channel capacity.

1337 The Priority of Service mechanism might be passive or active. In a passive Priority of Service system, the system will select the task with the highest priority when given a choice between two waiting applications. While using passive Priority of Service mechanisms, when a low priority task is running, it cannot be interrupted by a high priority task.While using an active Priority of Service mechanisms, lower priority tasks might be interrupted by new high priority tasks.

1338 The audit requirement states that all reasons for rejection should be audited. It is left to the developer to argue that an operation is not rejected but delayed.

### FRU_PRS.1 Limited priority of service

#### User application notes

1339 This component defines priorities for a subject, and the resources for which this priority will be used. If a subject attempts to take action on a resource controlled by the Priority of Service requirements, the access and/or time of access will be dependent on the subject's priority, the priority of the currently acting subject, and the priority of the subjects still in the queue.

#### Operations

##### Assignment:

1340 In **FRU_PRS.1.2**, the PP/ST author should specify the list of controlled resources for which the TSF enforces priority of service (e.g. resources such as processes, disk space, memory, bandwidth).

### FRU_PRS.2 Full priority of service

#### User application notes

1341 This component defines priorities for a subject. All shareable resources in the TSC will be subjected to the Priority of Service mechanism. If a subject attempts to take action on a shareable TSC resource, the access and/or time of access will be dependent on the subject's priority, the priority of the currently acting subject, and the priority of the subjects still in the queue.

## K.3 Resource allocation (FRU_RSA)

### User notes

1342 The requirements of this family allow the TSF to control the use of resources within the TSC by users and subjects such that unauthorised denial of service will not take place by means of monopolisation of resources by other users or subjects.

1343 Resource allocation rules allow the creation of quotas or other means of defining limits on the amount of resource space or time that may be allocated on behalf of a specific user or subjects. These rules may, for example:

- Provide for object quotas that constrain the number and/or size of objects a specific user may allocate.
- Control the allocation/deallocation of preassigned resource units where these units are under the control of the TSF.

1344 In general, these functions will be implemented through the use of attributes assigned to users and resources.

1345 The objective of these components is to ensure a certain amount of fairness among the users (e.g. a single user should not allocate all the available space) and subjects. Since resource allocation often goes beyond the lifespan of a subject (i.e. files often exist longer than the applications that generated them), and multiple instantiations of subjects by the same user should not negatively affect other users too much, the components allow that the allocation limits are related to the users. In some situations the resources are allocated by a subject (e.g. main memory or CPU cycles). In those instances the components allow that the resource allocation be on the level of subjects.

1346 This family imposes requirements on resource allocation, not on the use of the resource itself. The audit requirements therefore, as stated, also apply to the allocation of the resource, not to the use of the resource.

### FRU_RSA.1 Maximum quotas

#### User application notes

1347 This component provides requirements for quota mechanisms that apply to only a specified set of the shareable resources in the TOE. The requirements allow the quotas to be associated with a user, possibly assigned to groups of users or subjects as applicable to the TOE.

#### Operations

##### Assignment:

1348 In **FRU_RSA.1.1**, the PP/ST author should specify the list of controlled resources for which maximum resource allocation limits are required (e.g. processes, disk space, memory, bandwidth). If all resources in the TSC need to be included, the words “all TSC resources” can be specified.

Selection:

1349 In **FRU_RSA.1.1**, the PP/ST author should select whether the maximum quotas apply to individual users, to a defined group of users, or subjects or any combination of these.

1350 In **FRU_RSA.1.1**, the PP/ST author should select whether the maximum quotas are applicable to any given time (simultaneously), or over a specific time interval.

## FRU_RSA.2 Minimum and maximum quotas

### User application notes

1351 This component provides requirements for quota mechanisms that apply to a specified set of the shareable resources in the TOE. The requirements allow the quotas to be associated with a user, or possibly assigned to groups of users as applicable to the TOE.

### Operations

Assignment:

1352 In **FRU_RSA.2.1**, the PP/ST author should specify the controlled resources for which maximum and minimum resource allocation limits are required (e.g. processes, disk space, memory, bandwidth). If all resources in the TSC need to be included, the words “all TSC resources” can be specified.

Selection:

1353 In **FRU_RSA.2.1**, the PP/ST author should select whether the maximum quotas apply to individual users, to a defined group of users, or subjects or any combination of these.

1354 In **FRU_RSA.2.1**, the PP/ST author should select whether the maximum quotas are applicable to any given time (simultaneously), or over a specific time interval.

Assignment:

1355 In **FRU_RSA.2.2**, the PP/ST author should specify the controlled resources for which a minimum allocation limit needs to be set (e.g. processes, disk space, memory, bandwidth). If all resources in the TSC need to be included the words “all TSC resources” can be specified.

Selection:

1356 In **FRU_RSA.2.2**, the PP/ST author should select whether the minimum quotas apply to individual users, to a defined group of users, or subjects or any combination of these.

1357 In **FRU_RSA.2.2**, the PP/ST author should select whether the minimum quotas are applicable to any given time (simultaneously), or over a specific time interval.

# L Class FTA: TOE access (normative)

1358 The establishment of a user's session typically consists of the creation of one or more subjects that perform operations in the TOE on behalf of the user. At the end of the session establishment procedure, provided the TOE access requirements are satisfied, the created subjects bear the attributes determined by the identification and authentication functions. This family specifies functional requirements for controlling the establishment of a user's session.

1359 A user session is defined as the period starting at the time of the identification/authentication, or if more appropriate, the start of an interaction between the user and the system, up to the moment that all subjects (resources and attributes) related to that session have been deallocated.

1360 Figure 32 shows the decomposition of this class into its constituent components.

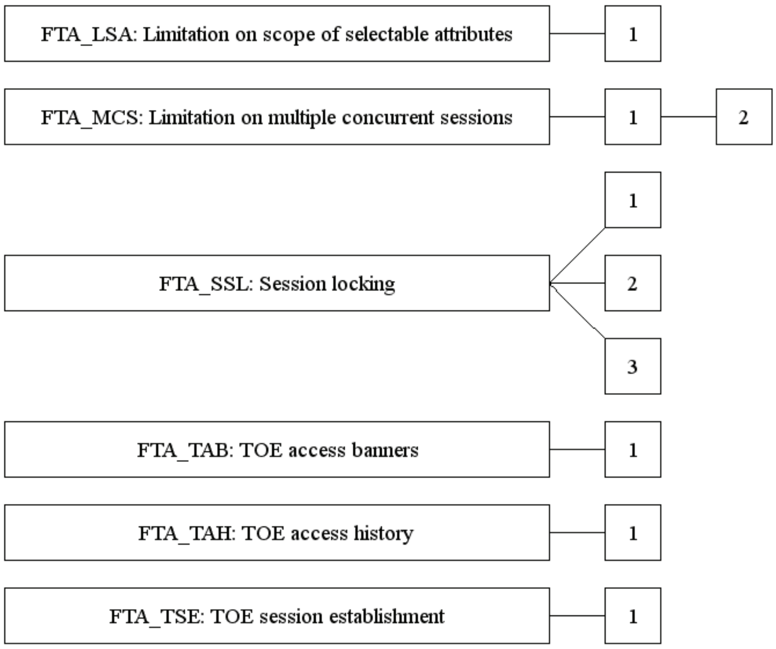



**Figure 32 - FTA: TOE access class decomposition**

## L.1 Limitation on scope of selectable attributes (FTA_LSA)

### User notes

1361 This family defines requirements that will limit the session security attributes a user may select, and the subjects to which a user may be bound, based on: the method of access; the location or port of access; and/or the time (e.g. time-of-day, day-of-week).

1362 This family provides the capability for a PP/ST author to specify requirements for the TSF to place limits on the domain of an authorised user's security attributes based on an environmental condition. For example, a user may be allowed to establish a “secret session” during normal business hours but outside those hours the same user may be constrained to only establishing “unclassified sessions”. The identification of relevant constraints on the domain of selectable attributes can be achieved through the use of the selection operation. These constraints can be applied on an attribute-by-attribute basis. When there exists a need to specify constraints on multiple attributes this component will have to be replicated for each attribute. Examples of attributes that could be used to limit the session security attributes are:

a) The method of access can be used to specify in which type of environment the user will be operating (e.g. file transfer protocol, terminal, vtam).

b) The location of access can be used to constrain the domain of a user's selectable attributes based on a user's location or port of access. This capability is of particular use in environments where dial-up facilities or network facilities are available.

c) The time of access can be used to constrain the domain of a user's selectable attributes. For example, ranges may be based upon time-of-day, day-of-week, or calendar dates. This constraint provides some operational protection against user actions that could occur at a time where proper monitoring or where proper procedural measures may not be in place.

## FTA_LSA.1 Limitation on scope of selectable attributes

### Operations

#### Assignment:

1363 In **FTA_LSA.1.1**, the PP/ST author should specify the set of session security attributes that are to be constrained. Examples of these session security attributes are user clearance level, integrity level and roles.

1364 In **FTA_LSA.1.1**, the PP/ST author should specify the set of attributes that can be use to determine the scope of the session security attributes. Examples of such attributes are user identity, originating location, time of access, and method of access.

## L.2 Limitation on multiple concurrent sessions (FTA_MCS)

### User notes

1365 This family defines how many sessions a user may have at the same time (concurrent sessions). This number of concurrent sessions can either be set for a group of users or for each individual user.

### FTA_MCS.1 Basic limitation on multiple concurrent sessions

#### User application notes

1366 This component allows the system to limit the number of sessions in order to effectively use the resources of the TOE.

#### Operations

##### Assignment:

1367 In **FTA_MCS.1.2**, the PP/ST author should specify the default number of maximum concurrent sessions to be used.

### FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions

#### User application notes

1368 This component provides additional capabilities over those of FTA_MCS.1 Basic limitation on multiple concurrent sessions, by allowing further constraints to be placed on the number of concurrent sessions that users are able to invoke. These constraints are in terms of a user's security attributes, such as a user's identity, or membership of a role.

#### Operations

##### Assignment:

1369 In **FTA_MCS.2.1**, the PP/ST author should specify the rules that determine the maximum number of concurrent sessions. An example of a rule is “maximum number of concurrent sessions is one if the user has a classification level of “secret” and five otherwise”.

1370 In **FTA_MCS.2.2**, the PP/ST author should specify the default number of maximum concurrent sessions to be used.

## L.3 Session locking (FTA_SSL)

### User notes

1371 This family defines requirements for the TSF to provide the capability for locking and unlocking of interactive sessions (e.g. keyboard locking).

1372 When a user is directly interacting with subjects in the TOE (interactive session), the user's terminal is vulnerable if left unattended. This family provides requirements for the TSF to disable (lock) the terminal or terminate the session after a specified period of inactivity, and for the user to initiate the disabling (locking) of the terminal. To reactivate the terminal, an event specified by the PP/ST author, such as the user re-authentication must occur.

1373 A user is considered inactive, if he/she has not provided any stimulus to the TOE for a period of time.

1374 A PP/ST author should consider whether FTP_TRP.1 Trusted path should be included. In that case, the function "session locking" should be included in the operation in FTP_TRP.1 Trusted path.

## FTA_SSL.1 TSF-initiated session locking

### User application notes

1375 FTA_SSL.1 TSF-initiated session locking, provides the capability for the TSF to lock an active user session after a specified period of time. Locking a terminal would prevent any further interaction with an existing active session through the use of the locked terminal.

1376 If display devices are overwritten, the replacement contents need not be static (i.e. "screen savers" are permitted).

1377 This component allows the PP/ST author to specify what events will unlock the session. These events may be related to the terminal (e.g. fixed set of keystrokes to unlock the session), the user (e.g. reauthentication), or time.

### Operations

#### Assignment:

1378 In **FTA_SSL.1.1**, the PP/ST author should specify the interval of user inactivity that will trigger the locking of an interactive session. If so desired the PP/ST author could, through the assignment, specify that the time interval is left to the authorised administrator or the user. The management functions in the FMT class can specify the capability to modify this time interval, making it the default value.

1379 In **FTA_SSL.1.2**, the PP/ST author should specify the event(s) that should occur before the session is unlocked. Examples of such an event are: "user re-authentication" or "user enters unlock key-sequence".

## FTA_SSL.2 User-initiated locking

### User application notes

1380 FTA_SSL.2 User-initiated locking, provides the capability for an authorised user to lock and unlock his/her own terminal. This would provide authorised users with the ability to effectively block further use of their active sessions without having to terminate the active session.

1381 If devices are overwritten, the replacement contents need not be static (i.e. "screen savers" are permitted).

Operations

Assignment:

1382 In **FTA_SSL.2.2**, the PP/ST author should specify the event(s) that should occur before the session is unlocked. Examples of such an event are: "user re-authentication", or "user enters unlock key-sequence".

### FTA_SSL.3 TSF-initiated termination

User application notes

1383 FTA_SSL.3 TSF-initiated termination, requires that the TSF terminate an interactive user session after a period of inactivity.

1384 The PP/ST author should be aware that a session may continue after the user terminated his/her activity, for example, background processing. This requirement would terminate this background subject after a period of inactivity of the user without regard to the status of the subject.

Operations

Assignment:

1385 In **FTA_SSL.3.1**, the PP/ST author should specify the interval of user inactivity that will trigger the termination of an interactive session. If so desired, the PP/ST author could, through the assignment, specify that the interval is left to the authorised administrator or the user. The management functions in the FMT class can specify the capability to modify this time interval, making it the default value.

## L.4 TOE access banners (FTA_TAB)

User notes

1386 Prior to identification and authentication, TOE access requirements provide the ability for the TOE to display an advisory warning message to potential users pertaining to appropriate use of the TOE.

### FTA_TAB.1 Default TOE access banners

#### User application notes

1387 This component requires that there is an advisory warning regarding the unauthorised use of the TOE. A PP/ST author could refine the requirement to include a default banner.

## L.5 TOE access history (FTA_TAH)

#### User notes

1388 This family defines requirements for the TSF to display to users, upon successful session establishment to the TOE, a history of unsuccessful attempts to access the account. This history may include the date, time, means of access, and port of the last successful access to the TOE, as well as the number of unsuccessful attempts to access the TOE since the last successful access by the identified user.

### FTA_TAH.1 TOE access history

#### User application notes

1389 This family can provide authorised users with information that may indicate the possible misuse of their user account.

1390 This component request that the user is presented with the information. The user should be able to review the information, but is not forced to do so. If a user so desires he might, for example, create scripts that ignore this information and start other processes.

#### Operations

##### Selection:

1391 In **FTA_TAH.1.1**, the PP/ST author should select the security attributes of the last successful session establishment that will be shown at the user interface. The items are: date, time, method of access (such as ftp), and/or location (e.g. terminal 50).

1392 In **FTA_TAH.1.2**, the PP/ST author should select the security attributes of the last unsuccessful session establishment that will be shown at the user interface. The items are: date, time, method of access (such as ftp), and/or location (e.g. terminal 50).

## L.6 TOE session establishment (FTA_TSE)

#### User notes

1393 This family defines requirements to deny an user permission to establish a session with the TOE based on attributes such as the location or port of access, the user's security attribute (e.g. identity, clearance level, integrity level, membership in a role), ranges of time (e.g. time-of-day, day-of-week, calendar dates) or combinations of parameters.

1394 This family provides the capability for the PP/ST author to specify requirements for the TOE to place constraints on the ability of an authorised user to establish a session with the TOE. The identification of relevant constraints can be achieved through the use of the selection operation. Examples of attributes that could be used to specify the session establishment constraints are:

a) The location of access can be used to constrain the ability of a user to establish an active session with the TOE, based on the user's location or port of access. This capability is of particular use in environments where dial-up facilities or network facilities are available.

b) The user's security attributes can be used to place constraints on the ability of a user to establish an active session with the TOE. For example, these attributes would provide the capability to deny session establishment based on any of the following:

   – a user's identity;

   – a user's clearance level;

   – a user's integrity level; and

   – a user's membership in a role.

1395 This capability is particularly relevant in situations where authorisation or login may take place at a different location from where TOE access checks are performed.

a) The time of access can be used to constrain the ability of a user to establish an active session with the TOE based on ranges of time. For example, ranges may be based upon time-of-day, day-of-week, or calendar dates. This constraint provides some operational protection against actions that could occur at a time where proper monitoring or where proper procedural measures may not be in place.

## FTA_TSE.1 TOE session establishment

### Operations

#### Assignment:

1396 In **FTA_TSE.1.1**, the PP/ST author should specify the attributes that can be used to restrict the session establishment. Example of possible attributes are user identity, originating location (e.g. no remote terminals), time of access (e.g. outside hours), or method of access (e.g. X-windows).

# M Class FTP: Trusted path/channels (normative)

1397 Users often need to perform functions through direct interaction with the TSF. A trusted path provides confidence that a user is communicating directly with the TSF whenever it is invoked. A user's response via the trusted path guarantees that untrusted applications cannot intercept or modify the user's response. Similarly, trusted channels are one approach for secure communication between the TSF and remote IT products.

1398 Figure 2 of this part of the CC illustrates the relationships between the various types of communication that may occur within a TOE or network of TOEs (i.e. Internal TOE transfers, Inter-TSF transfers, and Import/Export Outside of TSF Control) and the various forms of trusted paths and channels.

1399 Absence of a trusted path may allow breaches of accountability or access control in environments where untrusted applications are used. These applications can intercept user-private information, such as passwords, and use it to impersonate other users. As a consequence, responsibility for any system actions cannot be reliably assigned to an accountable entity. Also, these applications could output erroneous information on an unsuspecting user's display, resulting in subsequent user actions that may be erroneous and may lead to a security breach.

1400 Figure 33 shows the decomposition of this class into its constituent components.

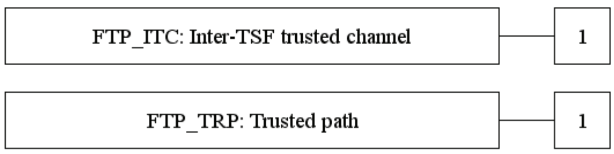



**Figure 33 - FTP: Trusted path/channels class decomposition**

## M.1 Inter-TSF trusted channel (FTP_ITC)

User notes

1401 This family defines the rules for the creation of a trusted channel connection that goes between the TSF and another trusted IT product for the performance of security critical operations between the products. An example of such a security critical operation is the updating of the TSF authentication database by the transfer of data from a trusted product whose function is the collection of audit data.

### FTP_ITC.1 Inter-TSF trusted channel

#### User application notes

1402 This component should be used when a trusted communication channel between the TSF and another trusted IT product is required.

#### Operations

##### Selection:

1403 In **FTP_ITC.1.2**, the PP/ST author must specify whether the local TSF, the remote trusted IT product, or both shall have the capability to initiate the trusted channel.

##### Assignment:

1404 In **FTP_ITC.1.3**, the PP/ST author should specify the functions for which a trusted channel is required. Examples of these functions may include transfer of user, subject, and/or object security attributes and ensuring consistency of TSF data.

## M.2 Trusted path (FTP_TRP)

#### User notes

1405 This family defines the requirements to establish and maintain trusted communication to or from users and the TSF. A trusted path may be required for any security-relevant interaction. Trusted path exchanges may be initiated by a user during an interaction with the TSF, or the TSF may establish communication with the user via a trusted path.

### FTP_TRP.1 Trusted path

#### User application notes

1406 This component should be used when trusted communication between a user and the TSF is required, either for initial authentication purposes only or for additional specified user operations.

#### Operations

##### Selection:

1407 In **FTP_TRP.1.1**, the PP/ST author should specify whether the trusted path must be extended to remote and/or local users.

1408 In **FTP_TRP.1.2**, the PP/ST author should specify whether the TSF, local users, and/or remote users should be able to initiate the trusted path.

1409 In **FTP_TRP.1.3**, the PP/ST author should specify whether the trusted path is to be used for initial user authentication and/or for other specified services.

### Assignment:

1410 In **FTP_TRP.1.3**, if selected, the PP/ST author should identify other services for which trusted path is required, if any.