COMMITTEE DRAFT **ISO/IEC 3rd CD 15408-2, revision**	Reference document: **SC 27 N19506**
Date: **2019-07-12**	Supersedes document N18804

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

ISO/IEC JTC 1/SC 27 Information security, cybersecurity and privacy protection Secretariat: Germany	Circulated to P- and O-members, and to technical committees and organizations in liaison for comments by: **2019-09-06** Please submit your comments via the online balloting application by the due date indicated.

ISO/IEC 3rd CD 15408-2, revision
Title: IT Security techniques – Evaluation criteria for IT security — Part 2: Security functional components
Project: 1.27.16.02 (ISO/IEC 15408-2, revision)

Explanatory Report

Status	**SC 27 Decision**	**Reference documents**	
		Input	**Output**
For details regarding previous development stages refer to 2nd page of this explanatory report.			

ISO/IEC 15408-2
1st WD
	54th WG 3 meeting, April 2017, Recommendations 5,10 11, 14 (N17041 = WG 3 N1413).	Results of call f. editor (N17276);
		SoV (N17026).
			Call f. project editor (N17319);
			Liaisons to:
			CCDB (WG 3 N1391);
			The Open Group (WG 3 N1394);
			ISO/TC 22/SC 32 (N17373);
			Text f. 1st WD (WG 3 N1436).

ISO/IEC NP 15408-2 (revision)
2nd WD
	55th WG 3 meeting, October / November 2017, Recommendations 8, 10 (N17666 = WG 3 N1494).	Results of call f. editor (N17389);
		SoCom (WG 3 N1464);
		Draft DoC (WG 3 N1501).
			Call / NB nomination for /of (N17319 / N17389);
			Editor's report (WG 3 N1465);
			Liaisons to:
			CCDB (WG 3 N1455);
			ISO/TC 22/SC 32 (N18103);
			DoC (WG 3 N1462);
			Text f. 2nd WD (WG 3 N1466).

ISO/IEC 15408-2
1st CD
	56th WG 3 meeting, April 2018, Recommendations 8, 10 (N18471 = WG 3 N1557) / 30th SC 27 Plenary, April 2018, Resolution 6 (N18710)	SoCom (WG 3 N1528);
		Late Com (WG 3 N1563).
			Liaison to:
			CCDB (WG 3 N1521);
			DoC (WG 3 N1527);
			Text f. 1st CD (N18701).

ISO/IEC 15408-2
2nd CD
	57th WG 3 meeting / CRM, Sep / Oct 2018, Recommendations 8, 10 (N18471 = WG 3 N1557) / 30th SC 27 Plenary, April 2018, Resolution 6 (N18710)	SoV (N18852);
		Draft DoC (N18924).
			Liaison to:
			CCDB (WG 3 N1619);
			DoC (N18802);
			Text f. 2nd CD (N18804).

ISO/IEC 15408-2
3rd CD
	58th WG 3 meeting / CRM April 2019, Recommendations 12, 14, 17, 21 (N19523 = WG 3 N1676).	SoV (N19488).	Liaison to:
			CCDB (WG 3 N1680);
			DoC (N19504);
			Text f. 3rd CD (N19506).

3rd CD Consideration
In accordance with Recommendation 14 (see SC 27 N19523) of the 58th SC 27/WG 3 meeting / CRM held in Tel Aviv, Israel, 2019-04-01/05 the hereby attached document is circulated for a 8-week 3rd CD letter ballot closing by

2019-09-06

Medium: http://isotc.iso.org/livelink/livelink/open/jtc1sc27

No. of pages: 2 + 283

Secretariat, ISO/IEC JTC 1/SC27 -
DIN Deutsches Institut für Normung e.V., Saatwinkler Damm 42/43, D-13627 [D-10772 postal] Berlin , Germany
Telephone: + 49 2601-2652; Facsimile: + 49 2601-42652; E-mail: krystyna.passia@din.de, http://www.din.de/jtc1sc27

Explanatory Report (2nd page)			
Status	**SC 27 Decision**	**Reference documents**	
		Input	**Output**
Study Period **IT security testing, evaluation and assurance standards and techniques**	51st WG 3 meeting, Oct. 2015, Recommendations 5, 6 (N15594 = WG 3 N1251).		Terms of Reference (WG 5 N1258); 1st /2nd call f. contr. (WG 3 N1259 /1317)..

	52nd WG 3 meeting, April 2016, Recommendation 5, 7 (N16026 = WG 3 N1296).	Expert contr. (WG 3 N1299, 1301).	3rd call f. contr. (WG 3 N1377);
			Rapporteur's report (WG 3 N1320);
			Liaison to:
			CCDB (WG 3 N1266).

ISO/IEC NP 15408-2 (revision)
Evaluation criteria for IT security -- Part 2
NWIP
	53rd WG 3 meeting, Oct. 2016, Recommendations 6, 15 (N16800 = WG 5 N600).	Expert contr. (WG 3 N1368, N1371, N1373).	SP report (WG 3 N1363);
			Call f. editor (WG 3 N1387 = N16886);
			Liaisons to:
			CCDB (WG 3 N1330);
			The Open Group (WG 3 N1332);
			Text f. NWIP (N16964 [replaces N16883]).

ISO/IEC JTC 1/SC 27 N19506

Date: 2019-07-12

ISO/IEC CD 15408-2:####(EN)

ISO/IEC JTC 1/SC 27 IT Security techniques

Secretariat: DIN

IT security techniques — Evaluation criteria for IT security — Part 2: Security functional components

Techniques de sécurité IT — Critères d'évaluation pour a sécurité des technologies de l'information — Partie 2 : Composants fonctionnels de sécurité

CD stage

Document type: International Standard

Document subtype:

Document stage: (30.20) Preparatory

Document language: E

READ ME FIRST

Editors general notes for this draft.

Red text in a box are the Editors comments.

In this draft the editors highlighted the keywords relating to the ISO verbal forms, shall, should, may, can and must using green text in order to highlight these words. This convention will be removed before the FDIS level documents.

The editors are aware that the figures are of low quality. In the final documents high quality images will be used. The Editors hope that they are legible in this draft

The Editors thank the WG 3 contributors for their contributions and support during the editing cycle.

Legal Notice:

The text for the legal notice agreed between ISO/IEC and the CCDB will be included here.

Contents Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see http://www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see http://www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) (see http://www.iso.org/iso/foreword.html).

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

A list of all parts in the ISO/IEC 15408 series can be found on the ISO website.

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

This fourth edition cancels and replaces the third edition (ISO 15408-2:2008), which has been technically revised.

The main changes compared to the previous edition are as follows:

- The document has been revised to comply with ISO/IEC Directives
- Technical changes have been introduced:
  - New security functional components have been introduced.

Introduction

Security functional components, as defined in this document, are the basis for the security functional requirements or components expressed in a Protection Profile (PP), PP-Module, functional package or a Security Target (ST). These requirements describe the desired security behaviour expected of a Target of Evaluation (TOE) and are intended to meet the security objectives as stated in a PP, PP-Module, functional package 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.

Security functional components allow for the expression of security functional requirements intended to counter threats in the assumed operating environment of the TOE and/or cover any identified organizational security policies.

The audience for this document includes consumers, developers, and evaluators of secure IT products. ISO/IEC 15408-1:20XX, Clause 5.3 provides additional information on the target audience of the ISO/IEC 15408 series, and on the use of the ISO/IEC 15408 series by the groups that comprise the target audience. These groups use this document as follows:

a) Consumers, who use this document when selecting components to express functional requirements which satisfy the security objectives expressed in a PP, PP-Module, functional package or ST. ISO/IEC 15408-1:20XX, Clause 7 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, will find a standardized method to understand those requirements in this document. They should also use the contents of this document as a basis for further defining the TOE security functionality and mechanisms that comply with those requirements.

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

IT Security techniques — Evaluation criteria for IT security — Part 2: Security functional components

1 Scope

This document 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.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 15408-1:20XX, *IT Security techniques — Evaluation criteria for IT security — Part 1: Introduction and general model*

ISO/IEC 15408-3:20XX, *IT Security techniques — Evaluation criteria for IT security — Part 3: Security assurance components*

3 Terms and Definitions

For the purposes of this document, the terms, definitions, and abbreviated terms given in ISO/IEC 15408-1:20XX apply.

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

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

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

4 Overview

4.1 General

ISO/IEC 15408 (all parts) and the associated security functional requirements described in this document are not intended to be a definitive answer to all the problems of IT security. This document offers a set of well understood security functional components that can be used to specify trusted products reflecting the needs of the market. These security functional components are presented as the current state of the art in security requirements specification.

This document does not include all possible security functional components but contains those that are known and agreed to be of value by this the contributors to this document.

Since the understanding and needs of consumers can change, the functional components in this document will need to be maintained. It is envisioned that some authors of PPs, PP-Modules, functional packages and STs could have security needs not covered by the security functional components in this document. In those cases, the PP, PP-Module, functional package or ST author may choose to consider using functional components and requirements that are not given in this document. The concepts of extensibility are explained in Annex D of ISO/IEC 15408-1:20XX.

4.2 Organization of this document

Clause 5 describes the paradigm used in the security functional requirements of this document.

Clause 6 introduces the catalogue of functional components while clauses 7 through 17 describe the functional classes.

Annex A provides explanatory information for potential users of the functional components.

Annex B provides a complete cross reference table of the functional component dependencies.

Annex C through Annex M provide the explanatory information for the functional classes. This material shall 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, PP-Module, functional package and ST author.

Those who author PPs, PP-Modules, functional packages, or STs shall refer to ISO/IEC 15408-1:20XX for relevant structures, rules, and guidance, in particular:

a) ISO/IEC 15408-1:20XX, Clause 3 defines the terms and definitions used in ISO/IEC 15408 (all parts).

b) ISO/IEC 15408-1:20XX, Clause 7 describes how security functional requirements can be specified using the security functional components.

c) ISO/IEC 15408-1:20XX, Clause 8 describes how security functional components are organized, and the operations that may be applied to them.

d) ISO/IEC 15408-1:20XX, Annex A provides further guidance on the structure for security functional packages.

e) ISO/IEC 15408-1:20XX, Annex B provides further guidance on the structure for PPs.

f) ISO/IEC 15408-1:20XX, Annex C provides further guidance on the structure of PP-Modules and PP-Configurations.

g) ISO/IEC 15408-1:20XX, Annex D provides further guidance on the structure for STs.

5 Functional requirements paradigm

Clause 5 describes the paradigm used in the security functional components and the derivation of security functional requirements. The key concepts discussed are highlighted in bold/italics.

This document is a catalogue of security functional components that may be used for the specification of security functional requirements describing a **Target of Evaluation (TOE)**.

TOE evaluation is concerned primarily with ensuring that a defined set of **security functional requirements (SFRs)** is enforced over the TOE resources. The SFRs define the rules by which the TOE governs access to and use of its resources, and thus information and services controlled by the TOE.

The SFRs may define multiple **Security Function Policies** (**SFPs**) to represent the rules that the TOE must enforce. Each SFP specifies its **scope of control**, by defining the subjects, objects, resources or information, and operations to which it applies. All SFPs are implemented by the TSF (see below), whose mechanisms enforce the rules defined in the SFRs and provide necessary capabilities.

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

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

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 SFRs.

TOE interfaces may be localized 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 SFRs of the other “trusted IT product” and the SFRs of the TOE have been administratively coordinated and the other trusted IT product is assumed to enforce its SFRs correctly (e. g. by being separately evaluated). Exchanges of information in this situation are called **inter-TSF transfers**, as they are between the TSFs of distinct trusted products.

b) The other IT product may not be trusted, it may be called an “untrusted IT product”. Therefore, its SFRs are either unknown or their implementation is not viewed as trustworthy. TSF mediated exchanges of information in this situation are called **transfers outside of the TOE**, as there is no TSF (or its policy characteristics are unknown) on the other IT product.

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 functionality that provide for the enforcement of the SFRs.

Users are outside of the TOE. However, in order to request that services be performed by the TOE that are subject to rules defined in the SFRs, users interact with the TOE through the TSFIs. There are two types of users of interest to this document: **human users** and **external IT entities**. Human users may further be differentiated as **local human users**, meaning they interact directly with the TOE via TOE devices or **remote human users**, meaning they interact indirectly with the TOE through another IT product.

EXAMPLE 1

An example of a TOE device is a workstation.

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.

EXAMPLE 2

user authentication, time of day, method of accessing the TOE, and number of allowed concurrent sessions (per user or in total).

This document uses the term **authorized** to signify a user who possesses the rights and/or privileges necessary to perform an operation. The term **authorized user**, therefore, indicates that it is allowable for a user to perform a specific operation or a set of operations as defined by the SFRs.

To express requirements that call for the separation of administrator duties, the relevant 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 operating in that role and the TOE. A TOE may support the definition of any number of roles.

EXAMPLE 3

Roles related to the secure operation of a TOE may include “Audit Administrator” and “User Accounts Administrator”.

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 SFRs over the resources and information that the TOE controls.

TOE resources can be structured and utilized in many different ways. However, this document makes a specific distinction that allows for the specification of desired security properties. All entities that can be created from resources can be characterized 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.

Active entities in the TOE that perform operations on objects are referred to as **subjects**. Several types of subjects may exist within a TOE:

a) those acting on behalf of an authorized user;

   EXAMPLE 4

   UNIX processes

b) those acting as a specific functional process that may in turn act on behalf of multiple users;

   EXAMPLE 5

   functions as might be found in client/server architectures

c) those acting as part of the TOE itself.

   EXAMPLE 6

   processes not acting on behalf of a user

This document addresses the enforcement of the SFRs over types of subjects as those listed above.

Passive entities in the TOE that contain or receive information and upon which subjects perform operations are called **objects**. In the case where a subject (an active entity) is the target of an operation, a subject may also be acted on as an object.

EXAMPLE 7

An example of a subject is an inter-process communication

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

Users, subjects, information, objects, sessions, and resources controlled by rules in the SFRs may possess certain **attributes** that contain information that is used by the TOE for its correct operation. Some attributes, such as file names, may be intended to be informational or may be used to identify individual resources while others, such as access control information, may exist specifically for the enforcement of the SFRs. These latter attributes are generally referred to as “**security attributes**”. The word attribute will be used as a shorthand in some places in this document for the term “security attribute”. However, no matter what the intended purpose of the attribute information, it may be necessary to have controls on attributes as dictated by the SFRs.

Data in a TOE is categorized as either user data or TSF data. Figure 1 depicts this relationship. **User Data** is information stored in TOE resources that can be operated upon by users in accordance with the SFRs and upon which the TSF places no special meaning. **TSF Data** is information used by the TSF in making decisions as required by the SFRs. TSF Data may be influenced by users if allowed by the SFRs.

EXAMPLE 8

User data:

- The content of an electronic mail message can be user data.

TSF data:

- Security attributes, authentication data, TSF internal status variables used by the rules defined in the SFRs or used for the protection of the TSF and access control list entries are examples of TSF data.

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 users, resources, subjects, objects, sessions, TSF status data 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.

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 be derived from the data in the container, stay with the information as it is processed by the TSF.

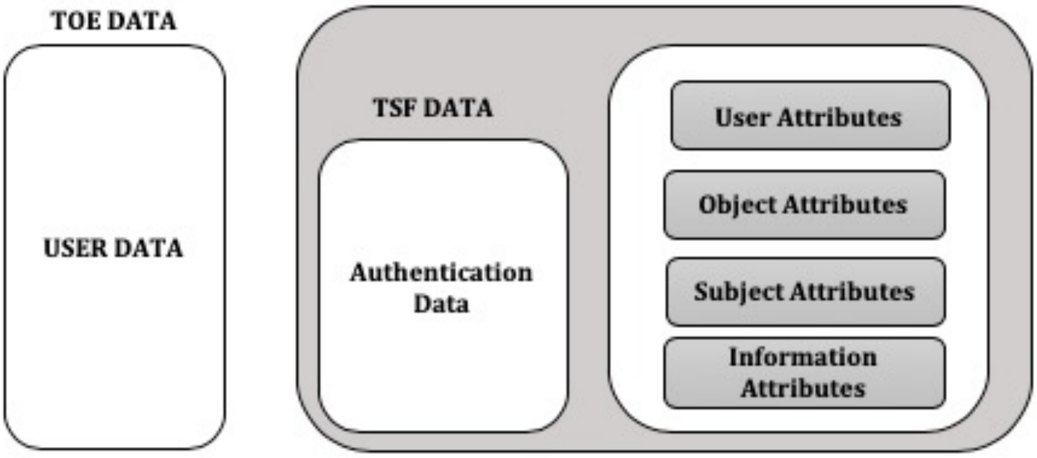

Figure 1 — Relationship between user data and TSF data

Two specific types of TSF data addressed by this document can be, but are not necessarily, the same. These are **authentication data** and **secrets**.

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 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.

EXAMPLE 9

Examples of biometric authentication devices include fingerprint readers and retinal scanners.

The term secrets, as used in this document, 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.

Therefore, some, but not all, authentication data needs to be kept secret and some, but not all, secrets are used as authentication data. Figure 2 shows this relationship between secrets and authentication data. In the Figure, the types of data typically encountered in the authentication data and the secrets subclauses are indicated.

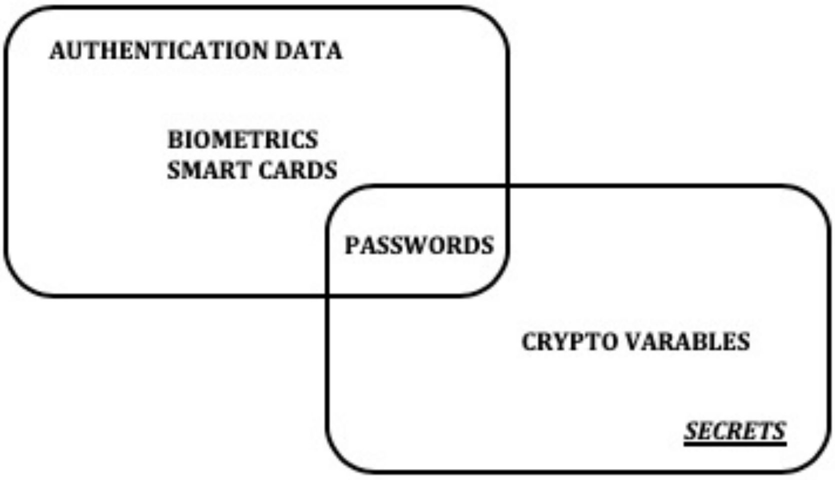

Figure 2 — Relationship between “authentication data” and “secrets”

6 Security functional components

6.1 Overview

6.1.1 General

Clause 6 defines the content and presentation of the functional requirements of this document and provides guidance on the organization of the requirements for new, extended components that may be included in a PP, PP-Module, functional package or ST. As described in ISO/IEC 15408-1 Clause 8, the functional components and requirements are expressed in classes, families, components and elements.

6.1.2 Class structure

Figure 3 illustrates the functional class structure in diagrammatic form. Each functional class includes a class name, class introduction, and one or more functional families.

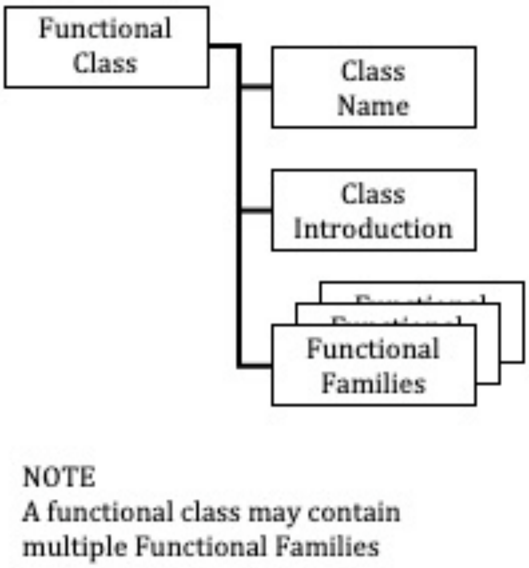

Figure 3 — Functional class structure

Class name

The class name subclause provides information necessary to identify and categorize 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.

Class introduction

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.

The class introduction provides a figure describing the families in this class and the hierarchy of the components in each family, as explained in 6.2.

6.1.3 Family structure

Figure 4 illustrates the functional family structure in diagrammatic form.

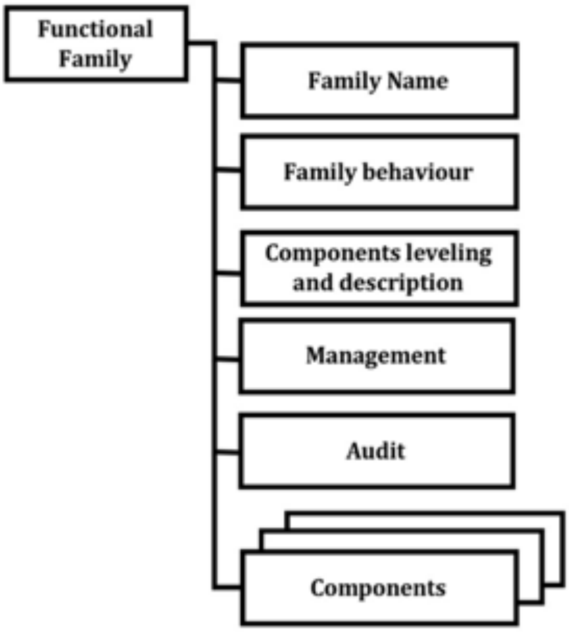

Figure 4 — Functional family structure

Family name

The family name subclause provides categorical and descriptive information necessary to identify and categorize 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 security components.

Family behaviour

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 SFRs derived from a component of this family;

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

Components leveling and description

Functional families contain one or more components, any one of which may be selected for inclusion in STs, PPs, PP-Modules or security functional packages. The goal of the Components leveling and description subclause 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.

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

The relationships between components within a functional family may be hierarchical. A component is hierarchical to another if it offers more security.

As explained in 6.2 the descriptions of the families provide a graphical overview of the hierarchy of the components in a family.

Management

The management clauses contain information for ST, PP, PP-Module, or security functional package authors to consider as management activities for a given component. The clauses reference components of the management class (FMT) and provide guidance regarding potential management activities that may be applied via operations to those components.

An author may select the indicated management components or may include other management requirements not listed to detail management activities. As such the information should be considered informative.

Audit

The audit requirements contain auditable events for the authors to select, if requirements from the class FAU, are included in the ST, PP, PP-Module, or security functional package. 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.

EXAMPLE 1

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.

It can be observed that the categorization of auditable events is hierarchical.

EXAMPLE 2

For example, when Basic Audit Generation is desired, all auditable events identified as being both Minimal and Basic are included in the PP, PP-Module, functional package or 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) are included in the PP, PP-Module, functional package or ST.

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

6.1.4 Component structure

Figure 5 illustrates the functional component structure.

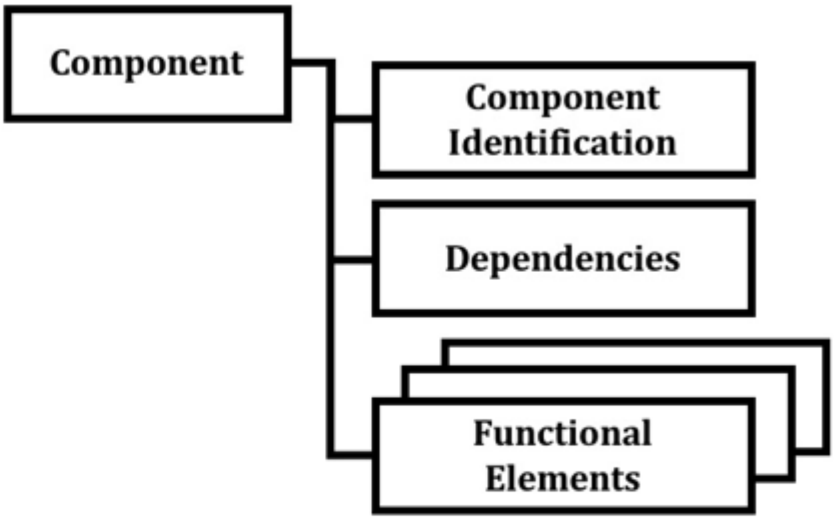

Figure 5 — Functional component structure

Component identification

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

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

A *unique short name*. A unique short form of the functional component name. This short name serves as the principal reference name for the categorization, 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.

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.

Functional elements

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

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, PP-Module, security functional package or an ST.

A unique short form of the functional element name is provided.

EXAMPLE

The component 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.

Dependencies

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.

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 other functional components that are required for this component 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 document. It is noted that in some cases the dependency is optional in that a number of functional components are provided, where each one of them would be sufficient to satisfy the dependency.

EXAMPLE

FDP_UIT.1 Data exchange integrity

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.

The dependencies indicated in this document are normative and they shall be satisfied within a package, PP or ST. In situations where the indicated dependencies are not applicable, the author shall satisfy the dependency by providing a rationale why it is not applicable and may leave the depended upon component from the package, PP or ST.

6.2 Component catalogue

6.2.1 General

The grouping of the components in this document does not reflect any formal taxonomy.

This document 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. Figure 6 is a useful indicator of the hierarchical relationship that may exist between components.

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

In each class, a figure describing the family hierarchy similar to Figure 6 is provided. In Figure 6 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.

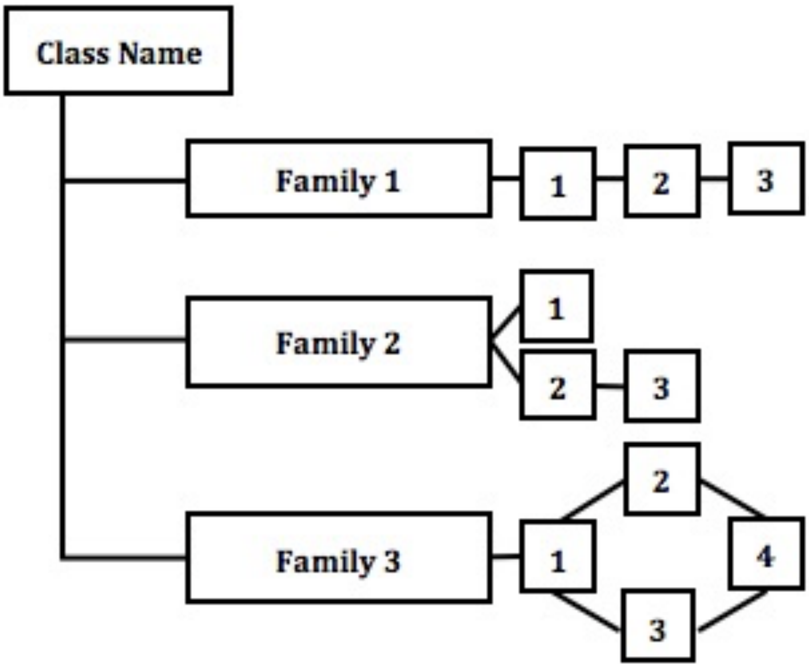

Figure 6 — Sample class decomposition diagram

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.

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.

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.

6.2.2 Component changes highlighting

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.

7 Class FAU: Security audit

7.1 Class description

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

Figure 7 shows the decomposition of this class, it’s families and components. Elements are not shown in the figure.

Annex C provides explanatory information for this class and should be consulted when using the components identified in this class.

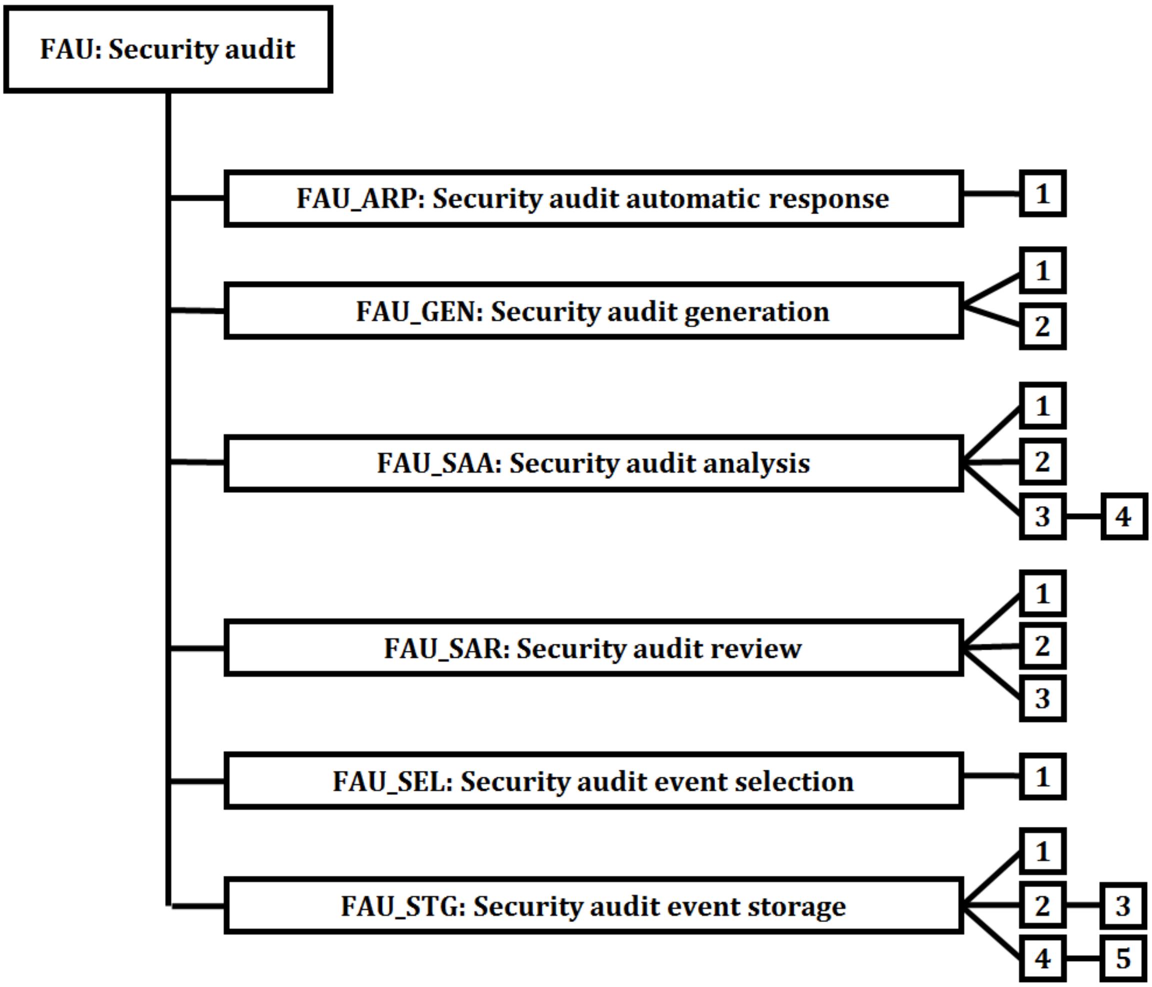
FAU: Security audit
FAU_ARP: Security audit automatic response
1
FAU_GEN: Security audit generation
1
2
FAU_SAA: Security audit analysis
1
2
3
4
FAU_SAR: Security audit review
1
2
3
FAU_SEL: Security audit event selection
1
FAU_STG: Security audit event storage
1
2
3
4
5

Figure 7 — FAU: Security audit class decomposition

7.2 Security audit automatic response (FAU_ARP)

7.2.1 Family behaviour

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

7.2.2 Components leveling and description

Figure 8 shows the component leveling for this family.

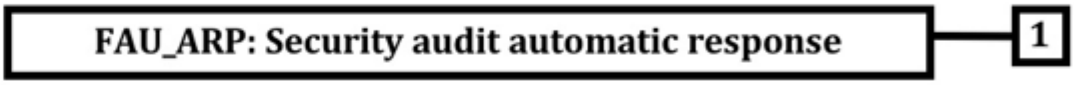

Figure 8 — FAU_ARP: Component leveling

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

7.2.3 Management of FAU_ARP.1

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

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

7.2.4 Audit of FAU_ARP.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

7.2.5 FAU_ARP.1 Security alarms

Component relationships

Hierarchical to:	No other components.
Dependencies:	FAU_SAA.1 Potential violation analysis

FAU_ARP.1.1

The TSF shall take [assignment: *list of actions*] upon detection of a potential security violation.

7.3 Security audit data generation (FAU_GEN)

7.3.1 Family behaviour

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.

7.3.2 Components leveling and description

Figure 9 shows the component leveling for this family.

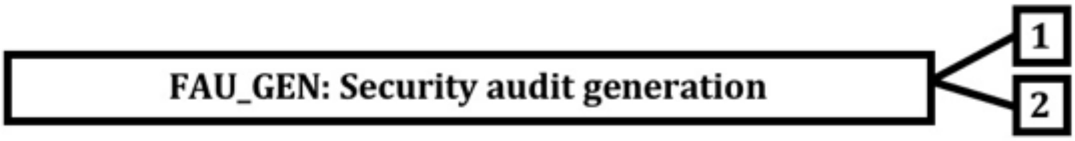

Figure 9 — FAU_GEN: Component leveling

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.

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

7.3.3 Management of FAU_GEN.1, FAU_GEN.2

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

a) There are no management activities foreseen.

7.3.4 Audit of FAU_GEN.1, FAU_GEN.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

7.3.5 FAU_GEN.1 Audit data generation

Component relationships

Hierarchical to:	No other components.
Dependencies:	FPT_STM.1 Reliable time stamps

FAU_GEN.1.1

The TSF shall be able to generate audit data 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 the audit data at least the following information:

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

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

7.3.6 FAU_GEN.2 User identity association

Component relationships

Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation FIA_UID.1 Timing of identification

FAU_GEN.2.1

For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

7.4 Security audit analysis (FAU_SAA)

7.4.1 Family behaviour

This family defines requirements for automated means that analyze 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 a potential security violation.

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

7.4.2 Components leveling and description

Figure 10 shows the component leveling for this family.

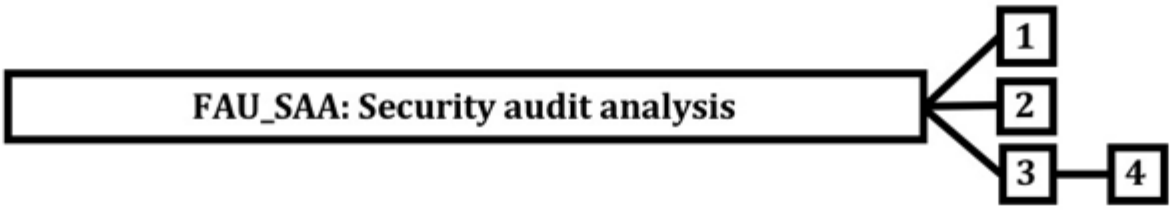

Figure 10 — FAU_SAA: Component leveling

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

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 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.

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 enforcement of the SFRs. This search for signature events may occur in real-time or during a post-collection batch-mode analysis.

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 enforcement of the SFRs.

7.4.3 Management of FAU_SAA.1

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.

7.4.4 Management of FAU_SAA.2

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.

7.4.5 Management of FAU_SAA.3

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

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

7.4.6 Management of FAU_SAA.4

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 sequences of system events.

7.4.7 Audit of FAU_SAA.1, FAU_SAA.2, FAU_SAA.3, FAU_SAA.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

b) Minimal: Automated responses performed by the tool.

7.4.8 FAU_SAA.1 Potential violation analysis

Component relationships

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 enforcement of the SFRs.

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*].

7.4.9 FAU_SAA.2 Profile based anomaly detection

Component relationships

Hierarchical to:	No other components.
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 a possible violation of the enforcement of the SFRs when a user's suspicion rating exceeds the following threshold conditions [assignment: *conditions under which anomalous activity is reported by the TSF*].

7.4.10 FAU_SAA.3 Simple attack heuristics

Component relationships

Hierarchical to:	No other components.
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 enforcement of the SFRs.

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 a potential violation of the enforcement of the SFRs when a system event is found to match a signature event that indicates a potential violation of the enforcement of the SFRs.

7.4.11 FAU_SAA.4 Complex attack heuristics

Component relationships

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 enforcement of the SFRs.

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 a potential violation of the enforcement of the SFRs when system **activity** is found to match a signature event **or event sequence** that indicates a potential violation of the enforcement of the SFRs.

7.5 Security audit review (FAU_SAR)

7.5.1 Family behaviour

This family defines the requirements for tools that are made available to authorized users to assist in the review of audit data.

7.5.2 Components leveling and description

Figure 11 shows the component leveling for this family.

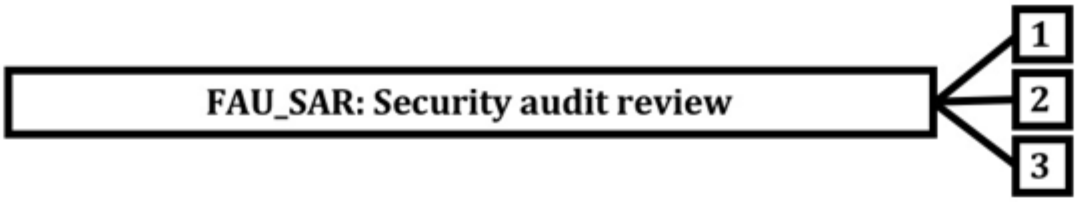

Figure 11 — FAU_SAR: Component leveling

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

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.

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

7.5.3 Management of FAU_SAR.1

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.

7.5.4 Management of FAU_SAR.2, FAU_SAR.3

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

a) There are no management activities foreseen.

7.5.5 Audit of FAU_SAR.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

7.5.6 Audit of FAU_SAR.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

7.5.7 Audit of FAU_SAR.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Detailed: the parameters used for the viewing.

7.5.8 FAU_SAR.1 Audit review

Component relationships

Hierarchical to:	No other components.
Dependencies:	FAU_GEN.1 Audit data generation

FAU_SAR.1.1

The TSF shall provide [assignment: *authorized users*] with the capability to read [assignment: *list of audit information*] from the audit data.

FAU_SAR.1.2

The TSF shall provide the audit data in a manner suitable for the user to interpret the information.

7.5.9 FAU_SAR.2 Restricted audit review

Component relationships

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 data, except those users that have been granted explicit read-access.

7.5.10 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 apply [assignment: *methods of selection and/or ordering*] of audit data based on [assignment: *criteria with logical relations*].

7.6 Security audit event selection (FAU_SEL)

7.6.1 Family behaviour

This family defines requirements to select the set of events to be audited during TOE operation from the set of all auditable events.

7.6.2 Components leveling and description

Figure 12 shows the component leveling for this family.

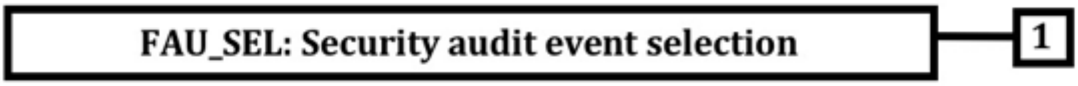

Figure 12 — FAU_SEL: Component leveling

FAU_SEL.1 Selective audit, requires the ability to select the set of events to be audited from the set of all auditable events, identified in FAU_GEN.1 Audit data generation, based upon attributes to be specified by the PP, PP-Module, functional package or ST author.

7.6.3 Management of FAU_SEL.1

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

a) Maintenance of the rights to view/modify the audit data.

7.6.4 Audit of FAU_SEL.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

7.6.5 FAU_SEL.1 Selective audit

Component relationships

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 select the set of events to be audited from the set of all auditable 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*]

7.7 Security audit data storage (FAU_STG)

7.7.1 Family behaviour

This family defines the requirements for the TSF to be able to create and maintain a secure audit trail. Stored audit data refers to those data stored within an audit trail, and not to any audit data that has been retrieved (to temporary storage) through selection.

7.7.2 Components leveling and description

Figure 13 shows the component leveling for this family.

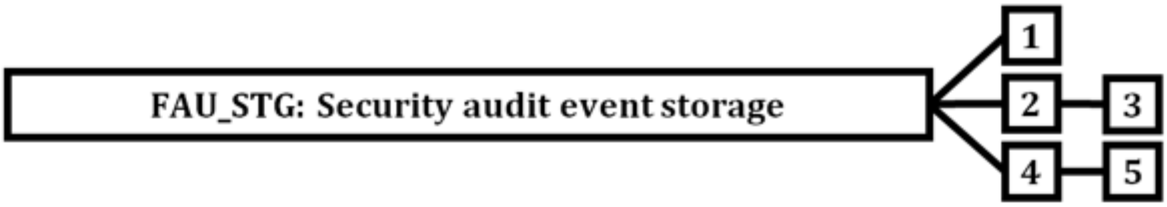

Figure 13 — FAU_STG: Component leveling

FAU_STG.1 Audit data storage location, requires that the storage location(s) for audit data be specified

FAU_STG.2 Protected audit data storage, requires that protections are placed on the audit data. It will be protected from unauthorized deletion and/or modification.

FAU_STG.3 Guarantees of audit data availability, specifies the guarantees that the TSF maintains over the audit data given the occurrence of an undesired condition.

FAU_STG.4 Action in case of possible audit data loss specifies actions to be taken if a threshold on the stored audit data is exceeded.

FAU_STG.5 Prevention of audit data loss specifies actions to be taken in the case that audit data storage is full.

7.7.3 Management of FAU_STG.1

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

a) Maintenance of remote audit storage locations

7.7.4 Management of FAU_STG.2

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

a) There are no management activities foreseen.

7.7.5 Management of FAU_STG.3

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

a) Maintenance of the parameters that control the audit data storage capability.

7.7.6 Management of FAU_STG.4

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 imminent audit data storage failure.

7.7.7 Management of FAU_STG.5

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 data storage failure.

7.7.8 Audit of FAU_STG.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Basic: Changes in the location of remote audit data storage.

7.7.9 Audit of FAU_STG.2, FAU_STG.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

7.7.10 Audit of FAU_STG.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

7.7.11 Audit of FAU_STG.5

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

7.7.12 FAU_STG.1 Audit data storage location

Component relationships

Hierarchical to:	No other components
Dependencies:	FAU_GEN.1 Audit data generation
	FTP_ITC.1 Inter-TSF trusted channel

FAU_STG.1.1

The TSF shall be able to store generated audit data on the [selection: *TOE itself, transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC, [assignment: other storage location(s)]*.]

7.7.13 FAU_STG.2 Protected audit data storage

Component relationships

Hierarchical to:	No other components
Dependencies:	FAU_GEN.1 Audit data generation

FAU_STG.2.1

The TSF shall protect the stored audit data in the audit trail from unauthorized deletion.

FAU_STG.2.2

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

7.7.14 FAU_STG.3 Guarantees of audit data availability

Component relationships

Hierarchical to:	FAU_STG.2 Protected audit data storage
Dependencies:	FAU_GEN.1 Audit data generation

FAU_STG.3.1

The TSF shall protect the stored audit data in the audit trail from unauthorized deletion.

FAU_STG.3.2

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

FAU_STG.3.3

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

7.7.15 FAU_STG.4 Action in case of possible audit data loss

Component relationships

Hierarchical to:	No other components
Dependencies:	FAU_STG.2 Protected audit data storage

FAU_STG.4.1

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

7.7.16 FAU_STG.5 Prevention of audit data loss

Component relationships

Hierarchical to:	FAU_STG.4 Action in case of possible audit data loss
Dependencies:	FAU_STG.2 Protected audit data storage
	FAU_GEN.1 Audit data generation

FAU_STG.5.1

The TSF shall [selection: *ignore audited events, "prevent audited events, except those taken by the authorized user with special rights", overwrite the oldest stored audit records*], [assignment: *other actions to be taken in case of audit storage failure and conditions for the actions*] if the audit data storage is full.

8 Class FCO: Communication

8.1 Class description

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 14 shows the decomposition of the class.

Figure 14 shows the decomposition of this class, it’s families and components. Elements are not shown in the figure.

Annex D provides explanatory information for this class and should be consulted when using the components identified in this class.

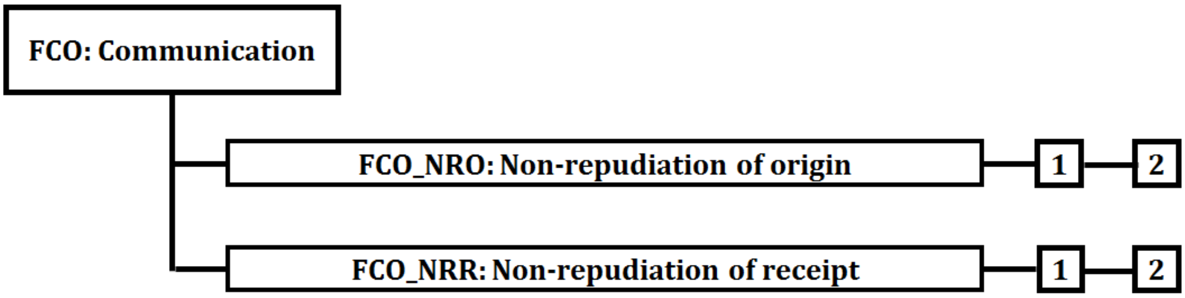
FCO: Communication
FCO_NRO: Non-repudiation of origin
1
2
FCO_NRR: Non-repudiation of receipt
1
2

Figure 14 — FCO: Communication class decomposition

8.2 Non-repudiation of origin (FCO_NRO)

8.2.1 Family behaviour

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.

8.2.2 Components leveling and description

Figure 15 shows the component leveling for this family.

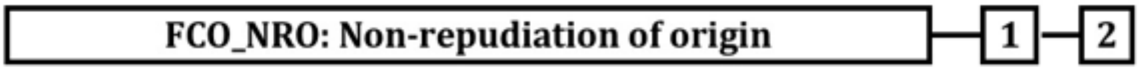

Figure 15 — FCO_NRO: Component leveling

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

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

8.2.3 Management of FCO_NRO.1, FCO_NRO.2

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.

8.2.4 Audit of FCO_NRO.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

8.2.5 Audit of FCO_NRO.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

8.2.6 FCO_NRO.1 Selective proof of origin

Component relationships

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*].

8.2.7 FCO_NRO.2 Enforced proof of origin

Component relationships

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*].

8.3 Non-repudiation of receipt (FCO_NRR)

8.3.1 Family behaviour

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.

8.3.2 Components leveling and description

Figure 16 shows the component leveling for this family.

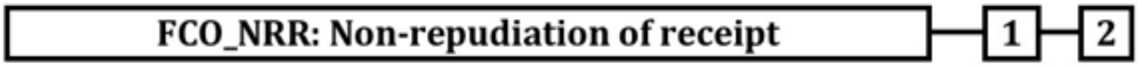

Figure 16 — FCO_NRR: Component leveling

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

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

8.3.3 Management of FCO_NRR.1, FCO_NRR.2

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-party recipients of evidence.

8.3.4 Audit of FCO_NRR.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

8.3.5 Audit of FCO_NRR.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

8.3.6 FCO_NRR.1 Selective proof of receipt

Component relationships

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*].

8.3.7 FCO_NRR.2 Enforced proof of receipt

Component relationships

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*] at **all times.**

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*].

9 Class FCS: Cryptographic support

9.1 Class description

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.

The FCS: Cryptographic support class is composed of four families.

Figure 17 shows the decomposition of this class, it's families and components. Elements are not shown in the figure.

Annex E provides explanatory information for this class and should be consulted when using the components identified in this class.

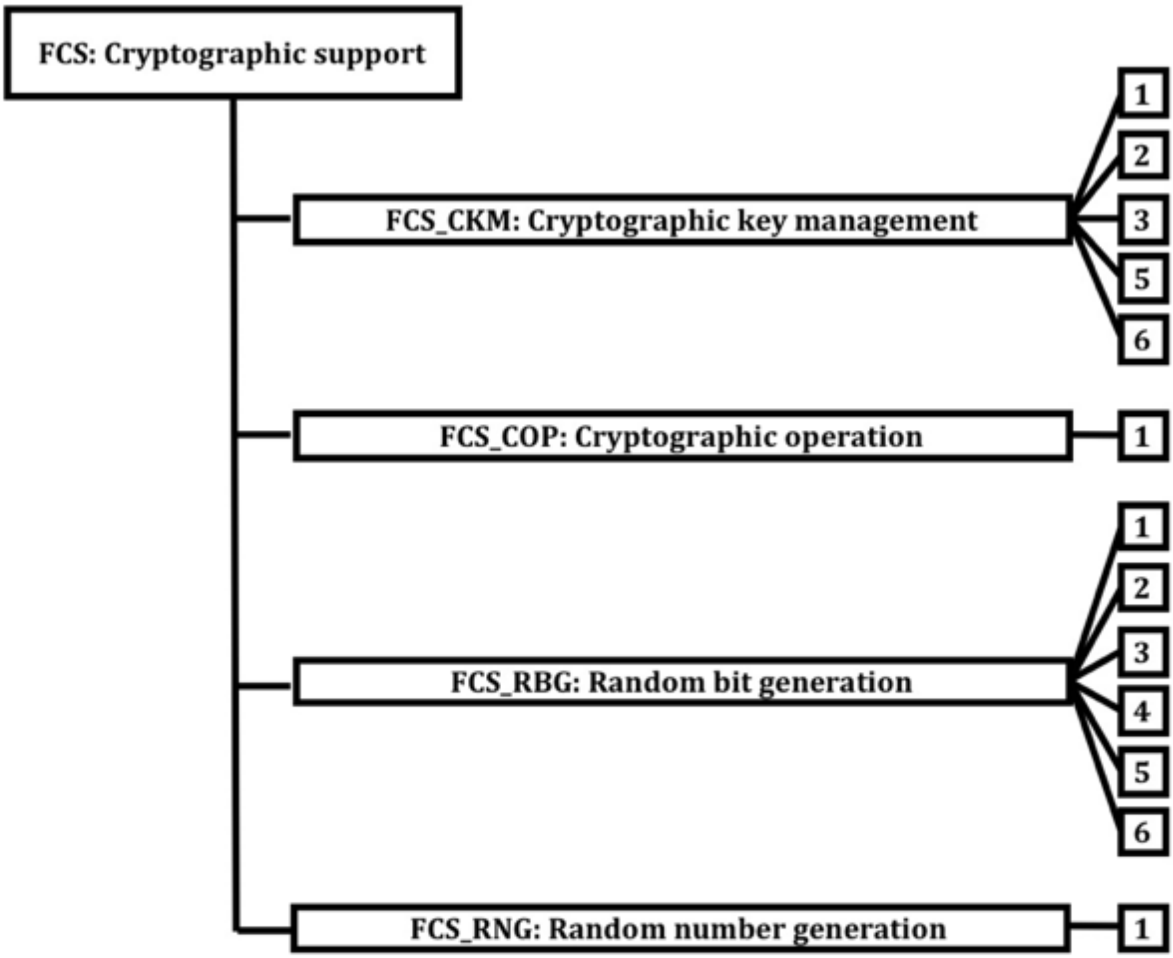
FCS: Cryptographic support
FCS_CKM: Cryptographic key management
1
2
3
5
6
FCS_COP: Cryptographic operation
1
FCS_RBG: Random bit generation
1
2
3
4
5
6
FCS_RNG: Random number generation
1

Figure 17 — FCS: Cryptographic support class decomposition

9.2 Cryptographic key management (FCS_CKM)

9.2.1 Family behaviour

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 derivation, cryptographic key distribution, cryptographic key access and timing and event of cryptographic key destruction. This family should be included whenever there are functional requirements for the management of cryptographic keys.

9.2.2 Components leveling and description

Figure 18 shows the component leveling for this family.

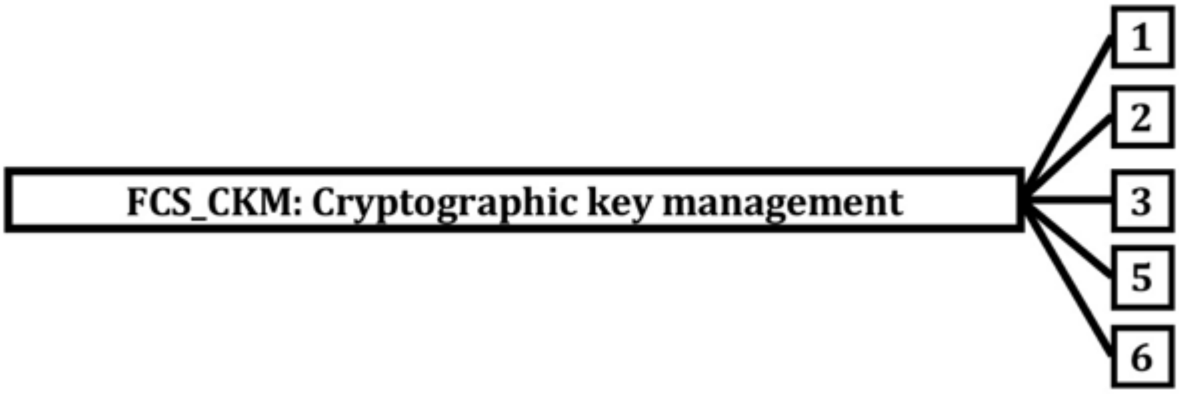

Figure 18 — FCS_CKM: Component leveling

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.

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.

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.

FCS_CKM.5 Cryptographic key derivation, requires that the methods, standards, and parameters for key-derivation are specified.

FCS_CKM.6 Timing and event of cryptographic key destruction, requires cryptographic keys to be destroyed in accordance with specified destruction methods which can be based on an assigned standard.

NOTE Previous editions of this standard specified FCS_CKM.4 which has been deprecated in this edition of ISO/IEC 15408-2. In order to preserve consistency when applying different editions of ISO/IEC 15408-2 the component number has not been re-used.

9.2.3 Management of FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.5, CKM.6

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

a) There are no management activities foreseen.

9.2.4 Audit of FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.5, CKM.6

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Success and failure of the activity.

b) Basic: The object attribute(s), and object value(s) excluding any sensitive information

9.2.5 FCS_CKM.1 Cryptographic key generation

Component relationships

Hierarchical to:	No other components.

Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_CKM.5 Cryptographic key derivation, or FCS_COP.1 Cryptographic operation]
	FCS_CKM.3 Cryptographic key access
	[FCS_RBG.1 Random bit generation, or FCS_RNG.1 Generation of random numbers]
	FCS_CKM.6 Timing and event of cryptographic key destruction

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*].

9.2.6 FCS_CKM.2 Cryptographic key distribution

Component relationships

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 or FCS_CKM.5 Cryptographic key derivation]
	FCS_CKM.3 Cryptographic key access

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*].

9.2.7 FCS_CKM.3 Cryptographic key access

Component relationships

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 or FCS_CKM.5 Cryptographic key derivation]

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*].

9.2.8 FCS_CKM.4 Cryptographic key destruction

The component has been deprecated. See FCS_CKM.6 Timing and event of cryptographic key destruction instead.

9.2.9 FCS_CKM.5 Cryptographic key derivation

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.6 Timing and event of cryptographic key destruction

FCS_CKM.5.1

The TSF shall derive cryptographic keys [assignment: *key type*] from [selection: *input parameters*] in accordance with a specified key derivation algorithm [selection: *key derivation algorithm*] and specified cryptographic key sizes [selection: *list of key sizes*] that meet the following: [assignment*: list of standards*].

NOTE See E.2.6. for information on using this component.

9.2.10 FCS_CKM.6 Timing and event of cryptographic key destruction

Component relationships

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.6.1

The TSF shall destroy [assignment: *list of cryptographic keys (including keying material)*] when [selection: *no longer needed, [assignment: other circumstances for key or key material destruction]*].

FCS_CKM.6.2

The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].

9.3 Cryptographic operation (FCS_COP)

9.3.1 Family behaviour

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.

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.

9.3.2 Components leveling and description

Figure 19 shows the component leveling for this family.

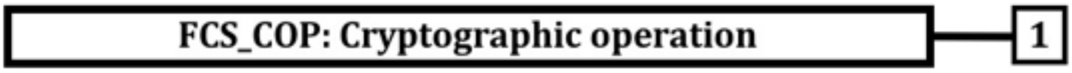

Figure 19 — FCS_COP: Component leveling

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.

9.3.3 Management of FCS_COP.1

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

a) There are no management activities foreseen.

9.3.4 Audit of FCS_COP.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

9.3.5 FCS_COP.1 Cryptographic operation

Component relationships

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, or FCS_CKM.5 Cryptographic key derivation] FCS_CKM.3 Cryptographic key access

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*].

9.4 Random bit generation (FCS_RBG)

9.4.1 Family behaviour

Components in this family address the requirements for random bit/number generation.

9.4.2 Components leveling and description

Figure 20 shows the component leveling for this family.

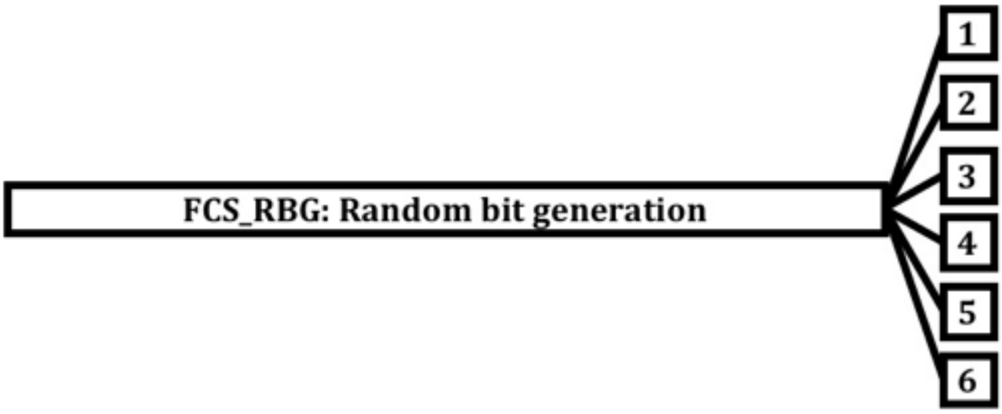

Figure 20 — FCS_RBG: Component leveling

FCS_RBG.1 Random bit generation (RBG) requires random bit generation to be performed in accordance with selected standards.

FCS_RBG.2 Random bit generation (external seeding) gives requirements for seeding by an external (outside the TOE) entropy source.

FCS_RBG.3 Random bit generation (internal seeding – single source) gives requirements for seeding using a TSF entropy source.

FCS_RBG.4 Random bit generation (internal seeding – multiple sources) gives requirements for seeding using multiple TSF entropy sources.

FCS_RBG.5 Random bit generation (combining entropy sources) gives requirements for combining multiple entropy sources (multiple internal sources, internal and external).

FCS_RBG.6 Random bit generation service requires random numbers to be supplied over an external interface as a service to other entities.

9.4.3 Management of FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5, FCS_RBG.6

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

a) There are no management activities foreseen.

9.4.4 Audit of FCS_RBG.1, FCS_RBG.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: failure of the randomization process, failure to initialize or reseed (as supported by the technology)

9.4.5 Audit of FCS_RBG.3, FCS_RBG.4, FCS_RBG.6, FCS_RBG.6

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

9.4.6 FCS_RBG.1 Random bit generation (RBG)

Component relationships

Hierarchical to:	No other components
Dependencies:	[FCS_RBG.2 Random bit generation (external seeding), or FCS_RBG.3 Random bit generation (internal seeding – single source)] FPT_FLS.1 Failure with preservation of secure state FPT_TST.1 TSF self-testing

FCS_RBG.1.1

The TSF shall perform deterministic random bit generation services using [assignment: *RBG algorithm*] in accordance with [assignment: *list of standards*] after initialization with a seed.

FCS_RBG.1.2

The TSF shall initialize and update the RBG state using a noise source under a specified condition as shown in the RBG State Update Table.

RBG State Update Table

Identifier	Noise source	Update type	Condition	list of standards
Source1	*[selection: TOE internal, external]*	initialize	initialization	*[assignment: list of standards]*
[assignment: identifier]	*[selection: TOE internal, external]*	*[selection: reseed, uninstantiate+instantiate]*	*[selection: on demand; on the condition: [assignment: condition]; after [assignment: time]]*	*[assignment: list of standards]*

9.4.7 FCS_RBG.2 Random bit generation (external seeding)

Component relationships

Hierarchical to:	No other components.
Dependencies:	FCS_RBG.1 Random bit generation (RBG)

FCS_RBG.2.1

The TSF shall be able to accept a minimum input of [assignment: *minimum input length greater than zero*] from an external interface for the purpose of seed generation.

9.4.8 FCS_RBG.3 Random bit generation (internal seeding – single source)

Component relationships

Hierarchical to:	No other components
Dependencies:	FCS_RBG.1 Random bit generation (RBG)

FCS_RBG.3.1

The TSF shall be able to seed the RBG using a single [selection: *TSF software-based noise source, TSF hardware-based noise source*] with a minimum of [assignment: *number of bits*] bits of min-entropy.

9.4.9 FCS_RBG.4 Random bit generation (internal seeding – multiple sources)

Component relationships

Hierarchical to:	No other components
Dependencies:	FCS_RBG.1 Random bit generation (RBG) FCS_RBG.3 Random bit generation (internal seeding – single source)

FCS_RBG.4.1

The TSF shall be able to seed the RBG using **[selection: *[assignment: number] TSF software-based noise source(s), [assignment: number] TSF hardware-based noise source(s)*].**

9.4.10 FCS_RBG.5 Random bit generation (combining entropy sources)

Component relationships

Hierarchical to:	No other components.
Dependencies:	FCS_RBG.1 Random bit generation (RBG) [FCS_RBG.2 Random bit generation (external seeding), or FCS_RBG.3 Random bit generation (internal seeding – single source)]

FCS_RBG.5.1 Combining entropy sources

The TSF shall [assignment: *combining operation*] [selection: *TSF entropy source(s), TOE external entropy source(s)*] to create the entropy input into the derivation function as defined in [assignment: *list of standards*], resulting in a minimum of [assignment: *number of bits*] bits of min-entropy.

9.4.11 FCS_RBG.6 Random bit generation service

Component relationships

Hierarchical to:	No other components.
Dependencies:	FCS_RBG.1 Random bit generation (RBG) [FCS_RBG.2 Random bit generation (external seeding), or FCS_RBG.3 Random bit generation (internal seeding – single source)]

FCS_RBG.6.1

The TSF shall provide a [selection: *hardware, software, [assignment: other interface type]*] interface to make the RBG output, as specified in FCS_RBG.1 Random bit generation (RBG), available as a service to entities outside of the TOE.

9.5 Generation of random numbers (FCS_RNG)

9.5.1 Family behaviour

This family defines quality requirements for the generation of random numbers which are intended to be use for cryptographic purposes.

9.5.2 Components leveling and description

Figure 21 shows the component leveling for this family.

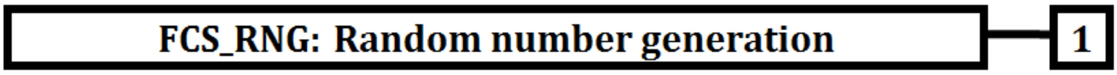

Figure 21 — FCS_RNG: Component leveling

FCS_RNG.1 Random number generation requires that random numbers meet a defined quality metric.

9.5.3 Management of FCS_RNG.1

There are no management activities foreseen.

9.5.4 Audit of FCS_RNG.1

There are no actions defined to be auditable.

9.5.5 FCS_RNG.1 Random number generation

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FCS_RNG.1.1

The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*] random number generator that implements: [assignment: *list of security capabilities*].

FCS_RNG.1.2

The TSF shall provide [selection: *bits, octets of bits, numbers* [assignment: *format of the numbers*]] that meet [assignment: *a defined quality metric*].

10 Class FDP: User data protection

10.1 Class description

This class contains families specifying requirements 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.

The families in this class are organized 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, PP-Module, functional package or 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);
   - Information Retention Control (FDP_IRC)
   - Residual information protection (FDP_RIP);
   - Rollback (FDP_ROL);
   - Stored data confidentiality (FDP_SDC); and
   - Stored data integrity (FDP_SDI).

c) Off-line storage, import and export:
   - Data authentication (FDP_DAU);
   - Export from the TOE (FDP_ETC);
   - Import from outside of the TOE (FDP_ITC).

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

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 22 shows the decomposition of this class, it’s families and components. Elements are not shown in the figure.

Annex F provides explanatory information for this class and should be consulted when using the components identified in this class.

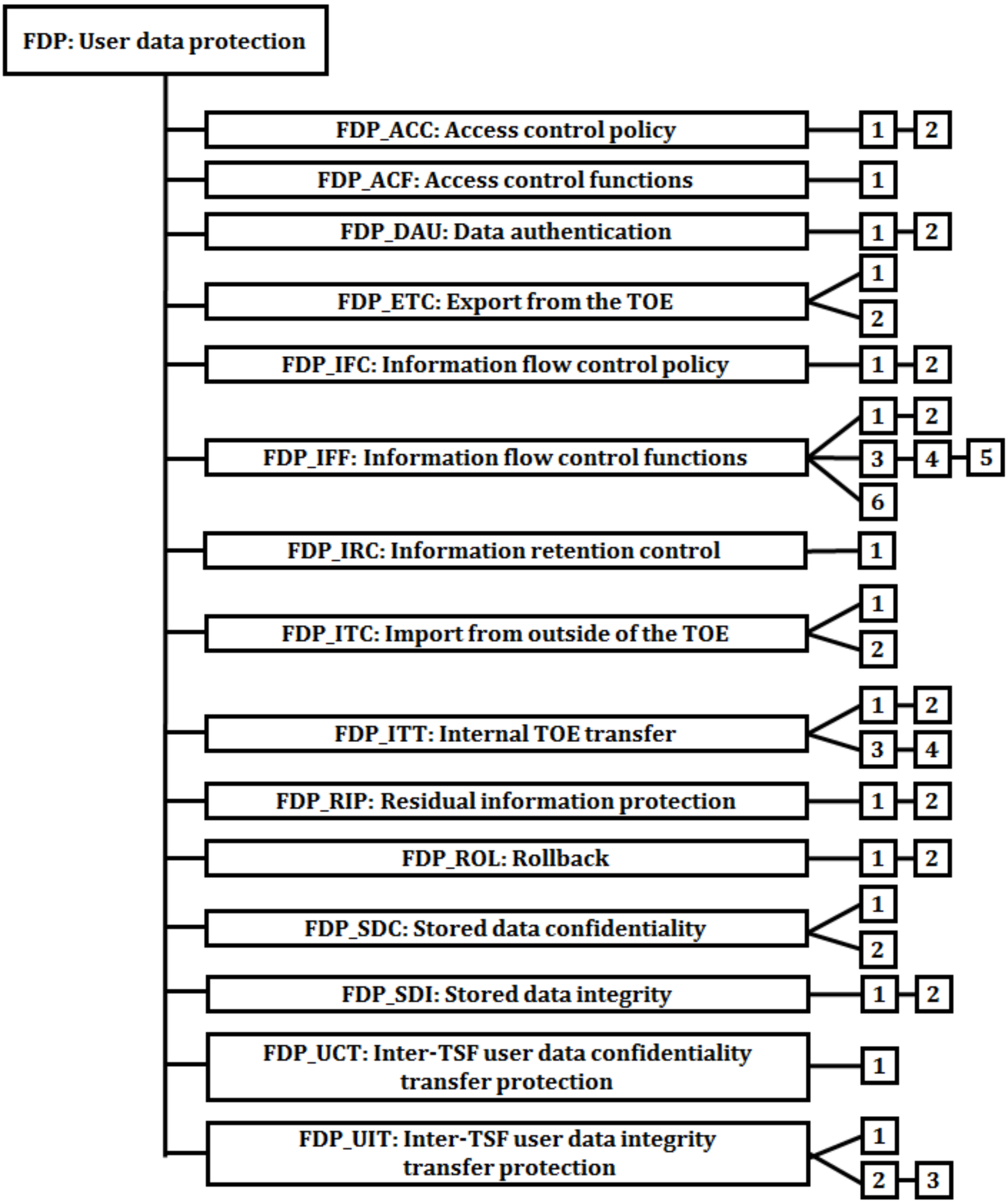
FDP: User data protection
FDP_ACC: Access control policy
1
2
FDP_ACF: Access control functions
1
FDP_DAU: Data authentication
1
2
FDP_ETC: Export from the TOE
1
2
FDP_IFC: Information flow control policy
1
2
FDP_IFF: Information flow control functions
1
2
3
4
5
6
FDP_IRC: Information retention control
1
FDP_ITC: Import from outside of the TOE
1
2
FDP_ITT: Internal TOE transfer
1
2
3
4
FDP_RIP: Residual information protection
1
2
FDP_ROL: Rollback
1
2
FDP_SDC: Stored data confidentiality
1
2
FDP_SDI: Stored data integrity
1
2
FDP_UCT: Inter-TSF user data confidentiality transfer protection
1
FDP_UIT: Inter-TSF user data integrity transfer protection
1
2
3

Figure 22 — FDP: User data protection class decomposition

10.2 Access control policy (FDP_ACC)

10.2.1 Family behaviour

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 SFRs related to the SFP. This scope of control is characterized 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 allow 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 Export from the TOE (FDP_ETC). 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.”

10.2.2 Components leveling and description

Figure 23 shows the component leveling for this family.

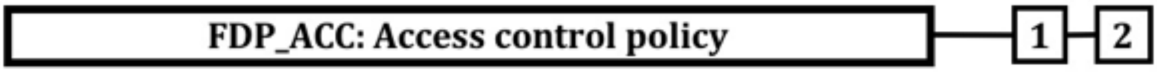

Figure 23 — FDP_ACC: Component leveling

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.

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 protected by the TSF are covered by at least one identified access control SFP.

10.2.3 Management of FDP_ACC.1, FDP_ACC.2

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

a) There are no management activities foreseen.

10.2.4 Audit of FDP_ACC.1, FDP_ACC.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

10.2.5 FDP_ACC.1 Subset access control

Component relationships

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*].

10.2.6 FDP_ACC.2 Complete access control

Component relationships

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 controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

10.3 Access control functions (FDP_ACF)

10.3.1 Family behaviour

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.

10.3.2 Components leveling and description

Figure 24 shows the component leveling for this family.

FDP_ACF: Access control functions
1

Figure 24 — FDP_ACF: Component leveling

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, PP-Module, functional package or ST author may also iterate this component to address multiple policies in the TOE.

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 authorize or deny access to an object based upon security attributes.

10.3.3 Management of FDP_ACF.1

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.

10.3.4 Audit of FDP_ACF.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

10.3.5 FDP_ACF.1 Security attribute-based access control

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control
	FMT_MSA.3 Static attribute

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 authorize access of subjects to objects based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorize access of subjects to objects*].

FDP_ACF.1.4

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

10.4 Data authentication (FDP_DAU)

10.4.1 Family behaviour

Data authentication permits an entity to accept responsibility for the authenticity of information. 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.

10.4.2 Components leveling and description

Figure 25 shows the component leveling for this family.

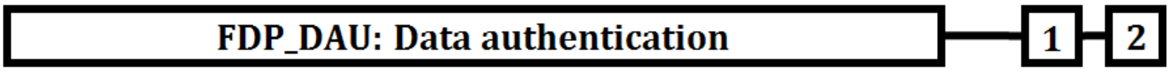

Figure 25 — FDP_DAU: Component leveling

FDP_DAU.1 Basic Data Authentication, requires that the TSF is capable of generating a guarantee of authenticity of the information content of objects.

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.

10.4.3 Management of FDP_DAU.1, FDP_DAU.2

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.

10.4.4 Audit of FDP_DAU.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

10.4.5 Audit of FDP_DAU.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

10.4.6 FDP_DAU.1 Basic Data Authentication

Component relationships

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.

10.4.7 FDP_DAU.2 Data Authentication with Identity of Guarantor

Component relationships

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.**

10.5 Export from the TOE (FDP_ETC)

10.5.1 Family behaviour

This family defines functions for TSF-mediated exporting of 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.

10.5.2 Components leveling and description

Figure 26 shows the component leveling for this family.

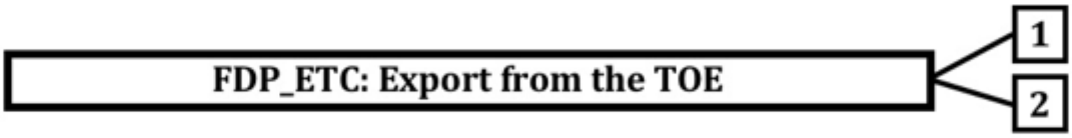

Figure 26 — FDP_ETC: Component leveling

FDP_ETC.1 Export of user data without security attributes, requires that the TSF enforces 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.

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

10.5.3 Management of FDP_ETC.1

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

a) There are no management activities foreseen.

10.5.4 Management of FDP_ETC.2

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.

10.5.5 Audit of FDP_ETC.1, FDP_ETC.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Successful export of information.

b) Basic: All attempts to export information.

10.5.6 FDP_ETC.1 Export of user data without security attributes

Component relationships

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 TOE.

FDP_ETC.1.2

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

10.5.7 FDP_ETC.2 Export of user data with security attributes

Component relationships

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 TOE.

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 TOE, are unambiguously associated with the exported user data.

FDP_ETC.2.4

The TSF shall ensure that interpretation of the security attributes of the exported user data is as intended by the owner of the user data.

FDP_ETC.2.5

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

10.6 Information flow control policy (FDP_IFC)

10.6.1 Family behaviour

This family identifies the information flow control SFPs (by name) and defines the scope of control for each named information flow control SFP. This scope of control is characterized 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 allow 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 Export from the TOE (FDP_ETC). 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.”

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.

10.6.2 Components leveling and description

Figure 27 shows the component leveling for this family.

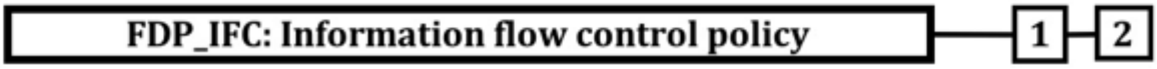

Figure 27 — FDP_IFC: Component leveling

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.

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 controlled by the TSF are covered by at least one identified information flow control SFP.

10.6.3 Management of FDP_IFC.1, FDP_IFC.2

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

a) There are no management activities foreseen.

10.6.4 Audit of FDP_IFC.1, FDP_IFC.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

10.6.5 FDP_IFC.1 Subset information flow control

Component relationships

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*].

10.6.6 FDP_IFC.2 Complete information flow control

Component relationships

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 TOE to flow to and from any subject in the TOE are covered by an information flow control SFP.

10.7 Information flow control functions (FDP_IFF)

10.7.1 Family behaviour

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.

10.7.2 Components leveling and description

Figure 28 shows the component leveling for this family.

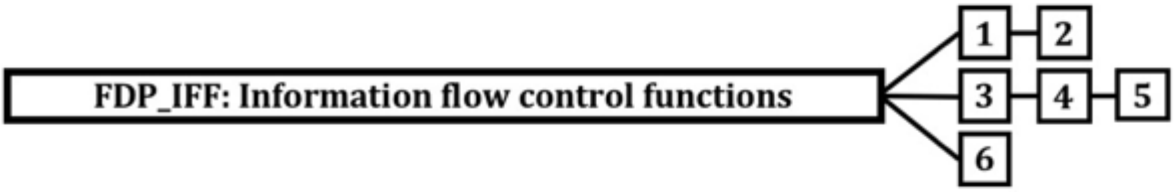

Figure 28 — FDP_IFF: Component leveling

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.

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 set of SFRs use hierarchical security attributes that form a lattice (as defined in mathematics). FDP_IFF.2.6 is derived from the mathematical properties of a lattice. A lattice consists of a set of elements with an ordering relationship with the property defined in the first bullet, a least upper bound which is the unique element in the set that is greater or equal (in the ordering relationship) than any other element of the lattice, and a greatest lower bound, which is the unique element in the set that is smaller or equal than any other element of the lattice.

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

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.

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

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

10.7.3 Management of FDP_IFF.1, FDP_IFF.2

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

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

10.7.4 Management of FDP_IFF.3, FDP_IFF.4, FDP_IFF.5

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

a) There are no management activities foreseen.

10.7.5 Management of FDP_IFF.6

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.

10.7.6 Audit of FDP_IFF.1, FDP_IFF.2, FDP_IFF.5

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

10.7.7 Audit of FDP_IFF.3, FDP_IFF.4, FDP_IFF.6

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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;
f) Detailed: The use of identified illicit information flow channels with estimated maximum capacity exceeding a specified value.

10.7.8 FDP_IFF.1 Simple security attributes

Component relationships

Hierarchical to:	No other components.
Dependencies:	FDP_IFC.1 Subset information flow control
	FMT_MSA.3 Static attribute

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 explicitly authorize an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorize information flows*].

FDP_IFF.1.5

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

10.7.9 FDP_IFF.2 Hierarchical security attributes

Component relationships

Hierarchical to:	FDP_IFF.1 Simple security attributes
Dependencies:	FDP_IFC.1 Subset information flow control
	FMT_MSA.3 Static attribute

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 explicitly authorize an information flow based on the following rules: [assignment: *rules, based on security attributes, that explicitly authorize information flows*].

FDP_IFF.2.5

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.2.6

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.

10.7.10 FDP_IFF.3 Limited illicit information flows

Component relationships

Hierarchical to: No other components.

Dependencies: 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*].

10.7.11 FDP_IFF.4 Partial elimination of illicit information flows

Component relationships

Hierarchical to: FDP_IFF.3 Limited illicit information flows

Dependencies: 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*].

10.7.12 FDP_IFF.5 No illicit information flows

Component relationships

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

Dependencies: 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*]**.

10.7.13 FDP_IFF.6 Illicit information flow monitoring

Component relationships

Hierarchical to: No other components.

Dependencies: 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*].

10.8 Information Retention Control (FDP_IRC)

10.8.1 Family behaviour

The “Information retention control” family addresses a basic need in secure information processing and storage applications for the secure management of data no longer needed by the TOE to perform its operations, but that is still stored in the TOE.

The historical view of IT systems as data storage systems suggested that once entered, data would seldom be deleted from the system, and if it was deleted, this would mainly be because of storage exhaustion problems.

However, in a multilateral or high security environment it is important to minimize the replication of data, as well as the time period during which data is stored in the system. It is also possible that users could want their IT products to avoid retaining sensitive data that they consider to be exploitable by third parties or that could threaten privacy. FDP_IRC may help users to gain confidence that the product is secure by deleting every copy of the data when it is no longer needed.

The FDP_RIP “Residual information protection” family addresses one side of this problem, but an explicit requirement on the management of data that is no longer needed is missing.

Of course, competing requirements may arise, since some data may be needed by the system for more operations over a longer time period. Possible solutions to this problem are:

- Better protecting the information objects stored in the TOE from access,
- Re-requesting the protected information from the user each time it is needed.

Information retention control ensures, that data no longer necessary for the operation of the TOE is deleted by the TOE. Components of this family require the PP, PP-Module, functional package or ST author to identify the TOE operations, including both simple and complex processing and the information objects, that are not to be kept in the TOE, that are the subject of those operations.

The TOE is also required to keep track of such stored information objects, and to delete both the on-line and the off-line information objects that are no longer required.

This family sets only requirements on information objects requested for specific activities in the TOE operation, and not on general data gathering. The policies which control the collection, storage, processing, disclosure, and elimination of general user data stored on the TOE must be detailed elsewhere, and are domain of the environmental objectives and organizational policies, not of the PP, PP-Module, functional package or ST.

When more than one operation requires the presence of a protected object, all operations, which refer to the required object shall end before deleting it.

10.8.2 Components leveling and description

Figure 29 shows the component leveling for this family.

FDP_IRC: Information retention control
1

Figure 29 — FDP_IRC: Component leveling

FDP_IRC.1 Information retention control requires that the TSF ensure that any copy of a defined set of objects in the TOE is deleted when no longer strictly necessary for the operation of the TOE, and to identify and define the operations for which the object is required.

10.8.3 Management of FDP_IRC.1

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

a) There are no management actions foreseen.

10.8.4 Audit of FDP_IRC.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

10.8.5 FDP_IRC.1 Information retention control

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FDP_IRC.1.1

The TSF shall enforce the [assignment: *information erasure policy*] on a [assignment: *list of objects*] required for [assignment: *list of operations*] so that the selected objects are deleted irreversibly and untraceably from the TOE promptly upon termination of the selected operations.

FDP_IRC.1.2

The TSF shall ensure that [assignment: *list of objects*] cannot be accessed after their release and prior to their irreversible and untraceable deletion.

10.9 Import from outside of the TOE (FDP_ITC)

10.9.1 Family behaviour

This family defines the mechanisms for TSF-mediated importing 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.

10.9.2 Components leveling and description

Figure 30 shows the component leveling for this family.

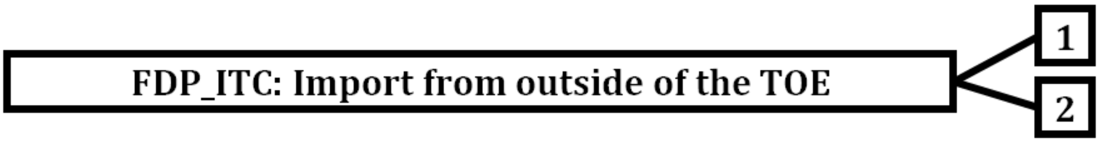

Figure 30 — FDP_ITC: Component leveling

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.

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 TOE.

10.9.3 Management of FDP_ITC.1, FDP_ITC.2

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

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

10.9.4 Audit of FDP_ITC.1, FDP_ITC.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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 authorized user.

10.9.5 FDP_ITC.1 Import of user data without security attributes

Component relationships

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 initialization

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 TOE.

FDP_ITC.1.2

The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3

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

10.9.6 FDP_ITC.2 Import of user data with security attributes

Component relationships

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(s) and/or information flow control SFP(s)*] when importing user data, controlled under the SFP, from outside of the TOE.

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 TOE: [assignment: *additional importation control rules*].

10.10 Internal TOE transfer (FDP_ITT)

10.10.1 Family behaviour

This family provides requirements that address protection of user data when it is transferred between separated 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 from the TOE (FDP_ETC) and Import from outside of the TOE (FDP_ITC), which address TSF-mediated transfer of data to or from outside the TOE.

10.10.2 Components leveling and description

Figure 31 shows the component leveling for this family.

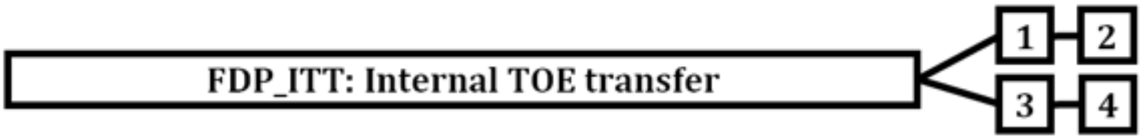

Figure 31 — FDP_ITT: Component leveling

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

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.

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

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.

10.10.3 Management of FDP_ITT.1, FDP_ITT.2

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.

10.10.4 Management of FDP_ITT.3, FDP_ITT.4

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.

10.10.5 Audit of FDP_ITT.1, FDP_ITT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

10.10.6 Audit of FDP_ITT.3, FDP_ITT.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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: Unauthorized attempts to change the integrity protection method.

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

10.10.7 FDP_ITT.1 Basic internal transfer protection

Component relationships

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.

10.10.8 FDP_ITT.2 Transmission separation by attribute

Component relationships

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*].

10.10.9 FDP_ITT.3 Integrity monitoring

Component relationships

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*].

10.10.10 FDP_ITT.4 Attribute-based integrity monitoring

Component relationships

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*].

10.11 Residual information protection (FDP_RIP)

10.11.1 Family behaviour

This family addresses the need to ensure that any data contained in a resource is not available when the resource is de-allocated from one object and reallocated to a different object. This family requires protection for any data contained in a resource that has been logically deleted or released but may still be present within the TSF-controlled resource which in turn may be re-allocated to another object.

10.11.2 Components leveling and description

Figure 32 shows the component leveling for this family.

FDP_RIP: Residual information protection
1
2

Figure 32 — FDP_RIP: Component leveling

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 controlled by the TSF upon the resource's allocation or deallocation.

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.

10.11.3 Management of FDP_RIP.1, FDP_RIP.2

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.

10.11.4 Audit of FDP_RIP.1, FDP_RIP.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

10.11.5 FDP_RIP.1 Subset residual information protection

Component relationships

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*].

10.11.6 FDP_RIP.2 Full residual information protection

Component relationships

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.

10.12 Rollback (FDP_ROL)

10.12.1 Family behaviour

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.

10.12.2 Components leveling and description

Figure 33 shows the component leveling for this family.

FDP_ROL: Rollback
1
2

Figure 33 — FDP_ROL: Component leveling

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

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

10.12.3 Management of FDP_ROL.1, FDP_ROL.2

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.

10.12.4 Audit of FDP_ROL.1, FDP_ROL.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

10.12.5 FDP_ROL.1 Basic rollback

Component relationships

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*].

10.12.6 FDP_ROL.2 Advanced rollback

Component relationships

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*].

10.13 Stored data confidentiality (FDP_SDC)

10.13.1 Family behaviour

This family provides requirements that address protection of user data confidentiality while the data is stored within memory areas protected by the TSF. The TSF provides access to the data in the memory through the specified interfaces only and prevents compromise of their information bypassing these interfaces. It complements the family Stored data integrity (FDP_SDI) which protects the user data from integrity errors while being stored in the memory.

10.13.2 Components leveling and description

Figure 34 shows the component leveling for this family.

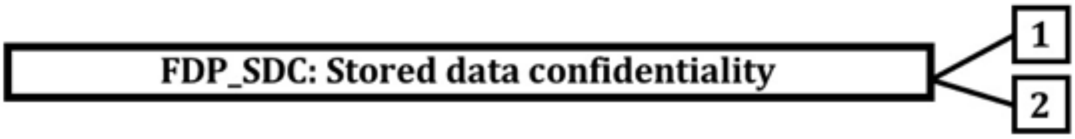

Figure 34 — FDP_SDC: Component leveling

FDP_SDC.1 Stored data confidentiality, requires the TSF to protect the confidentiality of information of the user data in specified memory areas.

FDP_SDC.2 Stored data confidentiality with dedicated method, requires the TSF to protect the confidentiality of the user data according to data characteristics leading to specify a dedicated method of protection of confidentiality.

10.13.3 Management of FDP_SDC.1, FDP_SDC.2

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

a) No specific management functions are identified

10.13.4 Audit of FDP_SDC.1, FDP_SDC.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

10.13.5 FDP_SDC.1 Stored data confidentiality

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FDP_SDC.1.1

The TSF shall ensure the confidentiality of [selection: *all user data, the following user data [assignment: list of user data]*] while it is stored in the [selection: *temporary memory, persistent memory, any memory*].

10.13.6 FDP_SDC.2 Stored data confidentiality with dedicated method

Component relationships

Hierarchical to:	No other components.
Dependencies:	FCS_COP.1.

FDP_SDC.2.1

The TSF shall ensure the confidentiality of the [selection: *all user data, the following user data [assignment: list of user data]*] according to [assignment: *data characteristics*] while it is stored under the control of the TSF.

FDP_SDC.2.2

The TSF shall ensure the confidentiality of the user data specified in FDP_SDC.2.1 without user intervention.

10.14 Stored data integrity (FDP_SDI)

10.14.1 Family behaviour

This family provides requirements that address protection of user data while it is stored within containers controlled by the TSF. 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.

10.14.2 Components leveling and description

Figure 35 shows the component leveling for this family.

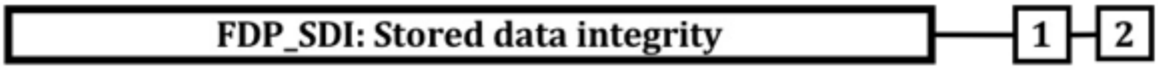

Figure 35 — FDP_SDI: Component leveling

FDP_SDI.1 Stored data integrity monitoring, requires that the TSF monitor user data stored within containers controlled by the TSF for identified integrity errors.

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.

10.14.3 Management of FDP_SDI.1

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

a) There are no management activities foreseen.

10.14.4 Management of FDP_SDI.2

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.

10.14.5 Audit of FDP_SDI.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

10.14.6 Audit of FDP_SDI.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

10.14.7 FDP_SDI.1 Stored data integrity monitoring

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FDP_SDI.1.1

The TSF shall monitor user data stored in containers controlled by the TSF for [assignment: *integrity errors*] on all objects, based on the following attributes: [assignment: *user data attributes*].

10.14.8 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 in containers controlled by the TSF 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*].

10.15 Inter-TSF user data confidentiality transfer protection (FDP_UCT)

10.15.1 Family behaviour

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.

10.15.2 Components leveling and description

Figure 36 shows the component leveling for this family.

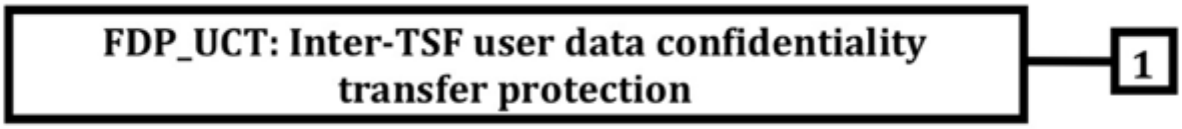
FDP_UCT: Inter-TSF user data confidentiality transfer protection
1

Figure 36 — FDP_UCT: Component leveling

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

10.15.3 Management of FDP_UCT.1

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

a) There are no management activities foreseen.

10.15.4 Audit of FDP_UCT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

b) Basic: The identity of any unauthorized 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.

10.15.5 FDP_UCT.1 Basic data exchange confidentiality

Component relationships

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 [selection: *transmit, receive*] user data in a manner protected from unauthorized disclosure.

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

10.16.1 Family behaviour

This family defines the requirements for providing integrity for user data in transit between the TOE 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.

10.16.2 Components leveling and description

Figure 37 shows the component leveling for this family.

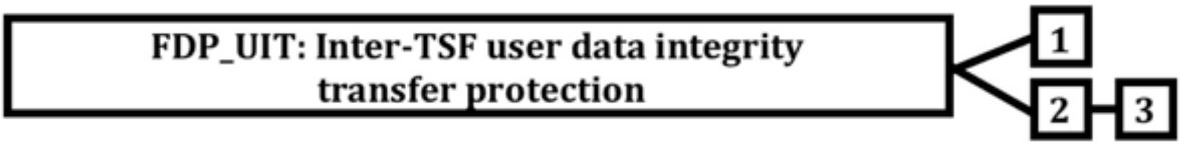

Figure 37 — FDP_UIT: Component leveling

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

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.

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.

10.16.3 Management of FDP_UIT.1, FDP_UIT.2, FDP_UIT.3

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

a) There are no management activities foreseen.

10.16.4 Audit of FDP_UIT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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 unauthorized 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.

10.16.5 Audit of FDP_UIT.2, FDP_UIT.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

b) Minimal: Successful recovery from errors including the 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 unauthorized 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.

10.16.6 FDP_UIT.1 Data exchange integrity

Component relationships

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 [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.

10.16.7 FDP_UIT.2 Source data exchange recovery

Component relationships

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.

10.16.8 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.

11 Class FIA: Identification and authentication

11.1 Class description

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

The unambiguous identification of authorized 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 authorized user. Other classes of requirements are dependent upon correct identification and authentication of users in order to be effective.

Figure 38 shows the decomposition of this class, it’s families and components. Elements are not shown in the figure.

Annex G provides explanatory information for this class and should be consulted when using the components identified in this class.

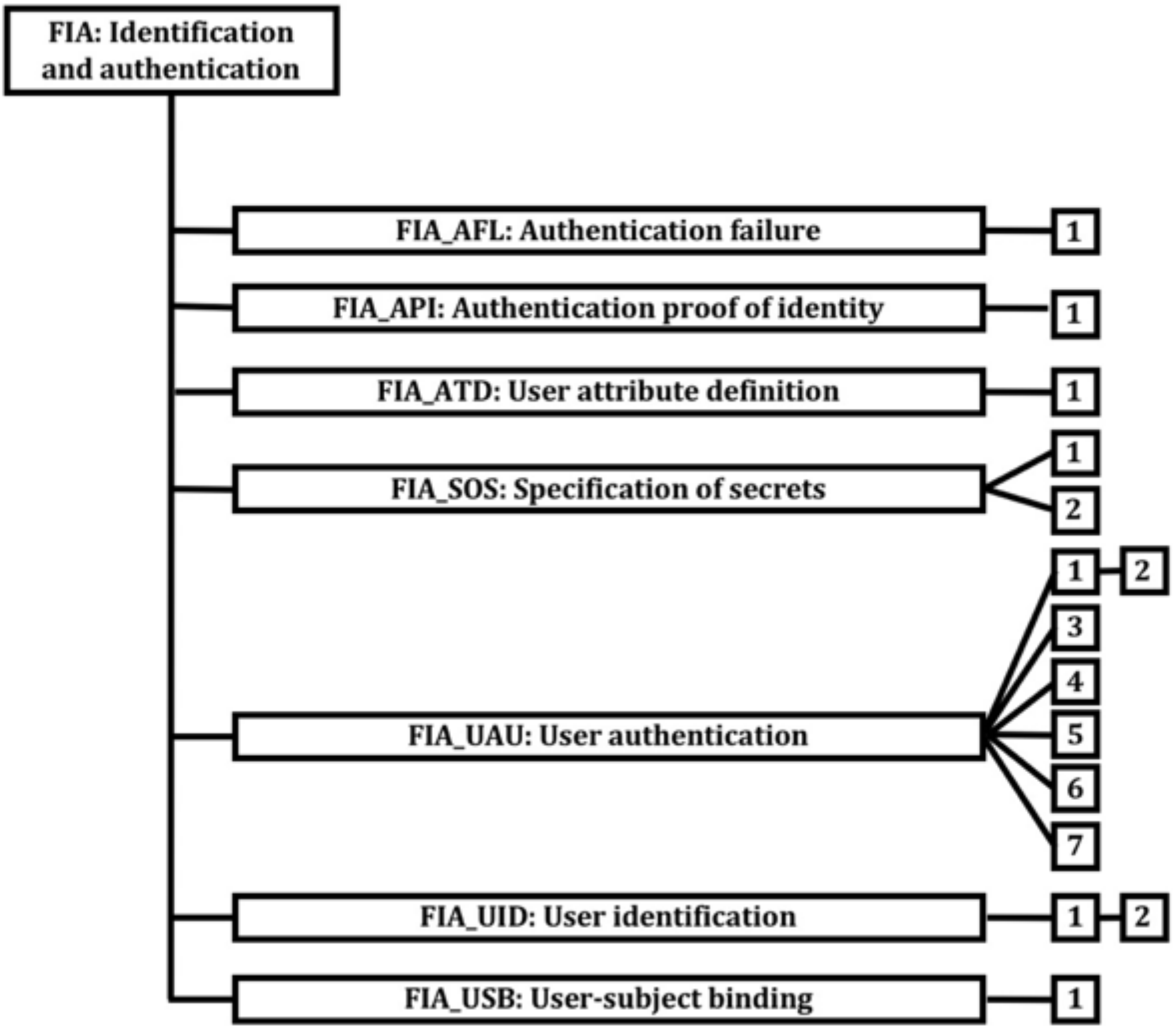
FIA: Identification and authentication
FIA_AFL: Authentication failure
1
FIA_API: Authentication proof of identity
1
FIA_ATD: User attribute definition
1
FIA_SOS: Specification of secrets
1
2
FIA_UAU: User authentication
1
2
3
4
5
6
7
FIA_UID: User identification
1
2
FIA_USB: User-subject binding
1

Figure 38 — FIA: Identification and authentication class decomposition

11.2 Authentication failures (FIA_AFL)

11.2.1 Family behaviour

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.

11.2.2 Components leveling and description

Figure 39 shows the component leveling for this family.

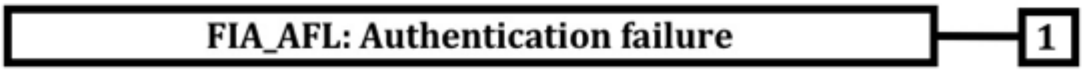

Figure 39 — FIA_AFL: Component leveling

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 from which the attempts were made until an administrator-defined condition occurs.

11.2.3 Management of FIA_AFL.1

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.

11.2.4 Audit of FIA_AFL.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: the reaching of the threshold for the unsuccessful authentication attempts and the actions taken and the subsequent, if appropriate, restoration to the normal state.

11.2.5 FIA_AFL.1 Authentication failure handling

Component relationships

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 [selection: *met, surpassed*], the TSF shall [assignment: *list of actions*].

11.3 Authentication proof of identity (FIA_API)

11.3.1 Family behaviour

This family defines functions provided by the TOE to prove its identity and so allow for verification of the TOE by an external entity in the TOE's IT environment.

11.3.2 Components leveling and description

Figure 40 shows the component leveling for this family.

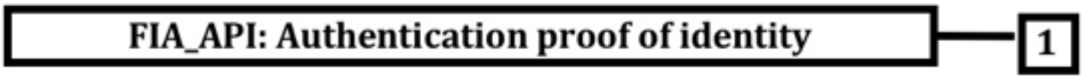

Figure 40 — FIA_API: Component leveling

FIA_API.1 Authentication Proof of Identity, provides proof of the identity of the TOE to an external entity.

11.3.3 Management of FIA_API.1

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

a) Management of authentication information used to prove the claimed identity.

11.3.4 Audit of FIA_API.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

11.3.5 FIA_API.1 Authentication proof of identity

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FIA_API.1.1

The TSF shall provide an [assignment: *authentication mechanism*] to prove the identity of the [assignment: *object, authorized user, or role*] to an external entity.

Editors' Note:

Editors observe that in many STs using this component the second completion is for "TOE" which is neither an object, authorized user or a role. Should FIA_API.1.1 be updated to allow for the specification of TOE in the assignment?

11.4 User attribute definition (FIA_ATD)

11.4.1 Family behaviour

All authorized users may have a set of security attributes, other than the user's identity, that is used to enforce the SFRs. This family defines the requirements for associating user security attributes with users as needed to support the TSF in making security decisions.

11.4.2 Components leveling and description

Figure 41 shows the component leveling for this family.

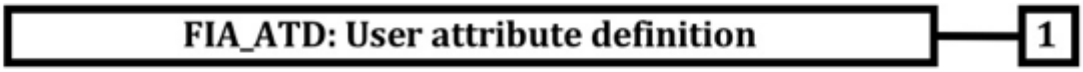

Figure 41 — FIA_ATD: Component leveling

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

11.4.3 Management of FIA_ATD.1

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

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

11.4.4 Audit of FIA_ATD.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

11.4.5 FIA_ATD.1 User attribute definition

Component relationships

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*].

11.5 Specification of secrets (FIA_SOS)

11.5.1 Family behaviour

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

11.5.2 Components leveling and description

Figure 42 shows the component leveling for this family.

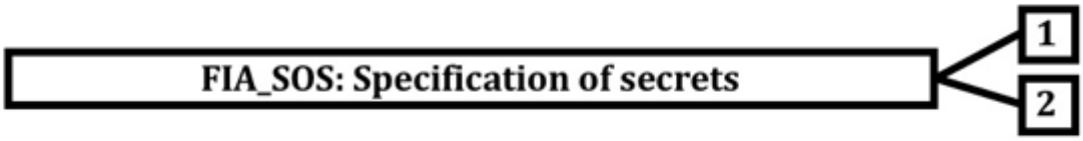

Figure 42 — FIA_SOS: Component leveling

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

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

11.5.3 Management of FIA_SOS.1

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

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

11.5.4 Management of FIA_SOS.2

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

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

11.5.5 Audit of FIA_SOS.1, FIA_SOS.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

11.5.6 FIA_SOS.1 Verification of secrets

Component relationships

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*].

11.5.7 FIA_SOS.2 TSF Generation of secrets

Component relationships

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*].

11.6 User authentication (FIA_UAU)

11.6.1 Family behaviour

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.

11.6.2 Components leveling and description

Figure 43 shows the component leveling for this family.

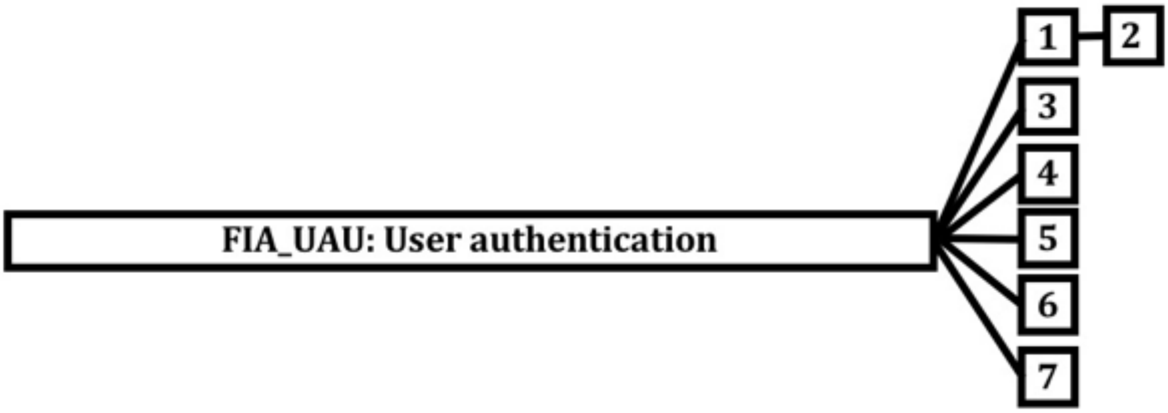

Figure 43 — FIA_UAU: Component leveling

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

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

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

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

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

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

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

11.6.3 Management of FIA_UAU.1

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.

11.6.4 Management of FIA_UAU.2

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.

11.6.5 Management of FIA_UAU.3, FIA_UAU.4, FIA_UAU.7

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

a) There are no management activities foreseen.

11.6.6 Management of FIA_UAU.5

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

a) the management of authentication mechanisms;

11.6.7 Management of FIA_UAU.6

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

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

11.6.8 Management of FIA_UAU.7

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

a) the management of the rules for authentication.

11.6.9 Audit of FIA_UAU.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

11.6.10 Audit of FIA_UAU.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Unsuccessful use of the authentication mechanism;

b) Basic: All use of the authentication mechanism.

11.6.11 Audit of FIA_UAU.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Detection of fraudulent authentication data;

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

11.6.12 Audit of FIA_UAU.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Attempts to reuse authentication data.

11.6.13 Audit of FIA_UAU.5

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: The final decision on authentication;

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

11.6.14 Audit of FIA_UAU.6

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Failure of re-authentication;

b) Basic: All re-authentication attempts.

11.6.15 Audit of FIA_UAU.7

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Well-formedness of rules regarding the semantics of rule-set;

b) Basic: verification of enforceability of rules ~~(and their writing).~~

Editors' Note:

A comment was received saying "b) should be changed to make it clearer."

Comments have been requested on this for several drafts... If no comments are received in this CD3 comment round then the text "and their writing" will be removed

11.6.16 FIA_UAU.1 Timing of authentication

Component relationships

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.

11.6.17 FIA_UAU.2 User authentication before any action

Component relationships

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.

11.6.18 FIA_UAU.3 Unforgeable authentication

Component relationships

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.

11.6.19 FIA_UAU.4 Single-use authentication mechanisms

Component relationships

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)*].

11.6.20 FIA_UAU.5 Multiple authentication mechanisms

Component relationships

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*].

11.6.21 FIA_UAU.6 Re-authenticating

Component relationships

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*].

11.6.22 FIA_UAU.7 Protected authentication feedback

Component relationships

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.

11.7 User identification (FIA_UID)

11.7.1 Family behaviour

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.

11.7.2 Components leveling and description

Figure 44 shows the component leveling for this family.

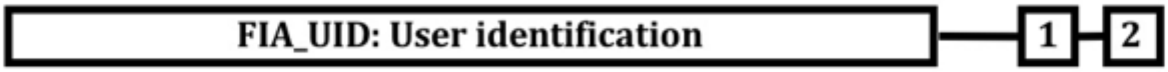

Figure 44 — FIA_UID: Component leveling

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

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

11.7.3 Management of FIA_UID.1

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

a) The management of the user identities;

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

11.7.4 Management of FIA_UID.2

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

a) The management of the user identities;

11.7.5 Audit of FIA_UID.1, FIA_UID.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

11.7.6 FIA_UID.1 Timing of identification

Component relationships

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 TSF-mediated actions on behalf of that user.

11.7.7 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 be successfully identified before allowing any TSF-mediated actions on behalf of that user.

11.8 User-subject binding (FIA_USB)

11.8.1 Family behaviour

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.

11.8.2 Components leveling and description

Figure 45 shows the component leveling for this family.

Figure 45 — FIA_USB: Component leveling

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.

11.8.3 Management of FIA_USB.1

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

a) An authorized administrator can define default subject security attributes;
b) An authorized administrator can change subject security attributes.

11.8.4 Audit of FIA_USB.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Unsuccessful binding of user security attributes to a subject
b) Basic: Success and failure of binding of user security attributes to a subject.

11.8.5 FIA_USB.1 User-subject binding

Component relationships

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*].

12 Class FMT: Security management

12.1 Class description

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.

This class has several objectives:

a) Management of TSF data;

b) Management of security attributes;

c) Management of functions of the TSF;

d) Definition of security roles.

Figure 46 shows the decomposition of this class, it’s families and components. Elements are not shown in the figure.

Annex H provides explanatory information for this class and should be consulted when using the components identified in this class.

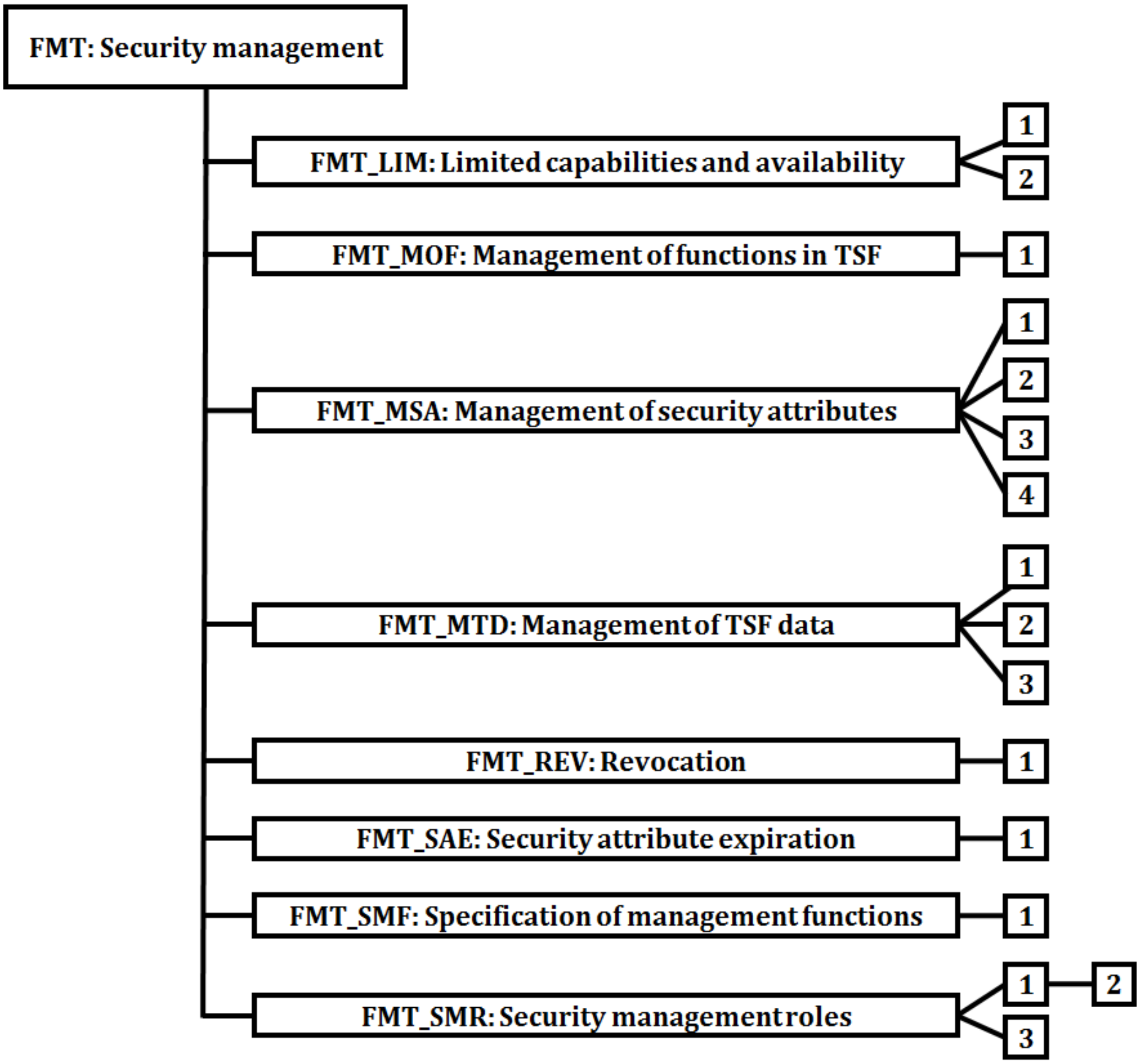
FMT: Security management
FMT_LIM: Limited capabilities and availability
1
2
FMT_MOF: Management of functions in TSF
1
FMT_MSA: Management of security attributes
1
2
3
4
FMT_MTD: Management of TSF data
1
2
3
FMT_REV: Revocation
1
FMT_SAE: Security attribute expiration
1
FMT_SMF: Specification of management functions
1
FMT_SMR: Security management roles
1
2
3

Figure 46 — FMT: Security management class decomposition

12.2 Limited capabilities and availability (FMT_LIM)

12.2.1 Family behaviour

This family defines requirements that limit the capabilities and availability of functions in a combined manner.

Note FDP_ACF restricts the access to functions whereas the component Limited Capability of this family requires the functions themselves to be designed in a specific manner.

12.2.2 Components leveling and description

Figure 47 shows the component leveling for this family.

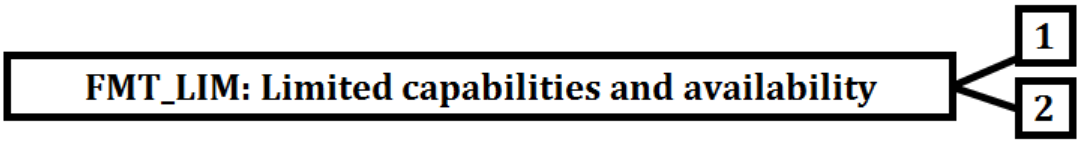

Figure 47 — FMT_LIM: Component leveling

FMT_LIM.1 Limited capabilities requires that the TSF is built to provide only the capabilities (perform action, gather information) necessary for its genuine purpose.

FMT_LIM.2 Limited availability requires that the TSF restrict the use of functions (refer to Limited capabilities (FMT_LIM.1)). This can be achieved, for instance, by removing or by disabling functions in a specific phase of the TOE’s life-cycle.

12.2.3 Management of FMT_LIM.1, FMT_LIM.2

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

a) There are no management activities foreseen.

12.2.4 Audit of FMT_LIM.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

12.2.5 FMT_LIM.1 Limited capabilities

Component relationships

Hierarchical to:	No other components.
Dependencies:	FMT_LIM.2 Limited availability

FMT_LIM.1.1

The TSF shall limit its capabilities so that in conjunction with “Limited availability (FMT_LIM.2)” the following policy is enforced [assignment: Limited capability and availability policy].

12.2.6 FMT_LIM.2 Limited availability

Component relationships

Hierarchical to:	No other components.

Dependencies:	FMT_LIM.1 Limited capabilities

FMT_LIM.2.1

The TSF shall be designed in a manner that limits its availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced [assignment: *Limited capability and availability policy*].

12.3 Management of functions in TSF (FMT_MOF)

12.3.1 Family behaviour

This family allows authorized users to control over the management of functions in the TSF.

12.3.2 Components leveling and description

Figure 48 shows the component leveling for this family.

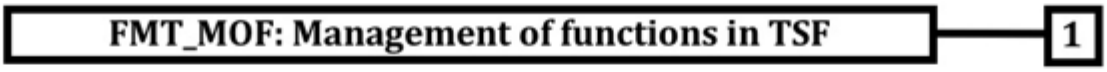

Figure 48 — FMT_MOF: Component leveling

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

12.3.3 Management of FMT_MOF.1

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.

12.3.4 Audit of FMT_MOF.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

12.3.5 FMT_MOF.1 Management of security functions behaviour

Component relationships

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 authorized identified roles*].

12.4 Management of security attributes (FMT_MSA)

12.4.1 Family behaviour

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

12.4.2 Components leveling and description

Figure 49 shows the component leveling for this family.

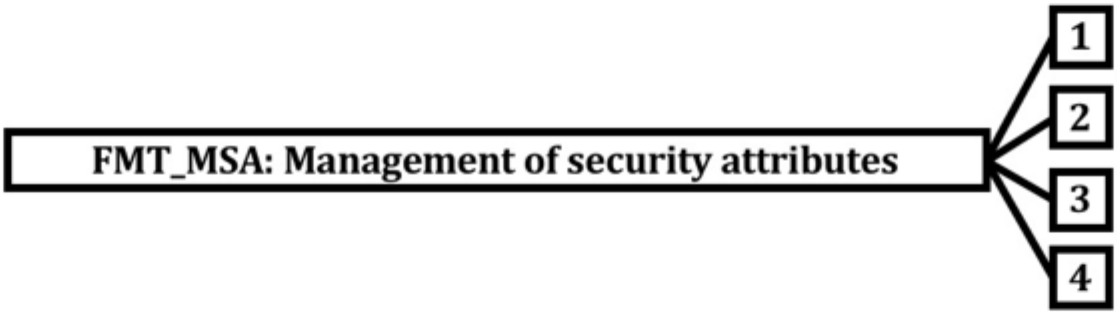

Figure 49 — FMT_MSA: Component leveling

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

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

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

FMT_MSA.4 Security attribute value inheritance allows the rules/policies to be specified that will dictate the value to be inherited by a security attribute.

12.4.3 Management of FMT_MSA.1

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;

b) Management of rules by which security attributes inherit specified values.

12.4.4 Management of FMT_MSA.2

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

a) Management of rules by which security attributes inherit specified values.

12.4.5 Management of FMT_MSA.3

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;

c) Management of rules by which security attributes inherit specified values.

12.4.6 Management of FMT_MSA.4

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

a) Specification of the role permitted to establish or modify security attributes.

12.4.7 Audit of FMT_MSA.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

12.4.8 Audit of FMT_MSA.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

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

12.4.9 Audit of FMT_MSA.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

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

12.4.10 Audit of FMT_MSA.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Basic: Modifications of security attributes, possibly with the old and/or values of security attributes that were modified.

12.4.11 FMT_MSA.1 Management of security attributes

Component relationships

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(s), information flow control SFP(s)*] to restrict the ability to [selection: c*hange_default, query, modify, delete, [assignment: other operations]*] the security attributes [assignment: *list of security attributes*] to [assignment: *the authorized identified roles*].

12.4.12 FMT_MSA.2 Secure security attributes

Component relationships

Hierarchical to:	No other components.
Dependencies:	[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 [assignment: *list of security attributes*].

12.4.13 FMT_MSA.3 Static attribute initialization

Component relationships

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 authorized identified roles*] to specify alternative initial values to override the default values when an object or information is created.

12.4.14 FMT_MSA.4 Security attribute value inheritance

Component relationships

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

FMT_MSA.4.1

The TSF shall use the following rules to set the value of security attributes: [assignment: *rules for setting the values of security attributes*].

12.5 Management of TSF data (FMT_MTD)

12.5.1 Family behaviour

This family allows authorized users (roles) control over the management of TSF data.

12.5.2 Components leveling and description

Figure 50 shows the component leveling for this family.

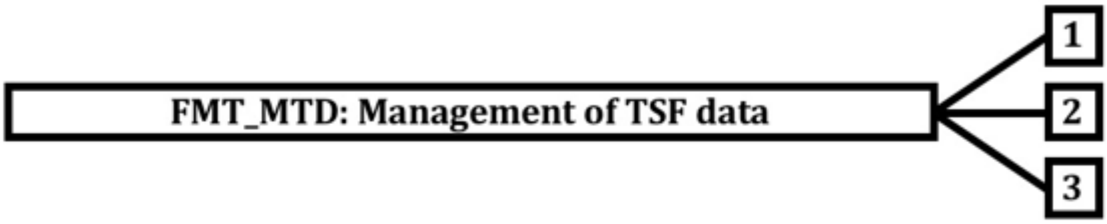

Figure 50 — FMT_MTD: Component leveling

FMT_MTD.1 Management of TSF data allows authorized users to manage TSF data.

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

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

12.5.3 Management of FMT_MTD.1

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.

12.5.4 Management of FMT_MTD.2

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.

12.5.5 Management of FMT_MTD.3

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

a) There are no management activities foreseen.

12.5.6 Audit of FMT_MTD.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

12.5.7 Audit of FMT_MTD.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

12.5.8 Audit of FMT_MTD.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: All rejected values of TSF data.

12.5.9 FMT_MTD.1 Management of TSF data

Component relationships

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 authorized identified roles*].

12.5.10 FMT_MTD.2 Management of limits on TSF data

Component relationships

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 authorized 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*].

12.5.11 FMT_MTD.3 Secure TSF data

Component relationships

Hierarchical to:	No other components.
Dependencies:	FMT_MTD.1 Management of TSF data

FMT_MTD.3.1

The TSF shall ensure that only secure values are accepted for [assignment: *list of TSF data*].

12.6 Revocation (FMT_REV)

12.6.1 Family behaviour

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

12.6.2 Components leveling and description

Figure 51 shows the component leveling for this family.

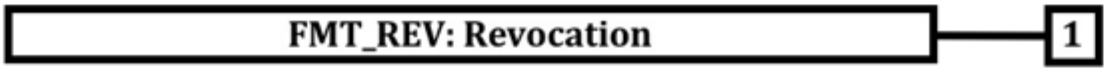

Figure 51 — FMT_REV: Component leveling

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

12.6.3 Management of FMT_REV.1

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.

12.6.4 Audit of FMT_REV.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Unsuccessful revocation of security attributes;

b) Basic: All attempts to revoke security attributes.

12.6.5 FMT_REV.1 Revocation

Component relationships

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles

FMT_REV.1.1

The TSF shall restrict the ability to revoke [assignment: *list of security attributes*] associated with the [selection: *users, subjects, objects, [assignment: other additional resources]*] under the control of the TSF to [assignment: *the authorized identified roles*].

FMT_REV.1.2

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

12.7 Security attribute expiration (FMT_SAE)

12.7.1 Family behaviour

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

12.7.2 Components leveling and description

Figure 52 shows the component leveling for this family.

Figure 52 — FMT_SAE: Component leveling

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

12.7.3 Management of FMT_SAE.1

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.

12.7.4 Audit of FMT_SAE.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

b) Basic: Action taken due to attribute expiration.

12.7.5 FMT_SAE.1 Time-limited authorization

Component relationships

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 authorized 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.

12.8 Specification of Management Functions (FMT_SMF)

12.8.1 Family behaviour

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.

12.8.2 Components leveling and description

Figure 53 shows the component leveling for this family.

Figure 53 — FMT_SMF: Component leveling

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

12.8.3 Management of FMT_SMF.1

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

a) There are no management activities foreseen.

12.8.4 Audit of FMT_SMF.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Use of the management functions.

12.8.5 FMT_SMF.1 Specification of Management Functions

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FMT_SMF.1.1

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

12.9 Security management roles (FMT_SMR)

12.9.1 Family behaviour

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.

12.9.2 Components leveling and description

Figure 54 shows the component leveling for this family.

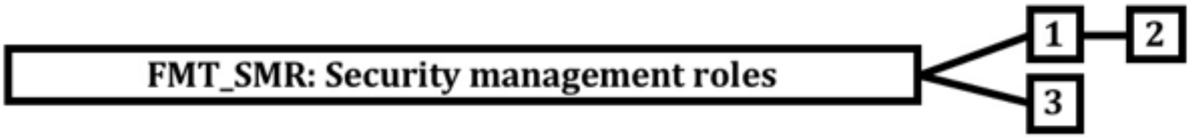

Figure 54 — FMT_SMR: Component leveling

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

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.

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

12.9.3 Management of FMT_SMR.1

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

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

12.9.4 Management of FMT_SMR.2

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.

12.9.5 Management of FMT_SMR.3

There are no management activities foreseen.

12.9.6 Audit of FMT_SMR.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

12.9.7 Audit of FMT_SMR.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

12.9.8 Audit of FMT_SMR.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: explicit request to assume a role.

12.9.9 FMT_SMR.1 Security roles

Component relationships

Hierarchical to:	No other components.
Dependencies:	FIA_UID.1 Timing of identification

FMT_SMR.1.1

The TSF shall maintain the roles [assignment: *the authorized identified roles*].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

12.9.10 FMT_SMR.2 Restrictions on security roles

Component relationships

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: ***authorized*** *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.

12.9.11 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*].

13 Class FPR: Privacy

13.1 Class description

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

Figure 55 shows the decomposition of this class, it’s families and components. Elements are not shown in the figure.

Annex I provides explanatory information for this class and should be consulted when using the components identified in this class.

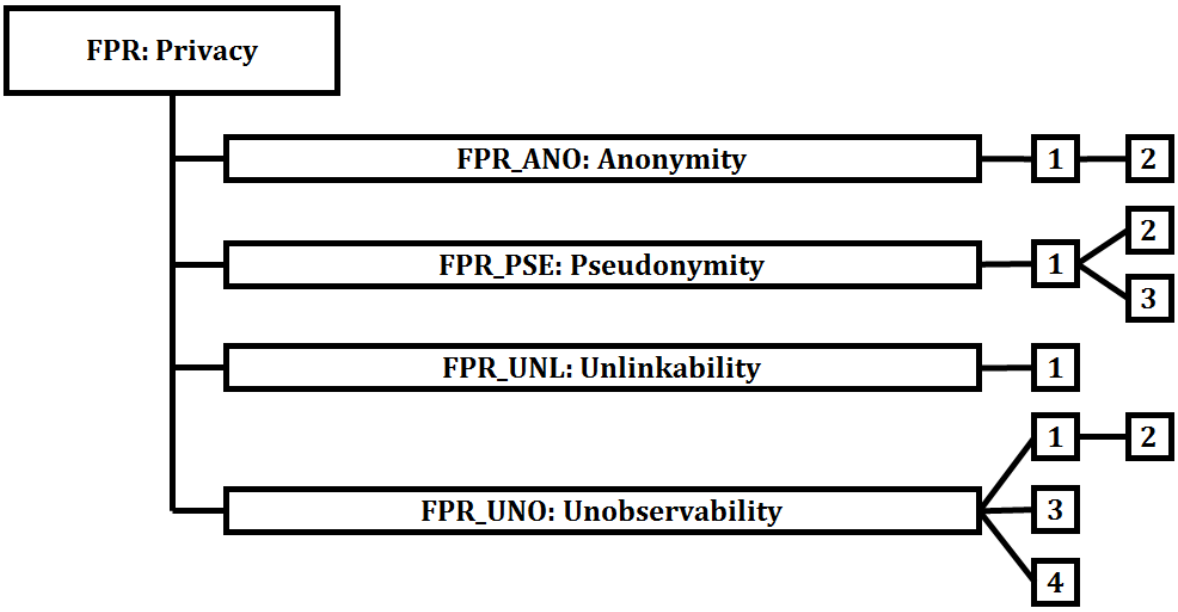

Figure 55 — FPR: Privacy class decomposition

13.2 Anonymity (FPR_ANO)

13.2.1 Family behaviour

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.

13.2.2 Components leveling and description

Figure 56 shows the component leveling for this family.

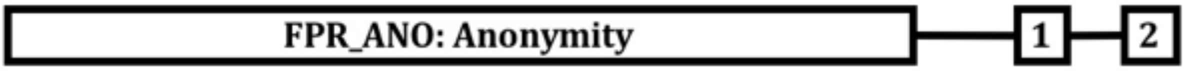

Figure 56 — FPR_ANO: Component leveling

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.

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.

13.2.3 Management of FPR_ANO.1, FPR_ANO.2

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

a) There are no management activities foreseen.

13.2.4 Audit of FPR_ANO.1, FPR_ANO.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: The invocation of the anonymity mechanism.

13.2.5 FPR_ANO.1 Anonymity

Component relationships

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*].

13.2.6 FPR_ANO.2 Anonymity without soliciting information

Component relationships

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.

13.3 Pseudonymity (FPR_PSE)

13.3.1 Family behaviour

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.

13.3.2 Components leveling and description

Figure 57 shows the component leveling for this family.

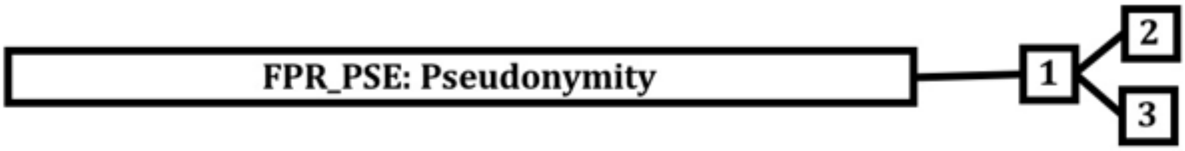

Figure 57 — FPR_PSE: Component leveling

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.

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

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

13.3.3 Management of FPR_PSE.1, FPR_PSE.2, FPR_PSE.3

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

a) There are no management activities foreseen.

13.3.4 Audit of FPR_PSE.1, FPR_PSE.2, FPR_PSE.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

13.3.5 FPR_PSE.1 Pseudonymity

Component relationships

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*].

13.3.6 FPR_PSE.2 Reversible pseudonymity

Component relationships

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 authorized 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*].

13.3.7 FPR_PSE.3 Alias pseudonymity

Component relationships

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.

13.4 Unlinkability (FPR_UNL)

13.4.1 Family behaviour

This family ensures that selected entities may be linked together without external entities being able to back trace these links.

13.4.2 Components leveling and description

Figure 58 shows the component leveling for this family.

FPR_UNL: Unlinkability
1

Figure 58 — FPR_UNL: Component leveling

FPR_UNL.1 Unlinkability of operations requires that users and/or subjects are unable to determine whether the same user caused certain specific operations in the system, or whether operations are related in some other manner. This component ensures that users cannot link different operations in the system and thereby obtain information.

13.4.3 Management of FPR_UNL.1

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

a) The management of the unlinkability function.

13.4.4 Audit of FPR_UNL.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: The invocation of the unlinkability mechanism.

13.4.5 FPR_UNL.1 Unlinkability of operations

Component relationships

Hierarchical to: No other components.

Dependencies: No dependencies.

FPR_UNL.1.1

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

NOTE This SFR does not only stipulate at the individual set of operations performed by one entity. This SFR intends to look at a chain of interlinked operations by multiple entities. This chain can be subsumed as a transaction.

13.5 Unobservability (FPR_UNO)

13.5.1 Family behaviour

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.

13.5.2 Components leveling and description

Figure 59 shows the component leveling for this family.

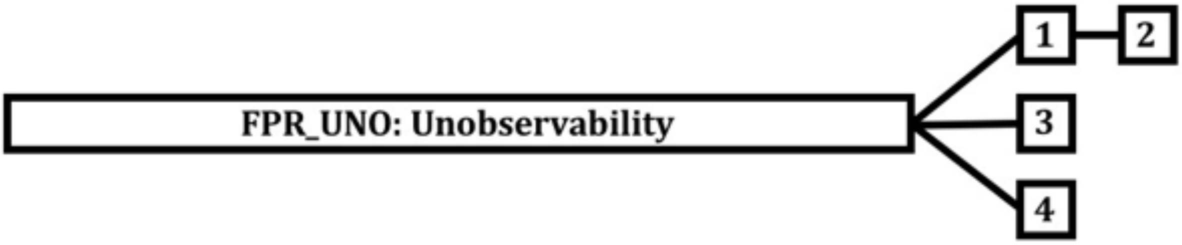

Figure 59 — FPR_UNO: Component leveling

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

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.

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

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

13.5.3 Management of FPR_UNO.1, FPR_UNO.2

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

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

13.5.4 Management of FPR_UNO.3

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

a) There are no management activities foreseen.

13.5.5 Management of FPR_UNO.4

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

a) The list of authorized users that are capable of determining the occurrence of operations.

13.5.6 Audit of FPR_UNO.1, FPR_UNO.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: The invocation of the unobservability mechanism.

13.5.7 Audit of FPR_UNO.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

13.5.8 Audit of FPR_UNO.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

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

13.5.9 FPR_UNO.1 Unobservability

Component relationships

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*].

13.5.10 FPR_UNO.2 Allocation of information impacting unobservability

Component relationships

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*].

13.5.11 FPR_UNO.3 Unobservability without soliciting information

Component relationships

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*].

13.5.12 FPR_UNO.4 Authorized user observability

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPR_UNO.4.1

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

14 Class FPT: Protection of the TSF

14.1 Class description

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF 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.

From the point of view of this class, regarding to the TSF there are three significant elements:

a) The TSF's implementation, which executes and implements the mechanisms that enforce the SFRs.

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

c) The external entities that the TSF may interact with in order to enforce the SFRs.

Figure 60 shows the decomposition of this class, it’s families and components. Elements are not shown in the figure.

Annex J provides explanatory information for this class and should be consulted when using the components identified in this class.

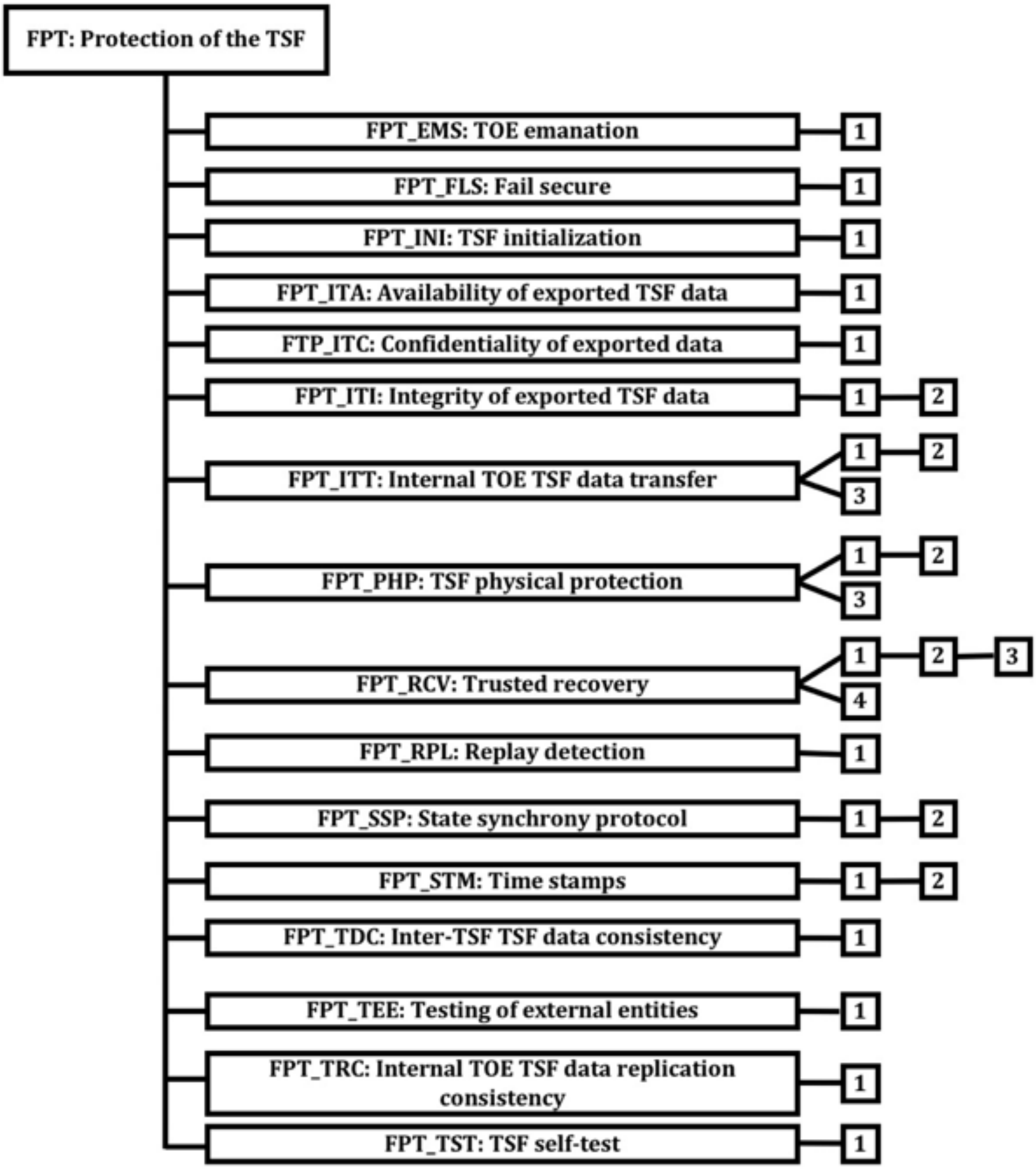
FPT: Protection of the TSF
FPT_EMS: TOE emanation
1
FPT_FLS: Fail secure
1
FPT_INI: TSF initialization
1
FPT_ITA: Availability of exported TSF data
1
FTP_ITC: Confidentiality of exported data
1
FPT_ITI: Integrity of exported TSF data
1
2
FPT_ITT: Internal TOE TSF data transfer
1
2
3
FPT_PHP: TSF physical protection
1
2
3
FPT_RCV: Trusted recovery
1
2
3
4
FPT_RPL: Replay detection
1
FPT_SSP: State synchrony protocol
1
2
FPT_STM: Time stamps
1
2
FPT_TDC: Inter-TSF TSF data consistency
1
FPT_TEE: Testing of external entities
1
FPT_TRC: Internal TOE TSF data replication consistency
1
FPT_TST: TSF self-test
1

Figure 60 — FPT: Protection of the TSF class decomposition

14.2 TOE emanation (FPT_EMS)

14.2.1 Family behaviour

The family FPT_EMS (TOE Emanation) of the Class FPT (Protection of the TSF) is defined here to describe the IT security functional requirements of the TOE related to leakage of information based on emanation.

The TOE shall prevent attacks against the TOE and secret data processed by the TOE where the attack is based on external observable phenomena of the TOE during its operation. Hereby, different types of emissions and interfaces of the TOE as well as different types of TSF data and user data may be addressed.

EXAMPLE

Examples of such attacks against the TOE and its processed secret data are simple power analysis (SPA), differential power analysis (DPA), simple electromagnetic analysis (SEMA), differential electromagnetic analysis (DEMA), timing attacks, padding oracle attacks, cache miss attacks, etc.

This family describes the functional requirements for the limitation of intelligible emanations which are not directly addressed by any other component of ISO/IEC 15408-2.

14.2.2 Components leveling and description

Figure 61 shows the component leveling for this family.

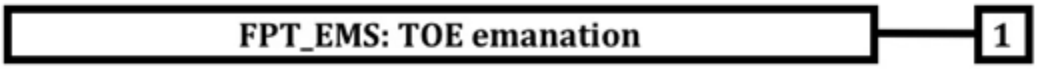

Figure 61 — FPT_EMS: Component leveling

This family consists of only one component, FPT_EMS.1 Emanation of TSF and User data, which defines requirements for the TOE to mitigate intelligible emanations.

14.2.3 Management of FPT_EMS.1

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

a) There are no management activities foreseen.

14.2.4 Audit of FPT_EMS.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

14.2.5 FPT_EMS.1 Emanation of TSF and User data

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPT_EMS.1.1

The TSF shall ensure that the TOE does not emit emissions over its attack surface in such amount that these emissions enable access to TSF data and user data as specified in the following table:

FPT_EMS.1.1 Table

ID	Emissions	attack surface	TSF data	User data
1	**[assignment: *list of types of emissions*]**	**[assignment: *list of types of attack surface*]**	**[assignment: *list of types of TSF data*]**	**[assignment: *list of types of user data*]**
...	**...**	**...**	**...**	**...**

14.3 Fail secure (FPT_FLS)

14.3.1 Family behaviour

The requirements of this family ensure that the TOE will always enforce its SFRs in the event of identified categories of failures in the TSF.

14.3.2 Components leveling and description

Figure 62 shows the component leveling for this family.

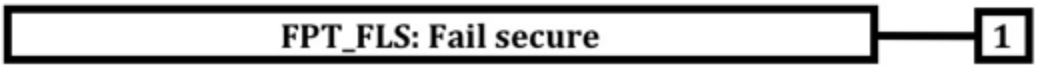

Figure 62 — FPT_FLS: Component leveling

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.

14.3.3 Management of FPT_FLS.1

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

a) There are no management activities foreseen.

14.3.4 Audit of FPT_FLS.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or /ST:

a) Basic: Failure of the TSF.

14.3.5 FPT_FLS.1 Failure with preservation of secure state

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

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*].

14.4 TSF initialization (FPT_INI)

14.4.1 Family behaviour

This family describes the functional requirements for the initialization of the TSF by a dedicated function of the TOE that ensures the initialization in a correct and secure operational state.

14.4.2 Components leveling and description

Figure 63 shows the component leveling for this family.

Figure 63 — FPT_INI: Component leveling

This family consists of only one component, Component FPT_INI.1. This component requires the TOE to provide a TSF initialization function that brings the TSF into a secure operational state at power-on.

14.4.3 Management of FPT_INI.1

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

a) There are no management activities foreseen.

14.4.4 Audit of FPT_INI.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or /ST:

a) There are no auditable events foreseen.

14.4.5 FPT_INI.1 TSF initialization

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPT_INI.1.1

The TOE shall provide an initialization function which is self-protected for integrity and authenticity.

FPT_INI.1.2

The TOE initialization function shall ensure that certain properties hold on certain elements immediately before establishing the TSF in a secure initial state, as specified below:

FPT_INI.1.2 Table

ID	Properties	Elements
1	**[assignment:** ***property, for instance authenticity, integrity, correct version*****]**	**[assignment:** ***list of TSF/user firmware, software or data*****]**
...	**...**	**...**

FPT_INI.1.3

The TOE initialization function shall detect and respond to errors and failures during initialization such that the TOE [selection: ***is halted, successfully completes initialization with [selection: reduced functionality, signaling error state, [assignment: list of actions]*****].**

FPT_INI.1.4

The TOE initialization function shall only interact with the TSF in [assignment: ***defined methods*****] during initialization.**

14.5 Availability of exported TSF data (FPT_ITA)

14.5.1 Family behaviour

This family defines the rules for the prevention of loss of availability of TSF data moving between the TSF and another trusted IT product.

14.5.2 Components leveling and description

Figure 64 shows the component leveling for this family.

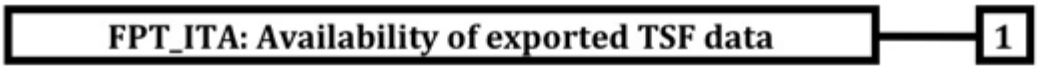

Figure 64 — FPT_ITA: Component leveling

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 another trusted IT product.

14.5.3 Management of FPT_ITA.1

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 another trusted IT product.

14.5.4 Audit of FPT_ITA.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: the absence of TSF data when required by a TOE.

14.5.5 FPT_ITA.1 Inter-TSF availability within a defined availability metric

Component relationships

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 another trusted IT product within [assignment: *a defined availability metric*] given the following conditions [assignment: *conditions to ensure availability*].

14.6 Confidentiality of exported TSF data (FPT_ITC)

14.6.1 Family behaviour

This family defines the rules for the protection from unauthorized disclosure of TSF data during transmission between the TSF and another trusted IT product.

14.6.2 Components leveling and description

Figure 65 shows the component leveling for this family.

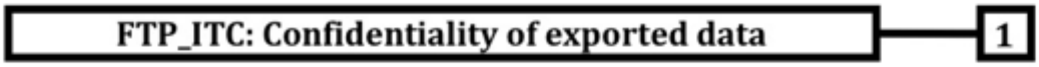

Figure 65 — FPT_ITC: Component leveling

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 another trusted IT product is protected from disclosure while in transit.

14.6.3 Management of FPT_ITC.1

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

a) There are no management activities foreseen.

14.6.4 Audit of FPT_ITC.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

14.6.5 FPT_ITC.1 Inter-TSF confidentiality during transmission

Component relationships

Hierarchical to: No other components.
Dependencies: No dependencies.

FPT_ITC.1.1

The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product from unauthorized disclosure during transmission.

14.7 Integrity of exported TSF data (FPT_ITI)

14.7.1 Family behaviour

This family defines the rules for the protection, from unauthorized modification, of TSF data during transmission between the TSF and another trusted IT product.

14.7.2 Components leveling and description

Figure 66 shows the component leveling for this family.

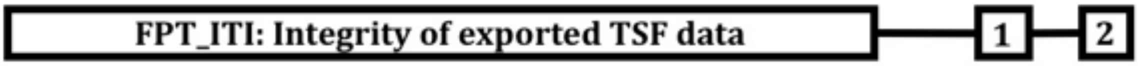

Figure 66 — FPT_ITI: Component leveling

FPT_ITI.1 Inter-TSF detection of modification, provides the ability to detect modification of TSF data during transmission between the TSF and another trusted IT product, under the assumption that another trusted IT product is cognizant of the mechanism used.

FPT_ITI.2 Inter-TSF detection and correction of modification, provides the ability for another trusted IT product not only to detect modification, but to correct modified TSF data under the assumption that another trusted IT product is cognizant of the mechanism used.

14.7.3 Management of FPT_ITI.1

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

a) There are no management activities foreseen.

14.7.4 Management of FPT_ITI.2

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

a) Management of the types of TSF data that the TSF tries to correct if modified in transit;

b) Management of the types of action that the TSF takes if TSF data is modified in transit.

14.7.5 Audit of FPT_ITI.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: the detection of modification of transmitted TSF data.

b) Basic: the action taken upon detection of modification of transmitted TSF data.

14.7.6 Audit of FPT_ITI.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

14.7.7 FPT_ITI.1 Inter-TSF detection of modification

Component relationships

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 another 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 another trusted IT product and perform [assignment: *action to be taken*] if modifications are detected.

14.7.8 FPT_ITI.2 Inter-TSF detection and correction of modification

Component relationships

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 another 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 another 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 another trusted IT product.

14.8 Internal TOE TSF data transfer (FPT_ITT)

14.8.1 Family behaviour

This family provides requirements that address protection of TSF data when it is transferred between separate parts of a TOE across an internal channel.

14.8.2 Components leveling and description

Figure 67 shows the component leveling for this family.

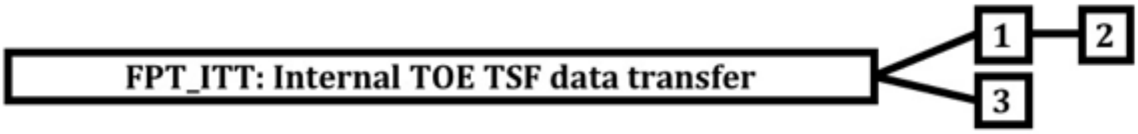

Figure 67 — FPT_ITT: Component leveling

FPT_ITT.1 Basic internal TSF data transfer protection, requires that TSF data be protected when transmitted between separate parts of the TOE.

FPT_ITT.2 TSF data transfer separation, requires that the TSF separate user data from TSF data during transmission.

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.

14.8.3 Management of FPT_ITT.1

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.

14.8.4 Management of FPT_ITT.2

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.

14.8.5 Management of FPT_ITT.3

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 actions that will be taken.

14.8.6 Audit of FPT_ITT.1, FPT_ITT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

14.8.7 Audit of FPT_ITT.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: the detection of modification of TSF data;

b) Basic: the action taken following detection of an integrity error.

14.8.8 FPT_ITT.1 Basic internal TSF data transfer protection

Component relationships

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.

14.8.9 FPT_ITT.2 TSF data transfer separation

Component relationships

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.

14.8.10 FPT_ITT.3 TSF data integrity monitoring

Component relationships

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*].

14.9 TSF physical protection (FPT_PHP)

14.9.1 Family behaviour

TSF physical protection components refer to restrictions on unauthorized physical access to the TSF, and to the deterrence of, and resistance to, unauthorized physical modification, or substitution of the TSF.

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.

14.9.2 Components leveling and description

Figure 68 shows the component leveling for this family.

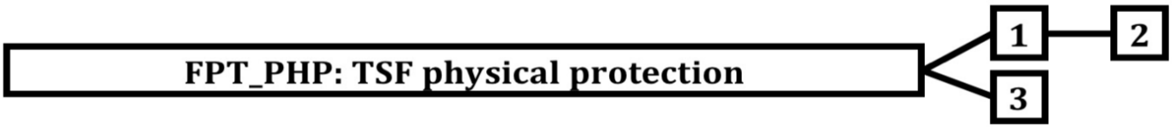

Figure 68 — FPT_PHP: Component leveling

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 authorized user must invoke a security administrative function or perform manual inspection to determining if tampering has occurred.

FPT_PHP.2 Notification of physical attack, provides for automatic notification of tampering for an identified subset of physical penetrations.

FPT_PHP.3 Resistance to physical attack, provides for features that prevent or resist physical tampering with TSF devices and TSF elements.

14.9.3 Management of FPT_PHP.1

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.

14.9.4 Management of FPT_PHP.2

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.

14.9.5 Management of FPT_PHP.3

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

a) Management of the automatic responses to physical tampering.

14.9.6 Audit of FPT_PHP.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: if detection by IT means, detection of intrusion.

14.9.7 Audit of FPT_PHP.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: detection of intrusion.

14.9.8 Audit of FPT_PHP.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

14.9.9 FPT_PHP.1 Passive detection of physical attack

Component relationships

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.

14.9.10 FPT_PHP.2 Notification of physical attack

Component relationships

Hierarchical to:	FPT_PHP.1 Passive detection of physical attack
Dependencies:	FMT_LIM.1 Limited capabilities

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.

14.9.11 FPT_PHP.3 Resistance to physical attack

Component relationships

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 SFRs are always enforced.

14.10 Trusted recovery (FPT_RCV)

14.10.1 Family behaviour

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.

14.10.2 Components leveling and description

Figure 69 shows the component leveling for this family.

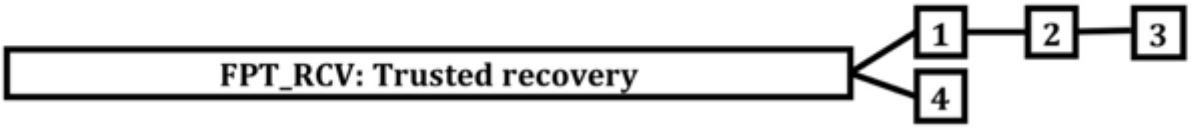

Figure 69 — FPT_RCV: Component leveling

FPT_RCV.1 Manual recovery, allows a TOE to only provide mechanisms that involve human intervention to return to a secure state.

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 that can require human intervention.

FPT_RCV.3 Automated recovery without undue loss, also provides for automated recovery, but strengthens the requirements by disallowing undue loss of protected objects.

FPT_RCV.4 Function recovery, provides for recovery at the level of particular functions, ensuring either successful completion or rollback of TSF data to a secure state.

14.10.3 Management of FPT_RCV.1

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.

14.10.4 Management of FPT_RCV.2, FPT_RCV.3

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.

14.10.5 Management of FPT_RCV.4

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

a) There are no management activities foreseen.

14.10.6 Audit of FPT_RCV.1, FPT_RCV.2, FPT_RCV.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

14.10.7 Audit of FPT_RCV.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: if possible, the impossibility to return to a secure state after a failure of the TSF;

b) Basic: if possible, the detection of a failure of a function.

14.10.8 FPT_RCV.1 Manual recovery

Component relationships

Hierarchical to:	No other components.
Dependencies:	AGD_OPE.1 Operational user guidance

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.

14.10.9 FPT_RCV.2 Automated recovery

Component relationships

Hierarchical to:	FPT_RCV.1 Manual recovery
Dependencies:	AGD_OPE.1 Operational user guidance

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.

14.10.10 FPT_RCV.3 Automated recovery without undue loss

Component relationships

Hierarchical to:	FPT_RCV.2 Automated recovery
Dependencies:	AGD_OPE.1 Operational user guidance

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 under the control of the TSF.

FPT_RCV.3.4

The TSF shall provide the capability to determine the objects that were or were not capable of being recovered.

14.10.11 FPT_RCV.4 Function recovery

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPT_RCV.4.1

The TSF shall ensure that [assignment: *list of functions and failure scenarios*] have the property that the function either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

14.11 Replay detection (FPT_RPL)

14.11.1 Family behaviour

This family addresses detection of replay for various types of entities and subsequent actions to correct. In the case where replay may be detected, this effectively prevents it.

14.11.2 Components leveling and description

Figure 70 shows the component leveling for this family.

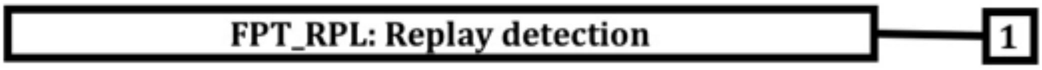

Figure 70 — FPT_RPL: Component leveling

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.

14.11.3 Management of FPT_RPL.1

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

a) Management of the list of identified entities for which replay is detected;

b) Management of the list of actions that need to be taken in case of replay.

14.11.4 Audit of FPT_RPL.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Basic: Detected replay attacks.

b) Detailed: Action to be taken based on the specific actions.

14.11.5 FPT_RPL.1 Replay detection

Component relationships

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.

14.12 State synchrony protocol (FPT_SSP)

14.12.1 Family behaviour

Distributed TOEs can give rise to greater complexity than monolithic TOEs through the potential for differences in state between parts of the TOE, and through delays in communication. In most cases synchronization 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.

State synchrony protocol (FPT_SSP) establishes the requirement for certain critical functions of the TSF to use this trusted protocol. State synchrony protocol (FPT_SSP) ensures that two distributed parts of the TOE have synchronized their states after a security-relevant action.

14.12.2 Components leveling and description

Figure 71 shows the component leveling for this family.

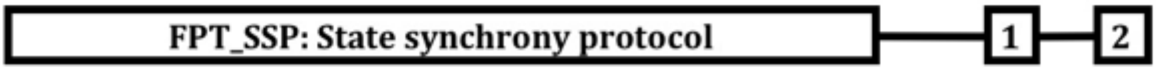

Figure 71 — FPT_SSP: Component leveling

FPT_SSP.1 Simple trusted acknowledgement, requires only a simple acknowledgment by the data recipient.

FPT_SSP.2 Mutual trusted acknowledgement, requires mutual acknowledgment of the data exchange.

14.12.3 Management of FPT_SSP.1, FPT_SSP.2

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

a) There are no management activities foreseen.

14.12.4 Audit of FPT_SSP.1, FPT_SSP.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: failure to receive an acknowledgement when expected.

14.12.5 FPT_SSP.1 Simple trusted acknowledgement

Component relationships

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.

14.12.6 FPT_SSP.2 Mutual trusted acknowledgement

Component relationships

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.

14.13 Time stamps (FPT_STM)

14.13.1 Family behaviour

This family addresses requirements for a reliable time stamp function within a TOE.

14.13.2 Components leveling and description

Figure 72 shows the component leveling for this family.

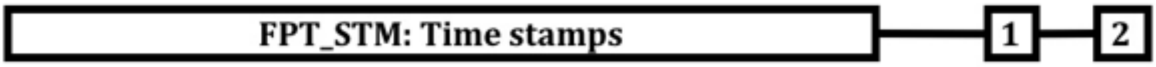

Figure 72 — FPR_STM: Component leveling

FPT_STM.1 Reliable time stamps, requires that the TSF provide reliable time stamps for TSF functions.

FPT_STM.2 Time source, requires the description of the time source used in timestamps

14.13.3 Management of FPT_STM.1

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

a) Management of the time.

14.13.4 Management of FPT_STM.2

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

a) Setting of time by user authorized according to security policy.

14.13.5 Audit of FPT_STM.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: changes to the time.

b) Detailed: providing a timestamp.

14.13.6 Audit of FPT_STM.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: discontinuous changes to the time;

b) Detailed: changes to the time source.

14.13.7 FPT_STM.1 Reliable time stamps

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPT_STM.1.1

The TSF shall be able to provide reliable time stamps.

14.13.8 FPT_STM.2 Time source

Component relationships

Hierarchical to:	No other components.
Dependencies:	FPT_STM.1 Reliable time stamps FMT_SMR.1 Security roles

FPT_STM.2.1

The TSF shall allow the [assignment: *user authorized by security policy*] to [assignment: *set the time, configure another time source*]].

14.14 Inter-TSF TSF data consistency (FPT_TDC)

14.14.1 Family behaviour

In a distributed environment, a TOE may need to exchange TSF data 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.

14.14.2 Components leveling and description

Figure 73 shows the component leveling for this family.

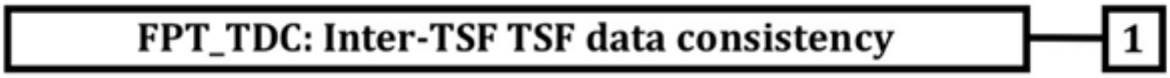

Figure 73 — FPT_TDC: Component leveling

FPT_TDC.1 Inter-TSF basic TSF data consistency, requires that the TSF provide the capability to ensure consistency of attributes between TSFs.

14.14.3 Management of FPT_TDC.1

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

a) There are no management activities foreseen.

14.14.4 Audit of FPT_TDC.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

14.14.5 FPT_TDC.1 Inter-TSF basic TSF data consistency

Component relationships

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.

14.15 Testing of external entities (FPT_TEE)

14.15.1 Family behaviour

This family defines requirements for the TSF to perform tests on one or more external entities.

This component is not intended to be applied to human users.

External entities can include applications running on the TOE, hardware or software running “underneath” the TOE (platforms, operating systems etc.) or applications/boxes connected to the TOE (intrusion detection systems, firewalls, login servers, time servers etc.).

14.15.2 Components leveling and description

Figure 74 shows the component leveling for this family.

FPT_TEE: Testing of external entities
1

Figure 74 — FPT_TEE: Component leveling

FPT_TEE.1 Testing of external entities, provides for testing of the external entities by the TSF.

14.15.3 Management of FPT_TEE.1

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

a) Management of the conditions under which the testing of external entities occurs, such as during initial start-up, regular interval, or under specified conditions;

b) Management of the time interval if appropriate.

14.15.4 Audit of FPT_TEE.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Basic: Execution of the tests of the external entities and the results of the tests.

14.15.5 FPT_TEE.1 Testing of external entities

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPT_TEE.1.1

The TSF shall run a suite of tests [selection: *during initial start-up, periodically during normal operation, at the request of an authorized user, [assignment: other conditions]*] to check the fulfillment of [assignment: *list of properties of the external entities*].

FPT_TEE.1.2

If the test fails, the TSF shall [assignment: *action(s)*].

14.16 Internal TOE TSF data replication consistency (FPT_TRC)

14.16.1 Family behaviour

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.

14.16.2 Components leveling and description

Figure 75 shows the component leveling for this family.

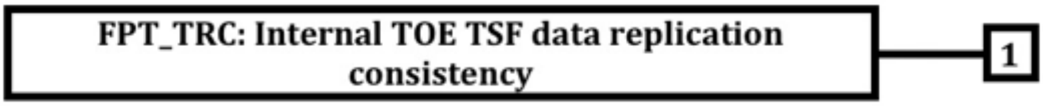

Figure 75 — FPT_TRC: Component leveling

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.

14.16.3 Management of FPT_TRC.1

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

a) There are no management activities foreseen.

14.16.4 Audit of FPT_TRC.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: restoring consistency upon reconnection;

b) Basic: Detected inconsistency between TSF data.

14.16.5 FPT_TRC.1 Internal TSF consistency

Component relationships

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 functions dependent on TSF data replication consistency*].

14.17 TSF self-test (FPT_TST)

14.17.1 Family behaviour

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 authorized 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.

The requirements of this family are also needed to detect the corruption of TSF data and TSF itself (i.e. TSF executable code or TSF hardware component) 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 cannot 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.

14.17.2 Components leveling and description

Figure 76 shows the component leveling for this family.

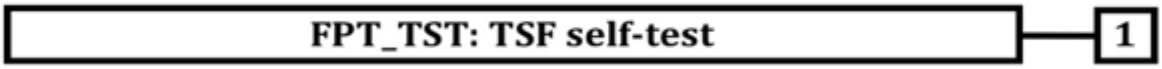

Figure 76 — FPT_TST: Component leveling

FPT_TST.1 TSF self-testing, provides the ability to test the TSF's correct operation. These tests can be performed at start-up, periodically, at the request of the authorized user, or when other conditions are met. It also provides the ability to verify the integrity of TSF data and TSF itself.

14.17.3 Management of FPT_TST.1

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.

14.17.4 Audit of FPT_TST.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Indication that the TSF self-tests were completed and any failures of the tests.

b) Basic: Execution of the TSF self-tests and the results of the tests.

14.17.5 FPT_TST.1 TSF self-testing

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FPT_TST.1.1

The TSF shall run a suite of the following self-tests [selection: *during initial start-up, periodically during normal operation, at the request of the authorized user, at the conditions [assignment: conditions under which self-test should occur]*] to demonstrate the correct operation of [selection: *[assignment: parts of TSF], the TSF*]: [assignment: *list of self-tests run by the TSF*].

FPT_TST.1.2

The TSF shall provide authorized users with the capability to verify the integrity of [selection: *[assignment: parts of TSF data], TSF data*].

FPT_TST.1.3

The TSF shall provide authorized users with the capability to verify the integrity of [selection: *[assignment: parts of TSF], TSF*].

15 Class FRU: Resource utilization

15.1 Class description

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 monopolized by lower priority tasks. The family Resource Allocation provides limits on the use of available resources, therefore preventing users from monopolizing the resources.

Figure 77 shows the decomposition of this class, it’s families and components. Elements are not shown in the figure.

Annex K provides explanatory information for this class and should be consulted when using the components identified in this class.

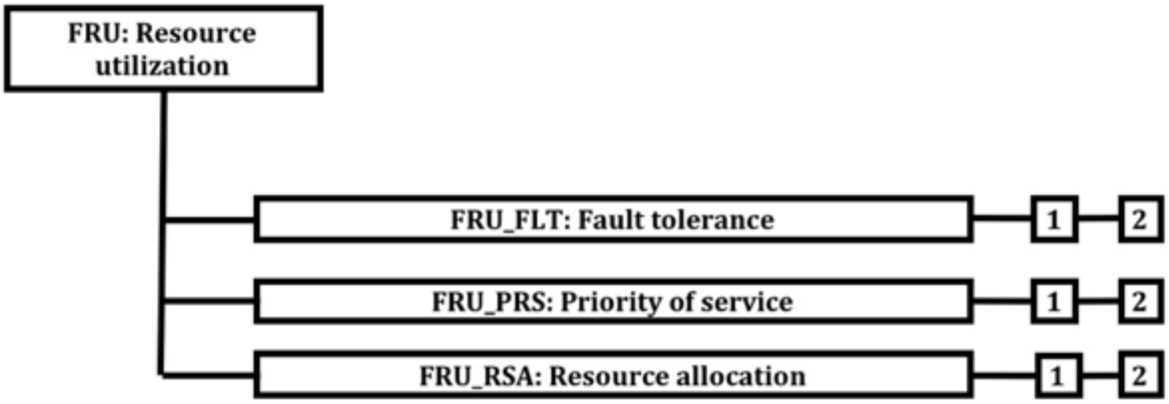
FRU: Resource utilization
FRU_FLT: Fault tolerance
1
2
FRU_PRS: Priority of service
1
2
FRU_RSA: Resource allocation
1
2

Figure 77 — FRU: Resource utilization class decomposition

15.2 Fault tolerance (FRU_FLT)

15.2.1 Family behaviour

The requirements of this family ensure that the TOE will maintain correct operation even in the event of failures.

15.2.2 Components leveling and description

Figure 78 shows the component leveling for this family.

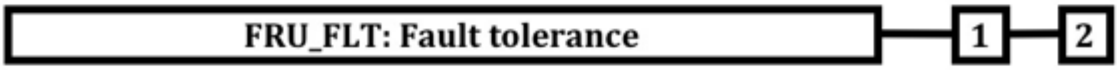

Figure 78 — FRU_FLT: Component leveling

FRU_FLT.1 Degraded fault tolerance, requires the TOE to continue correct operation of identified capabilities in the event of identified failures.

FRU_FLT.2 Limited fault tolerance, requires the TOE to continue correct operation of all capabilities in the event of identified failures.

15.2.3 Management of FRU_FLT.1, FRU_FLT.2

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

a) There are no management activities foreseen.

15.2.4 Audit of FRU_FLT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Any failure detected by the TSF.

b) Basic: All TOE capabilities being discontinued due to a failure.

15.2.5 Audit of FRU_FLT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Any failure detected by the TSF.

15.2.6 FRU_FLT.1 Degraded fault tolerance

Component relationships

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*].

15.2.7 FRU_FLT.2 Limited fault tolerance

Component relationships

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*].

15.3 Priority of service (FRU_PRS)

15.3.1 Family behaviour

The requirements of this family allow the TSF to control the use of resources under the control of the TSF by users and subjects such that high priority activities under the control of the TSF will always be accomplished without undue interference or delay caused by low priority activities.

15.3.2 Components leveling and description

Figure 79 shows the component leveling for this family.

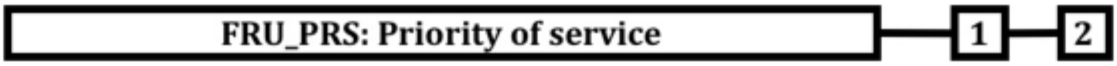

Figure 79 — FRU_PRS: Component leveling

FRU_PRS.1 Limited priority of service, provides priorities for a subject's use of a subset of the resources under the control of the TSF.

FRU_PRS.2 Full priority of service, provides priorities for a subject's use of all of the resources under the control of the TSF.

15.3.3 Management of FRU_PRS.1, FRU_PRS.2

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

a) Assignment of priorities to each subject in the TSF.

15.3.4 Audit of FRU_PRS.1, FRU_PRS.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

15.3.5 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.

15.3.6 FRU_PRS.2 Full priority of service

Component relationships

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.

15.4 Resource allocation (FRU_RSA)

15.4.1 Family behaviour

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 unauthorized monopolization of resources.

15.4.2 Components leveling and description

Figure 80 shows the component leveling for this family.

FRU_RSA: Resource allocation
1
2

Figure 80 — FRU_RSA: Component leveling

FRU_RSA.1 Maximum quotas, provides requirements for quota mechanisms that ensure that users and subjects will not monopolize a controlled resource.

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 monopolize a controlled resource.

15.4.3 Management of FRU_RSA.1

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.

15.4.4 Management of FRU_RSA.2

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.

15.4.5 Audit of FRU_RSA.1, FRU_RSA.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

15.4.6 FRU_RSA.1 Maximum quotas

Component relationships

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*].

15.4.7 FRU_RSA.2 Minimum and maximum quotas

Component relationships

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, subjects*] 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*].

16 Class FTA: TOE access

16.1 Class description

This family specifies functional requirements for controlling the establishment of a user's session.

Figure 81 shows the decomposition of this class, it's families and components. Elements are not shown in the figure.

Annex L provides explanatory information for this class and should be consulted when using the components identified in this class.

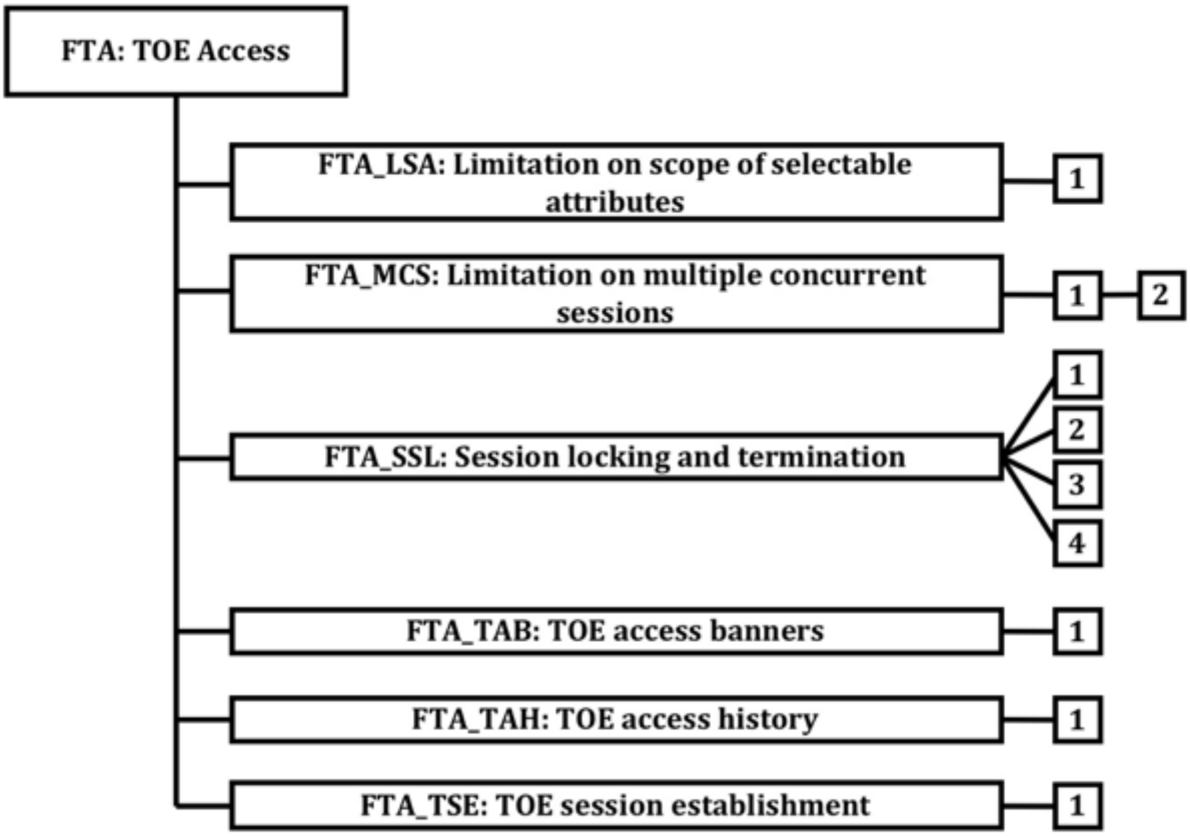
FTA: TOE Access
FTA_LSA: Limitation on scope of selectable attributes
1
FTA_MCS: Limitation on multiple concurrent sessions
1
2
FTA_SSL: Session locking and termination
1
2
3
4
FTA_TAB: TOE access banners
1
FTA_TAH: TOE access history
1
FTA_TSE: TOE session establishment
1

Figure 81 — FTA: TOE access class decomposition

16.2 Limitation on scope of selectable attributes (FTA_LSA)

16.2.1 Family behaviour

This family defines requirements to limit the scope of session security attributes that a user can select for a session.

16.2.2 Components leveling and description

Figure 82 shows the component leveling for this family.

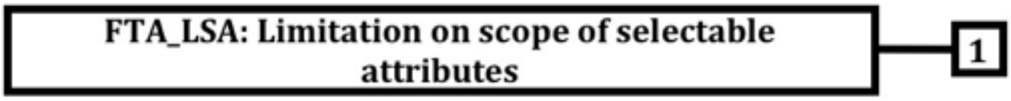

Figure 82 — FTA_LSA: Component leveling

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.

16.2.3 Management of FTA_LSA.1

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.

16.2.4 Audit of FTA_LSA.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: All failed attempts at selecting session security attributes.
b) Basic: All attempts at selecting session security attributes.
c) Detailed: Capture of the values of each of the session security attributes.

16.2.5 FTA_LSA.1 Limitation on scope of selectable attributes

Component relationships

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*].

16.3 Limitation on multiple concurrent sessions (FTA_MCS)

16.3.1 Family behaviour

This family defines requirements to place limits on the number of concurrent sessions that belong to the same user.

16.3.2 Components leveling and description

Figure 83 shows the component leveling for this family.

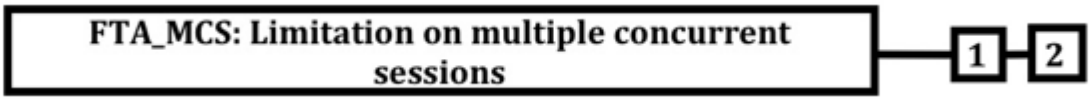

Figure 83 — FTA_MCS: Component leveling

FTA_MCS.1 Basic limitation on multiple concurrent sessions, provides limitations that apply to all users of the TSF.

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.

16.3.3 Management of FTA_MCS.1

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.

16.3.4 Management of FTA_MCS.2

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.

16.3.5 Audit of FTA_MCS.1, FTA_MCS.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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).

16.3.6 FTA_MCS.1 Basic limitation on multiple concurrent sessions

Component relationships

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.

16.3.7 FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions

Component relationships

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.

16.4 Session locking and termination (FTA_SSL)

16.4.1 Family behaviour

This family defines requirements for the TSF to provide the capability for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

16.4.2 Components leveling and description

Figure 84 shows the component leveling for this family.

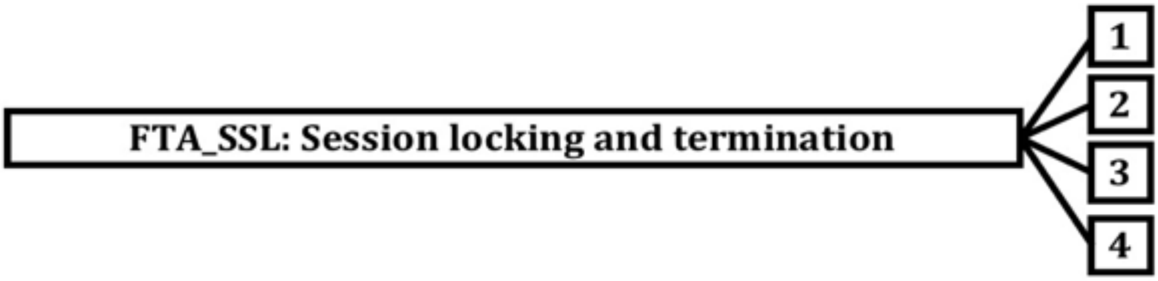

Figure 84 — FTA_SSL: Component leveling

FTA_SSL.1 TSF-initiated session locking includes system-initiated locking of an interactive session after a specified period of user inactivity.

FTA_SSL.2 User-initiated locking, provides capabilities for the user to lock and unlock the user's own interactive sessions.

FTA_SSL.3 TSF-initiated termination, provides requirements for the TSF to terminate the session after a specified period of user inactivity.

FTA_SSL.4 User-initiated termination, provides capabilities for the user to terminate the user's own interactive sessions.

16.4.3 Management of FTA_SSL.1

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 occur prior to unlocking the session.

16.4.4 Management of FTA_SSL.2

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

a) Management of the events that occur prior to unlocking the session.

16.4.5 Management of FTA_SSL.3

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.

16.4.6 Management of FTA_SSL.4

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

a) There are no management activities foreseen.

16.4.7 Audit of FTA_SSL.1, FTA_SSL.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

16.4.8 Audit of FTA_SSL.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Termination of an interactive session by the session locking mechanism.

16.4.9 Audit of FTA_SSL.4

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Termination of an interactive session by the user.

16.4.10 FTA_SSL.1 TSF-initiated session locking

Component relationships

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*].

16.4.11 FTA_SSL.2 User-initiated locking

Component relationships

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*].

16.4.12 FTA_SSL.3 TSF-initiated termination

Component relationships

Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles

FTA_SSL.3.1

The TSF shall terminate an interactive session after a [assignment: time interval of user inactivity].

16.4.13 FTA_SSL.4 User-initiated termination

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FTA_SSL.4.1

The TSF shall allow user-initiated termination of the user's own interactive session.

16.5 TOE access banners (FTA_TAB)

16.5.1 Family behaviour

This family defines requirements to display a configurable advisory warning message to users regarding the appropriate use of the TOE.

16.5.2 Components leveling and description

Figure 85 shows the component leveling for this family.

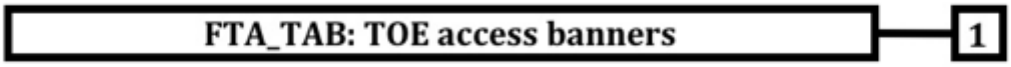

Figure 85 — FTA_TAB: Component leveling

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.

16.5.3 Management of FTA_TAB.1

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

a) Maintenance of the banner by the authorized administrator.

16.5.4 Audit of FTA_TAB.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

16.5.5 FTA_TAB.1 Default TOE access banners

Component relationships

Hierarchical to:	No other components.

Dependencies: No dependencies.

FTA_TAB.1.1

Before establishing a user session, the [selection: *TSF, TOE platform*] shall display an [assignment: *description of the message*] message.

16.6 TOE access history (FTA_TAH)

16.6.1 Family behaviour

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.

16.6.2 Components leveling and description

Figure 86 shows the component leveling for this family.

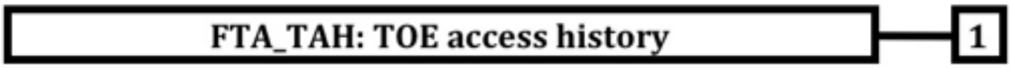

Figure 86 — FTA_TAH: Component leveling

FTA_TAH.1 TOE access history, provides the requirement for a TOE to display information related to previous attempts to establish a session.

16.6.3 Management of FTA_TAH.1

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

a) There are no management activities foreseen.

16.6.4 Audit of FTA_TAH.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) There are no auditable events foreseen.

16.6.5 FTA_TAH.1 TOE access history

Component relationships

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.

16.7 TOE session establishment (FTA_TSE)

16.7.1 Family behaviour

This family defines requirements to deny a user permission to establish a session with the TOE.

16.7.2 Components leveling and description

Figure 87 shows the component leveling for this family.

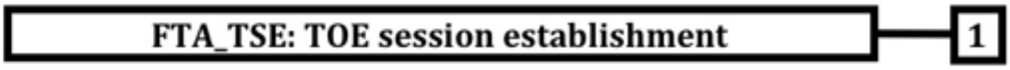

Figure 87 — FTA_TSE: Component leveling

FTA_TSE.1 TOE session establishment, provides requirements for denying users access to the TOE based on attributes.

16.7.3 Management of FTA_TSE.1

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

a) Management of the session establishment conditions by the authorized administrator.

16.7.4 Audit of FTA_TSE.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

16.7.5 FTA_TSE.1 TOE session establishment

Component relationships

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*].

17 Class FTP: Trusted path/channels

17.1 Class description

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 can 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).

In this paradigm, a trusted channel is a communication channel that **can** be initiated by either side of the channel and provides non-repudiation characteristics with respect to the identity of the sides of the channel.

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 **can** also be desired at other times during a user's session. Trusted path exchanges **can** 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.

Families describing the use of commonly used communication protocols used in the provision of trusted channels and paths are also given.

Figure 88 shows the decomposition of this class, it’s families and components. Elements are not shown in the figure.

Annex M provides explanatory information for this class and should be consulted when using the components identified in this class.

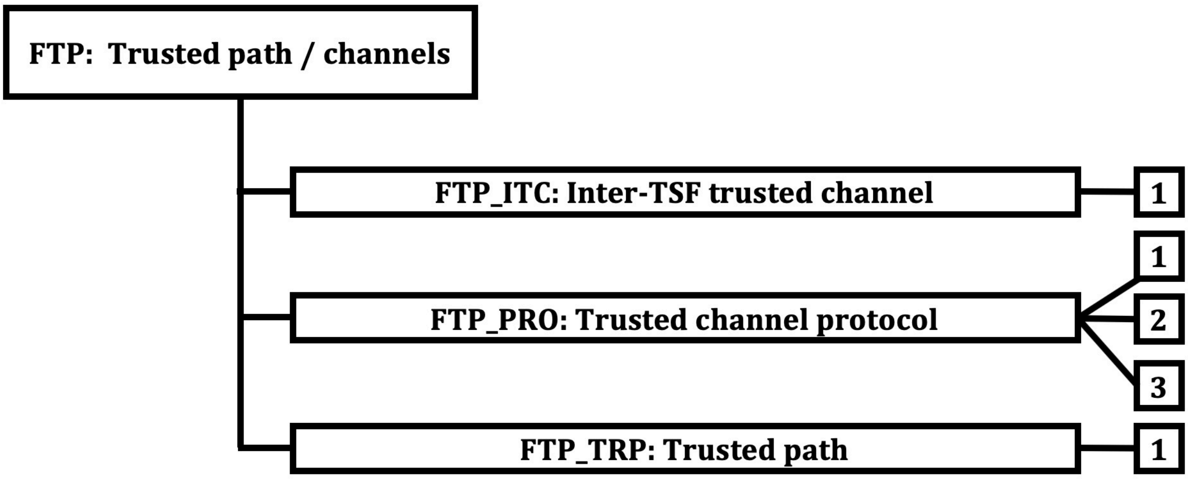
FTP: Trusted path / channels
FTP_ITC: Inter-TSF trusted channel
1
FTP_PRO: Trusted channel protocol
1
2
3
FTP_TRP: Trusted path
1

Figure 88 — FTP: Trusted path/channels class decomposition

17.2 Inter-TSF trusted channel (FTP_ITC)

17.2.1 Family behaviour

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. The components of this family may be included whenever there are requirements for the secure communication of user or TSF data between the TOE and other trusted IT products.

17.2.2 Components leveling and description

Figure 89 shows the component leveling for this family.

Figure 89 — FTP_ITC: Component leveling

FTP_ITC.1 Inter-TSF trusted channel, requires that the TSF provide a trusted communication channel between itself and another trusted IT product.

17.2.3 Management of FTP_ITC.1

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

a) Configuring the actions that require trusted channel, if supported.

17.2.4 Audit of FTP_ITC.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

17.2.5 FTP_ITC.1 Inter-TSF trusted channel

Component relationships

Hierarchical to:	No other components.
Dependencies:	No dependencies.

FTP_ITC.1.1

The TSF shall provide a communication channel between itself and another 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, another 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*].

17.3 Trusted channel protocol (FTP_PRO)

17.3.1 Family behavior

This family defines requirements for establishing a trusted channel and using the trusted channel to transfer the TSF data or user data securely.

17.3.2 Components leveling and description

Figure 90 shows the component leveling for this family.

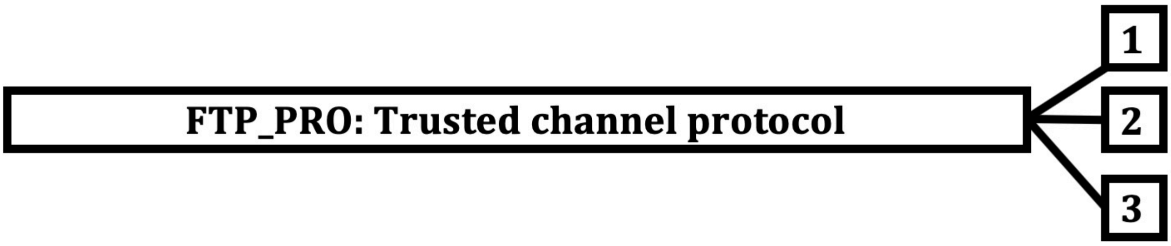

Figure 90 — FTP_PRO: Component leveling

FTP_PRO.1 Trusted channel protocol requires that communication be established in accordance with a defined protocol.

FTP_PRO.2 Trusted channel establishment requires that keys be securely established between the peers.

FTP_PRO.3 Trusted channel data protection requires that data in transit be protected.

17.3.3 Management of FTP_PRO.1

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

a) Configuring the protocols needed for the trusted channel,

b) Configuring the credentials for using the trusted channel,

c) Configuring the conditions for initializing and terminating the trusted channel.

17.3.4 Management of FTP_PRO.2

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

a) Configuring the parameters for shared secrets,

b) Configuring the parameters for cryptographic key derivation.

17.3.5 Management of FTP_PRO.3

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

a) Configuring the encryption and integrity mechanisms used by the trusted channel.

17.3.6 Audit of FTP_PRO.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or ST:

a) Minimal: Failure of the trusted channel establishment
b) Minimal: Identification of the initiator and target of failed trusted channel establishment
c) Basic: All attempted uses of the trusted channel
d) Basic: Identification of the initiator and target of all trusted channel attempts.

Other events should be considered according to the specific protocols used.

17.3.7 Audit of FTP_PRO.2

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

a) Minimal: Authentication failures during channel establishment
b) Basic: All authentication attempts.

17.3.8 Audit of FTP_PRO.3

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

a) Minimal: Failures when attempting to verify channel properties in FTP_PRO.3.2.

17.3.9 FTP_PRO.1 Trusted channel protocol

Component relationships

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.2 Trusted channel establishment
	FTP_PRO.3 Trusted channel data protection.

FTP_PRO.1.1

The TSF shall implement [assignment: *trusted channel protocol*] acting as [assignment: *defined protocol role(s)*] in accordance with: [assignment: *list of standards*].

FTP_PRO.1.2

The TSF shall enforce usage of the trusted channel for [assignment: *purpose(s) of the trusted channel*] in accordance with: [assignment: *list of standards*].

FTP_PRO.1.3

The TSF shall permit [selection: *itself, its peer*] to initiate communication via the trusted channel.

FTP_PRO.1.4

The TSF shall enforce the following rules for the trusted channel: [assignment: *rules governing operation and use of the trusted channel and/or its protocol*].

FTP_PRO.1.5

The TSF shall enforce the following static protocol options: [assignment: *list of options and references to standards in which each is defined*].

FTP_PRO.1.6

The TSF shall negotiate one of the following protocol configurations with its peer: [assignment: *list of configurations and reference to standards in which each is defined*].

17.3.10 FTP_PRO.2 Trusted channel establishment

Component relationships

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.1 Trusted channel protocol
	[FCS_CKM.1 Cryptographic key generation, or FCS_CKM.2 Cryptographic key distribution]
	FCS_CKM.5 Cryptographic key derivation
	FCS_COP.1 Cryptographic operation.

FTP_PRO.2.1

The TSF shall establish a shared secret with its peer using one of the following mechanisms: [assignment: *list of key establishment mechanisms*].

FTP_PRO.2.2

The TSF shall authenticate [selection: *its peer, itself to its peer*] using one of the following mechanisms: [assignment: *list of authentication mechanisms*] and according to the following rules: [assignment: *list of rules for carrying out the authentication*].

FTP_PRO.2.3

The TSF shall use [assignment: *key derivation function*] to derive the following cryptographic keys from a shared secret: [assignment: *list of cryptographic keys*].

17.3.11 FTP_PRO.3 Trusted channel data protection

Component relationships

Hierarchical to:	No other components.
Dependencies:	FTP_PRO.1 Trusted channel protocol
	FTP_PRO.2 Trusted channel establishment
	FCS_COP.1 Cryptographic operation.

FTP_PRO.3.1

The TSF shall protect data in transit from unauthorised disclosure using one of the following mechanisms: [assignment: *list of encryption mechanisms*].

FTP_PRO.3.2

The TSF shall protect data in transit from [selection: *modification, deletion, insertion, replay, [assignment: other]*] using one of the following mechanisms: [assignment: *list of integrity protection mechanisms*].

17.4 Trusted path (FTP_TRP)

17.4.1 Family behaviour

This family defines the requirements to establish and maintain trusted communication to or from users and the TSF. A trusted path **can** be required for any security-relevant interaction. Trusted path exchanges **can** be initiated by a user during an interaction with the TSF, or the TSF **can** establish communication with the user via a trusted path.

17.4.2 Components leveling and description

Figure 91 shows the component leveling for this family.

Figure 91 — FTP_TRP: Component leveling

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, PP-Module, functional package or ST author. The user and/or the TSF **can** have the ability to initiate the trusted path.

17.4.3 Management of FTP_TRP.1

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

a) Configuring the actions that require trusted path, if supported.

17.4.4 Audit of FTP_TRP.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP, PP-Module, functional package or 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.

17.4.5 FTP_TRP.1 Trusted path

Component relationships

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 [selection: *modification, disclosure, [assignment: other types of integrity or confidentiality violation]*].

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*]].

Annex A
(informative)
Security functional requirements structure of the application notes

A.1 General information

This annex contains additional guidance for the families and components defined in this document, 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 main clauses of this document.

A.2 Structure of the notes

A.2.1 General

The content and presentation of the notes related to functional requirements in this document will be defined below.

A.2.2 Class structure

A.2.2.1 General

Figure A.1 illustrates the functional class structure in this annex.

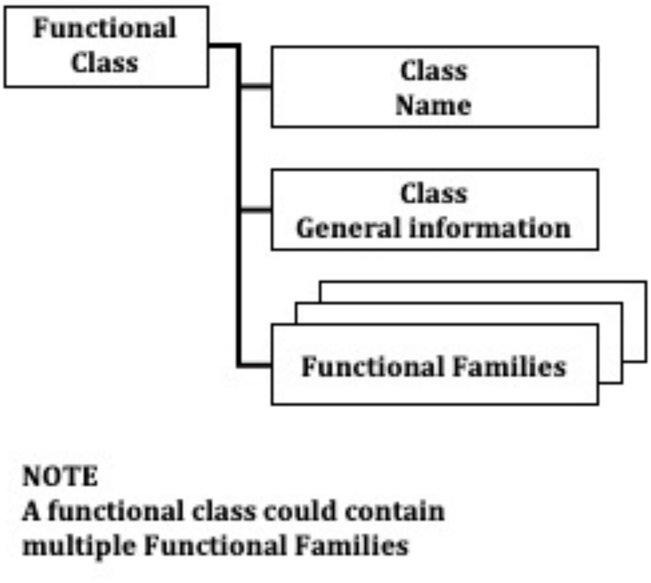
Functional Class
Class Name
Class General information
Functional Families
NOTE
A functional class could contain multiple Functional Families

Figure A.1 — Functional class structure

A.2.2.2 Class name

This is the unique name of the class defined within the normative elements of this document.

A.2.2.3 Class introduction

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 organization of each class with the families in each class and the hierarchical relationship between components in each family.

A.2.3 Family structure

A.2.3.1 General

Figure A.2 illustrates the functional family structure for application notes in diagrammatic form.

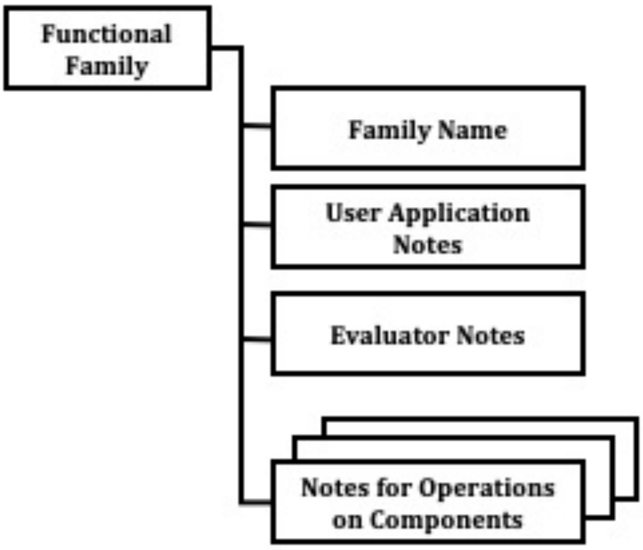

Figure A.2 — Functional family structure for application notes

A.2.3.2 Family name

This is the unique name of the family defined within the normative elements of this document.

A.2.3.3 User application notes

The user notes contain additional information that is of interest to potential users of the family, that is PP, PP-Module, 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.

NOTE In the annexes the term PP, PP-Module, functional package or ST author includes authors of documents used to formulate a PP or ST, this includes PP-Modules and functional packages.

A.2.3.4 Evaluator notes

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.

These User Notes and Evaluator Notes subclauses are not mandatory and appear only if appropriate.

A.2.4 Component structure

A.2.4.1 General

Figure A.3 illustrates the functional component structure for the application notes.

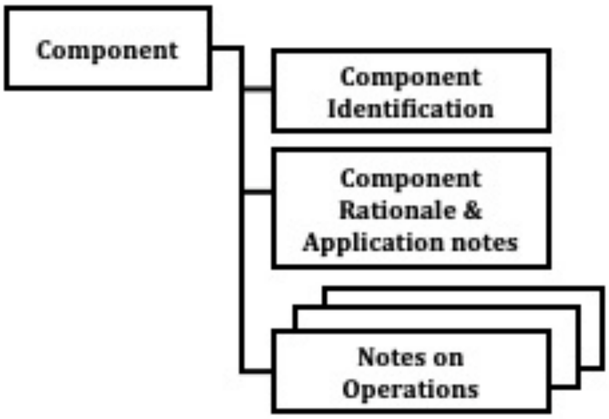

Figure A.3 — Functional component structure

A.2.4.2 Component identification

This is the unique name of the component defined within the normative elements of this document.

A.2.4.3 Component rationale and application notes

Any specific information related to the component is found in the component rationale and application notes subclause.

- The *rationale* contains the specifics of the rationale that refine the general statements on rationale for the specific level and is 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 may pertain to user notes, and/or evaluator notes as described in A.2.3. This refinement may be used to explain the nature of the dependencies.

The component rationale and application notes subclause is not mandatory and appears only if appropriate.

A.2.4.4 Notes on operations

This portion of each component contains advice relating to the permitted operations of the component.

The permitted operations subclause is not mandatory and appears only if appropriate.

Annex B
(informative)
Dependency tables for security functional components

B.1 Dependency tables

The tables from B.1 to B.11 show the hierarchical, direct, indirect, and optional dependencies among functional components.

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 indicates whether the column label component is a hierarchical requirement (indicated by an “H”). 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. Sets of optional requirements are indicated by using a subscript group, e.g. O^1 and O^2.

NOTE Depending upon the optional requirements chosen, some indirect dependencies are not applicable.

If no character is presented, the component is not dependent upon another component.

EXAMPLE

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.

Table B.1 — Dependency table for Class FAU: Security audit

	FAU_GEN.1	FAU_SAA.1	FAU_SAA.3	FAU_SAR.1	FAU_STG.1	FAU_STG.2	FAU_STG.4	FIA_UID.1	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FPT_STM.1	FTP_ITC.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			H										
FAU_SAR.1	X											-	
FAU_SAR.2	-			X								-	
FAU_SAR.3	-			X								-	
FAU_SEL.1	X							-	X	-	-	-	
FAU_STG.1	X											-	X
FAU_STG.2	X											-	
FAU_STG.3	X					H						-	
FAU_STG.4	-					X						-	
FAU_STG.5	X					X	H					-	

Table B.2 — Dependency table for Class FCO: Communication

	FIA_UID.1	FCO_NRR.1	FCO_NRO.1
FCO_NRO.1	X		
FCO_NRO.2	X		H
FCO_NRR.1	X		
FCO_NRR.2	X	H	

Table B.3 — Dependency table for Class FCS: Cryptographic support

	FCS_CKM.1	FCS_CKM.2	FCS_CKM.3	FCS_CKM.5	FCS_CGM.6	FCS_COP.1	FCS_RBG.1	FCS_RBG.2	FCS_RBG.3	FCS_RNG.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.3	FMT_SMF.1	FMT_SMR.1	FPT_FLS.1	FPT_TST.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1
FCS_CKM.1	-	O^1	X	O^1	X	O^1	O^2	-	-	O^2	-	-	-	-	-	-	-	-	-	-	-	-	-	-		-
FCS_CKM.2	O^1	-	X	O^1	-	-	-	-	-	-	-	-	-	-	O^1	O^1	-	-	-	-	-	-	-	-	-	-
FCS_CKM.3	O^1	-	-	O^1	-	-	-	-	-	-	-	-	-	-	O^1	O^1	-	-	-	-	-	-	-	-	-	-
FCS_CKM.5	-	O^1	-	-	X	O^1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
FCS_CKM.6	O^1	-	-	-	-	-	-	-	-	-	-	-	-	-	O^1	O^1	-	-	-	-	-	-	-	-	-	-
FCS_COP.1	O^2	-	X	O^2	-	-	-	-	-	-	-	-	-	-	O^1	O^1	-	-	-	-	-	-	-	-	-	-
FCS_RBG.1							-	O^1	O^1													X	X			
FCS_RBG.2							X	-	-													-	-			
FCS_RBG.3							X	-	-													-	-			
FCS_RBG.4							X	-	X													-	-			
FCS_RBG.5							X	O^1	O^1													-	-			
FCS_RBG.6							X	O^1	O^1													-	-			
FCS_RNG.1																										

Table B.4 — Dependency table for Class FDP: User data protection

	FCS_CKM.1	FCS_CKM.3	FCS_CKM.5	FCS_CKM.6	FCS_COP.1	FCS_RBG.1	FCS_RBG.2	FCS_RBG.3	FCS_RNG.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_IFF.3	FDP_IFF.4	FDP_ITC.1	FDP_ITC.2	FDP_ITT.1	FDP_ITT.2	FDP_ITT.3	FDP_RIP.1	FDP_ROL.1	FDP_SDI.1	FDP_UIT.1	FDP_UIT.2	FIA_UID.1	FMT_MSA.1	FMT_MSA.3	FMT_SMF.1	FMT_SMR.1	FPT_FLS.1	FPT_TST.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1
FDP_ACC.1										-	X	-	-													-	-	-	-	-					
FDP_ACC.2										H	X	-	-													-	-	-	-	-					
FDP_ACF.1										X	-	-	-													-	-	X	-	-					
FDP_DAU.1																																			
FDP_DAU.2										H																X									
FDP_ETC.1										O[1]	-	O[1]	-													-	-	-	-	-					
FDP_ETC.2										O[1]	-	O[1]	-													-	-	-	-	-					
FDP_IFC.1										-	-	-	X													-	-	-	-	-					
FDP_IFC.2										-	-	H	X													-	-	-	-	-					
FDP_IFF.1										-	-	X	-													-	-	X	-	-					
FDP_IFF.2										-	-	X	H													-	-	X	-	-					
FDP_IFF.3										-	-	X	-													-	-	-	-	-					
FDP_IFF.4										-	-	X	-	H												-	-	-	-	-					
FDP_IFF.5										-	-	X	-		H											-	-	-	-	-					
FDP_IFF.6										-	-	X	-													-	-	-	-	-					
FDP_IRC.1																																			

	FCS_CKM.1	FCS_CKM.3	FCS_CKM.5	FCS_CKM.6	FCS_COP.1	FCS_RBG.1	FCS_RBG.2	FCS_RBG.3	FCS_RNG.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_IFF.3	FDP_IFF.4	FDP_ITC.1	FDP_ITC.2	FDP_ITT.1	FDP_ITT.2	FDP_ITT.3	FDP_RIP.1	FDP_ROL.1	FDP_SDI.1	FDP_UIT.1	FDP_UIT.2	FIA_UID.1	FMT_MSA.1	FMT_MSA.3	FMT_SMF.1	FMT_SMR.1	FPT_FLS.1	FPT_TST.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1
FDP_ITC.1										O^1	-	O^1	-													-	-	X	-	-					
FDP_ITC.2										O^1	-	O^1	-													-	-	-	-	-			X	O^2	O^2
FDP_ITT.1										O^1	-	O^1	-													-	-	-	-	-					
FDP_ITT.2										O^1	-	O^1	-					H								-	-	-	-	-					
FDP_ITT.3										O^1	-	O^1	-					X								-	-	-	-	-					
FDP_ITT.4										O^1	-	O^1	-						X	H						-	-	-	-	-					
FDP_RIP.1																																			
FDP_RIP.2																					H														
FDP_ROL.1										O^1	-	O^1	-													-	-	-	-	-					
FDP_ROL.2										O^1	-	O^1	-									H				-	-	-	-	-					
FDP_SDC.1																																			
FDP_SDC.2	-	-	-	-	X	-	-	-	-	-	-	-	-			-	-								-	-	-	-	-	-	-	-	-	-	-
FDP_SDI.1																																			
FDP_SDI.2																							H												
FDP_UCT.1										O^2	-	O^2	-													-	-	-	-	-				O^1	O^1
FDP_UIT.1										O^2	-	O^2	-													-	-	-	-	-				O^1	O^1
FDP_UIT.2										O^1	-	O^1	-											O^2		-	-	-	-	-				O^2	-

	FCS_CKM.1	FCS_CKM.3	FCS_CKM.5	FCS_CKM.6	FCS_COP.1	FCS_RBG.1	FCS_RBG.2	FCS_RBG.3	FCS_RNG.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_IFF.3	FDP_IFF.4	FDP_ITC.1	FDP_ITC.2	FDP_ITT.1	FDP_ITT.2	FDP_ITT.3	FDP_RIP.1	FDP_ROL.1	FDP_SDI.1	FDP_UIT.1	FDP_UIT.2	FIA_UID.1	FMT_MSA.1	FMT_MSA.3	FMT_SMF.1	FMT_SMR.1	FPT_FLS.1	FPT_TST.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1
FDP_UIT.3										O^1	-	O^1	-											O^2	H	-	-	-	-	-				O^2	-

Table B.5 — Dependency table for Class FIA: Identification and authentication

	FIA_ATD.1	**FIA_UAU.1**	**FIA_UID.1**
FIA_AFL.1		X	-
FIA_API.1			
FIA_ATD.1			
FIA_SOS.1			
FIA_SOS.2			
FIA_UAU.1			X
FIA_UAU.2		H	X
FIA_UAU.3			
FIA_UAU.4			
FIA_UAU.5			
FIA_UAU.6			
FIA_UAU.7		X	-
FIA_UID.1			
FIA_UID.2			H
FIA_USB.1	X		

Table B.6 — Dependency table for Class FMT: Security management

	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FIA_UID.1	FMT_LIM.1	FMT_LIM.2	FMT_MSA.1	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FPT_STM.1
FMT_LIM.1						-	X						
FMT_LIM.2						X	-						
FMT_MOF.1					-						X	X	
FMT_MSA.1	O^1	-	O^1	-	-			-	-		X	X	
FMT_MSA.2	O^1	-	O^1	-	-			X	-		-	X	
FMT_MSA.3	-	-	-	-	-			X	-		-	X	
FMT_MSA.4	O^1	-	O^1	-	-			-	-		-	-	
FMT_MTD.1					-						X	X	
FMT_MTD.2					-					X	-	X	
FMT_MTD.3					-					X	-	-	
FMT_REV.1					-							X	
FMT_SAE.1					-							X	X
FMT_SMF.1													
FMT_SMR.1					X								
FMT_SMR.2					X							H	
FMT_SMR.3					-							X	

Table B.7 — Dependency table for Class FPR: Privacy

	FIA_UID.1	FPR_ANO.1	FPR_PSE.1	FPR_UNO.1
FPR_ANO.1				
FPR_ANO.2		H		
FPR_PSE.1				
FPR_PSE.2	X		H	
FPR_PSE.3			H	
FPR_UNL.1				
FPR_UNO.1				
FPR_UNO.2				H
FPR_UNO.3				X
FPR_UNO.4				

Table B.8 — Dependency table for Class FPT: Protection of the TSF

	AGD_OPE.1	ADV_FSP.1	FIA_UID.1	FMT_LIM.1	FMT_LIM.2	FMT_SMF.1	FMT_SMR.1	FPT_ITI.1	FPT_ITT.1	FTP_PHP.1	FPT_RCV.1	FPT_RCV.2	FPT_SSP.1	FPT_STM.1
FPT_EMS.1														
FPT_FLS.1														
FPT_INI.1														
FPT_ITA.1														
FPT_ITC.1														
FPT_ITI.1														
FPT_ITI.2								H						
FPT_ITT.1														
FPT_ITT.2									H					
FPT_ITT.3									X					
FPT_PHP.1														
FPT_PHP.2				X	-					H				
FPT_PHP.3														
FPT_RCV.1	X	-												
FPT_RCV.2	X	-									H			
FPT_RCV.3	X	-										H		
FPT_RCV.4														
FPT_RPL.1														
FPT_SSP.1									X					
FPT_SSP.2									X				H	
FPT_STM.1														
FPT_STM.2			-				X							X
FPT_TDC.1														
FPT_TEE.1														
FPT_TRC.1									X					
FPT_TST.1														

NOTE The AGD and ADV classes and their dependencies are described in ISO/IEC 15408-3

Table B.9 — Dependency table for Class FRU: Resource utilization

	FPT_FLS.1	FRU_FLT.1	FRU_PRS.1	FRU_RSA.1
FRU_FLT.1	X			
FRU_FLT.2	X	H		
FRU_PRS.1				
FRU_PRS.2			H	
FRU_RSA.1				
FRU_RSA.2				H

Table B.10 — Dependency table for Class FTA: TOE access

	FIA_UAU.1	FIA_UID.1	FMT_SMR.1	FTA_MCS.1
FTA_LSA.1				
FTA_MCS.1		X		
FTA_MCS.2		X		H
FTA_SSL.1	X	-		
FTA_SSL.2	X	-		
FTA_SSL.3			X	
FTA_SSL.4				
FTA_TAB.1				
FTA_TAH.1				
FTA_TSE.1				

Table B.11 — Dependency table for Class FTP: Trusted Path/channels

	FCS_CKM.1	FCS_CKM.2	FCS_CKM.3	FCS_CKM.5	FCS_CKM.6	FCS_COP.1	FCS_RBG.1	FCS_RBG.2	FCS_RBG.3	FCS_RNG.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.3	FMT_SMF.1	FMT_SMR.1	FPT_FLS.1	FPT_TST.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1	FTP_PRO.1	FTP_PRO.2	FTP_PRO.3
FTP_ITC.1																													
FTP_PRO.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		X	X
FTP_PRO.2	O[1]	O[1]	-	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X		
FTP_PRO.3	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	X	
FTP_TRP.1																													

Annex C
(normative)

Class FAU: Security audit - application notes

C.1 General information

C.1.1 General information about audit requirements

ISO/IEC 15408 audit families allow PP, PP-Module, functional package or /ST authors the ability to define requirements for monitoring user activities and, in some cases, detecting real, possible, or imminent violations of the enforcement of the SFRs. 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 records may be presented in human-readable format either directly or indirectly or both.

EXAMPLE 1

An example of direct presentation is storing the audit records in human-readable format

An example of indirect presentation is by using audit reduction tools.

While developing the security audit requirements, the PP, PP-Module, functional package or 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.

EXAMPLE 2

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.1.2 Audit requirements in a distributed environment

The implementation of audit requirements for networks and other large systems can 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 can be managed, due to lowered feasibility of interpreting (or even storing) what gets collected. The traditional notion of a time-ordered list, set of records or “trail” of audited events is not always applicable in a global asynchronous network with many arbitrary events occurring at once.

Also, different hosts and servers on a distributed TOE can have differing naming policies and values. Further, the use of symbolic names for audit review requires a net-wide convention to avoid redundancies and “name clashes.”

A multi-object audit repository, portions of which are accessible by a potentially wide variety of authorized users, are usually required if audit repositories are to serve a useful function in distributed systems.

Finally, misuse of authority by authorized users can be addressed by systematically avoiding local storage of audit data pertaining to administrator actions.

C.2 Security audit automatic response (FAU_ARP)

C.2.1 User application notes

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).

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.

An audit event is defined to be an "potential security violation" if so indicated by the Security audit analysis (FAU_SAA) components.

C.2.2 FAU_ARP.1 Security alarms

C.2.2.1 Component rationale and application notes

One or more actions should be taken for follow up action in the event of an alarm.

These actions could include informing the authorized user of the alarm, presenting the authorized user with a set of possible containment actions, or options for the authorized user to take corrective actions.

The timing of the actions should be carefully considered by the PP, PP-Module, functional package or /ST author.

C.2.2.2 Operations

In FAU_ARP.1.1, the PP, PP-Module, functional package or ST author specifies the actions to be taken in case of a potential security violation.

EXAMPLE

An example of such a list is: "inform the authorized user, disable the subject that created the potential security violation."

The list may also specify that the action to be taken can be specified by an authorized user.

C.3 Security audit data generation (FAU_GEN)

C.3.1 User application notes

The Security audit data generation family includes requirements to specify the audit events that shall be generated by the TSF for security-relevant events.

This family is presented in a manner that avoids a dependency on all components requiring audit support. Each component has an audit subclause developed in which the events to be audited for that functional area are listed. When the PP, PP-Module, functional package or /ST is written, the items in the audit area are used to complete the variable in these components. Thus, the specification of what could be audited for a functional area is localized in that functional area.

The list of auditable events is entirely dependent on the other functional families within the PP, PP-Module, functional package or ST./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, PP-Module, functional package or ST author with information necessary to ensure that all appropriate auditable events are specified in the PP, PP-Module, functional package or ST./ST. The following example shows how auditable events are to be specified in appropriate functional families:

EXAMPLE 1

"The following actions should be auditable if Security audit data generation (FAU_GEN) is included in the PP, PP-Module, functional package or 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, the new values of the attributes should be captured."

NOTE Sensitive attribute data items include passwords and cryptographic keys.

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. So, 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.

Observe that the categorization of auditable events (minimal, basic, detailed) is hierarchical in that order.

This means that

- When Minimal Audit Generation is desired, all auditable events identified as being Minimal should be included in the PP, PP-Module, functional package or ST through the use of the appropriate assignment operation.
- When Basic Audit Generation is desired, all auditable events identified as being either Minimal or Basic, should also be included in the PP, PP-Module, functional package or 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, PP-Module, functional package or ST.

A PP, PP-Module, functional package or ST author may decide to include other auditable events beyond those required for a given audit level.

EXAMPLE 2

For example, the PP, PP-Module, functional package or ST may claim only minimal audit capabilities while including most of the basic capabilities because the few excluded capabilities conflict with other PP, PP-Module, functional package or ST constraints (perhaps because they require the collection of unavailable data).

The functionality that creates the auditable event should be specified in the PP or ST as a functional requirement.

EXAMPLE 3

The following are examples of the types of the events that can be defined as auditable within each PP, PP-Module, functional package or ST functional component:

a) Introduction of objects within the control of the TSF 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 authorized user (such as. suppression of a TSF protection mechanism as human-readable labels);

i) Import/export of data from/to removable media (such as printed output, tapes, USB sticks).

C.3.1.1 Evaluator notes

FAU_GEN.1.1 has a dependency on FPT_STM.1 Reliable time stamps. If correctness of time is not an issue for this TOE, elimination of this dependency could be justified by the PP, PP-Module, functional package or ST author.

C.3.2 FAU_GEN.1 Audit data generation

C.3.2.1 Component rationale and application notes

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.

FAU_GEN.1 Audit data generation by itself might be used when the SFRs do not require that individual user identities be associated with audit events. This could be appropriate when the PP, PP-Module, functional package or ST also contains privacy requirements. If the user identity must be incorporated FAU_GEN.2 User identity association could be used in addition to FAU_GEN.1.

If the subject is a user, the user identity may be recorded as the subject identity. The identity of the user may not yet have been verified if User authentication (FIA_UAU) has not been applied. Therefore, in the instance of an invalid login the claimed user identity should be recorded. It should also be considered whether to indicate when a recorded identity has not been authenticated.

C.3.2.2 Operations

In FAU_GEN.1.1, the PP, PP-Module, functional package or ST author should select the level of auditable events called out in the audit subclause of other functional components included in the PP, PP-Module, functional package or ST. This level is one of the following: “minimum”, “basic”, “detailed” or “not specified”.

In FAU_GEN.1.1, the PP, PP-Module, functional package or 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).

In FAU_GEN.1.2, the PP, PP-Module, functional package or ST author should assign, for each of the auditable events included in the PP, PP-Module, functional package or ST, either a list of other audit relevant information to be included in audit events records or none.

C.3.3 FAU_GEN.2 User identity association

C.3.3.1 Component rationale and application notes

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.

There is a potential conflict between the audit and privacy requirements. For audit purposes, it may be desirable to know who performed an action. A user may want to keep his/her actions to himself/herself and not be identified by other persons such as a site with job offers. Or it might be required in the Organizational Security Policy that the identity of the users must be protected. In those cases, the objectives for audit and privacy could contradict each other. Therefore, if this requirement is selected and privacy is important, inclusion of the component user pseudonymity might be considered. Requirements on determining the real user name based on its pseudonym are specified in the privacy class.

If the identity of the user has not yet been verified through authentication, in the instance of an invalid login the claimed user identity should be recorded. It should be considered to indicate when a recorded identity has not been authenticated.

C.4 Security audit analysis (FAU_SAA)

C.4.1 User application notes

This family defines requirements for automated means that analyze 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 a potential security violation.

The action to be performed by the TSF on detection of a potential violation is defined in Security audit automatic response (FAU_ARP) components.

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 authorized users for review.

C.4.2 FAU_SAA.1 Potential violation analysis

C.4.2.1 Component rationale and application notes

This component is used to specify the set of auditable events whose occurrence or accumulated occurrence held to indicate a potential violation of the enforcement of the SFRs, and any rules to be used to perform the violation analysis.

C.4.2.2 Operations

In FAU_SAA.1.2, the PP, PP-Module, functional package or 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 enforcement of the SFRs.

In FAU_SAA.1.2, the PP, PP-Module, functional package or 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. 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”.

EXAMPLE

Period of time: period of the day, duration

C.4.3 FAU_SAA.2 Profile based anomaly detection

C.4.3.1 Component rationale and application notes

A *profile* is a structure that characterizes 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. The ways in which the various types of activity are recorded in the profile are referred to as *profile metrics*.

EXAMPLE

Patterns of usage: patterns in exceptions raised, patterns in resource utilization (when, which, how), patterns in actions performed.

Profile metrics: resource measures, event counters, timers

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.

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.

The sophistication of the anomaly detection tool will largely be determined by the number of target profile groups required by the PP, PP-Module, functional package or ST and the complexity of the required profile metrics.

The PP, PP-Module, functional package or ST author should enumerate specifically what activity should be monitored and/or analysed by the TSF. The PP, PP-Module, functional package or ST author should also identify specifically what information pertaining to the activity is necessary to construct the usage profiles.

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, PP-Module, functional package or ST author.

EXAMPLE 2

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.

Administrative notification should be provided such that the authorized user understands the significance of the suspicion rating.

The PP, PP-Module, functional package or 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.

C.4.3.2 Operations

In FAU_SAA.2.1, the PP, PP-Module, functional package or ST author should specify the profile target group. A single PP, PP-Module, functional package or ST may include multiple profile target groups.

In FAU_SAA.2.3, the PP, PP-Module, functional package or 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.

C.4.4 FAU_SAA.3 Simple attack heuristics

C.4.4.1 Component rationale and application notes

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 1

Example of such events include the deletion of a key TSF security data file (such as 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.

The complexity of a given tool will depend greatly on the assignments defined by the PP, PP-Module, functional package or ST author in identifying the base set of *signature events*.

The PP, PP-Module, functional package or ST author should enumerate specifically what events should be monitored by the TSF in order to perform the analysis. The PP, PP-Module, functional package or ST author should identify specifically what information pertaining to the event is necessary to determine if the event maps to a signature event.

Administrative notification should be provided such that the authorized user understands the significance of the event and the appropriate possible responses.

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.

EXAMPLE 2

Examples of other input data include network datagrams, resource/accounting data, or combinations of various system data.

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 analyzed.

C.4.4.2 Operations

In FAU_SAA.3.1, the PP, PP-Module, functional package or 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 enforcement of the SFRs. These include events that by themselves indicate a clear violation to the enforcement of the SFRs, or whose occurrence is so significant that they warrant actions.

In FAU_SAA.3.2, the PP, PP-Module, functional package or 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, PP-Module, functional package or ST author should define precisely what system events and event attributes are being monitored within the input data.

C.4.5 FAU_SAA.4 Complex attack heuristics

C.4.5.1 Component rationale and application notes

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 1

Example of such events include the deletion of a key TSF security data file (such as 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.

The complexity of a given tool will depend greatly on the assignments defined by the PP, PP-Module, functional package or ST author in identifying the base set of signature events and event sequences.

The PP, PP-Module, functional package or ST author should enumerate specifically what events should be monitored by the TSF in order to perform the analysis. The PP, PP-Module, functional package or ST author should identify specifically what information pertaining to the event is necessary to determine if the event maps to a signature event.

Administrative notification should be provided such that the authorized user understands the significance of the event and the appropriate possible responses.

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.

EXAMPLE 2

Examples of other input data include network datagrams, resource/accounting data, or combinations of various system data.

Levelling, therefore, requires the PP, PP-Module, functional package or ST author to specify the type of input data used to monitor system activity.

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 analyzed.

C.4.5.2 Operations

In FAU_SAA.4.1, the PP, PP-Module, functional package or ST author should identify a base set of lists 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.

In FAU_SAA.4.1, the PP, PP-Module, functional package or 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 enforcement of the SFRs. These include events that by themselves indicate a clear violation to the SFRs, or whose occurrence is so significant they warrant action.

In FAU_SAA.4.2, the PP, PP-Module, functional package or 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, PP-Module, functional package or 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)

C.5.1 User application notes

The Security audit review family defines requirements related to review of the audit information.

These functions should allow pre-storage or post-storage audit selection.

EXAMPLE

An example of requirement related to review of the audit information is the ability to selectively review:

- the actions of one or more users (such as. 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 SFR attribute

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 select subsets of audit data based on a single criterion or multiple criteria with logical (i.e. and/or) relations and order the audit data before it is reviewed.

C.5.2 FAU_SAR.1 Audit review

C.5.2.1 Component rationale and application notes

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

This component is also used to specify that users and/or authorized 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.

The content of the audit records that can be viewed can be specified.

C.5.2.2 Operations

In FAU_SAR.1.1, the PP, PP-Module, functional package or ST author should specify the authorized users that can use this capability. If appropriate the PP, PP-Module, functional package or ST author may include security roles (see FMT_SMR.1 Security roles).

In FAU_SAR.1.1, the PP, PP-Module, functional package or ST author should specify the type of information the specified user is permitted to obtain from the audit records.

EXAMPLE

Examples are “all”, “subject identity”, “all information belonging to audit records referencing this user”.

When employing the SFR, FAU_SAR.1, it is not necessary to repeat, in full detail, the list of audit information first specified in FAU_GEN.1. Use of terms such as “all” or “all audit information” assist in eliminating ambiguity and the further need for comparative analysis between the two security requirements.

C.5.3 FAU_SAR.2 Restricted audit review

C.5.3.1 Component rationale and application notes

This component specifies that any users not identified in FAU_SAR.1 Audit review will not be able to read the audit records.

C.5.4 FAU_SAR.3 Selectable audit review

C.5.4.1 Component rationale and application notes

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

EXAMPLE

Means of manipulating audit data include sorting and filtering.

C.5.4.2 Operations

In FAU_SAR.3.1, the PP, PP-Module, functional package or ST author should specify whether capabilities to select and/or order audit data is required from the TSF.

In FAU_SAR.3.1, the PP, PP-Module, functional package or 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.

EXAMPLE

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)

C.6.1 User application notes

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.

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.

C.6.2 FAU_SEL.1 Selective audit

C.6.2.1 Component rationale and application notes

This component defines the selection criteria used, and the resulting audited subsets of the set of all auditable events, based on user attributes, subject attributes, object attributes, or event types.

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.

For a distributed environment, the host identity could be used as a selection criterion for events to be audited.

The management function FMT_MTD.1 Management of TSF data will handle the rights of authorized users to query or modify the selections.

C.6.2.2 Operations

In FAU_SEL.1.1, the PP, PP-Module, functional package or 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.

In FAU_SEL.1.1, the PP, PP-Module, functional package or 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 data storage (FAU_STG)

C.7.1 User application notes

The Security audit data storage family describes requirements for storing audit data for later use, including requirements controlling the loss of audit information due to TOE failure, attack and/or exhaustion of storage space.

C.7.2 FAU_STG.1 Audit data storage location

C.7.2.1 Component rationale and application notes

In a distributed environment, as the location of the audit trail is in the TSF, but not necessarily co-located with the function generating the audit data, the PP, PP-Module, functional package or ST author could request authentication of the originator of the audit record, or non-repudiation of the origin of the record prior to storing this record in the audit trail.

The TSF will protect the stored audit records in the audit trail from unauthorised deletion and modification. It is noted that in some TOEs the auditor (role) might not be authorized to delete the audit records for a certain period of time.

FAU_STG.1.1 is dependent upon FTP_ITC.1 Inter-TSF trusted channel, if “transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC” is not selected then the PP, PP-Module, functional package or ST author can satisfy the dependency by providing the rationale explaining why it was not selected.

C.7.2.2 Operations

In FAU_STG.1.1the PP, PP-Module, functional package or ST author should select where the audit data is stored. Audit data may be stored on the TOE itself, be transmitted to an external entity using a trusted channel, or other storage options can be specified in the assignment.

If additional or alternative storage locations for audit data need to be specified by the PP, PP-Module, functional package or ST author then this requirement can be specified in FAU_STG.1.1 using the assignment found within the selection.

C.7.3 FAU_STG.2 Protected audit data storage

C.7.3.1 Component rationale and application notes

In a distributed environment, as the location of the audit trail is in the TSF, but not necessarily co-located with the function generating the audit data, the PP, PP-Module, functional package or 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.

The TSF will protect the stored audit data in the audit trail from unauthorized deletion and modification. It is noted that in some TOEs the auditor (role) might not be authorized to delete the audit records for a certain period of time.

C.7.3.2 Operations

In FAU_STG.2.2, the PP, PP-Module, functional package or ST author should specify whether the TSF shall prevent or only be able to detect modifications of the stored audit data in the audit trail. Only one of these options may be chosen.

C.7.4 FAU_STG.3 Guarantees of audit data availability

C.7.4.1 Component rationale and application notes

This component allows the PP, PP-Module, functional package or ST author to specify to which metrics the audit trail should conform.

In a distributed environment, as the location of the audit trail is in the TSF, but not necessarily co-located with the function generating the audit data, the PP, PP-Module, functional package or 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.

C.7.4.2 Operations

In FAU_STG.3.3, PP, PP-Module, functional package or ST author should specify the metric that the TSF must ensure with respect to the stored audit records. 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.

EXAMPLE

An example of the metric could be “100,000” indicating that 100,000 audit records can be stored.

In FAU_STG.3.3 the PP, PP-Module, functional package or 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.

C.7.5 FAU_STG.4 Prevention of audit data loss

C.7.5.1 Component rationale and application notes

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 audited events can take place. The requirement also states that no matter how the requirement is instantiated, the authorized user with specific rights to this effect, can continue to generate audited events (actions). The reason is that otherwise the authorized user could not even reset the TOE. 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.

C.7.5.2 Operations

In FAU_STG.5.1, the PP, PP-Module, functional package or ST author should select whether the TSF shall ignore audited actions, or whether it should prevent audited 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.

In FAU_STG.5.1, the PP, PP-Module, functional package or ST author should specify other actions that should be taken in case of audit storage failure, such as informing the authorized user. If there is no other action to be taken in case of audit storage failure, this assignment can be completed with “none”.

C.7.6 FAU_STG.5 Action in case of possible audit data loss

C.7.6.1 Component rationale and application notes

This component requires that actions will be taken when the audit trail exceeds certain pre-defined limits.

C.7.6.2 Operations

In FAU_STG.5 Prevention of audit data loss, the PP, PP-Module, functional package or ST author should indicate the pre-defined limit. If the management functions indicate that this number might be changed by the authorized user, this value is the default value. The PP, PP-Module, functional package or ST author might choose to let the authorized user define this limit.

EXAMPLE

In the case that an authorized user defines the limit, an example of the assignment can be “an authorized user set limit”.

In FAU_STG.5 Prevention of audit data loss,, the PP, PP-Module, functional package or 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 authorized user.

Annex D
(normative)

Class FCO: Communication- application notes

D.1 General information

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.

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.

In the literature, the terms “proof of receipt” and “proof of origin” are commonly used terms. However, it is recognized 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.

D.2 Non-repudiation of origin (FCO_NRO)

D.2.1 User application notes

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 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.

EXAMPLE 1

Evidence of origin could be a digital signature

If the information or the associated attributes are altered in any way, validation of the evidence of origin might fail. Therefore, a PP, PP-Module, functional package or ST author should consider including integrity requirements such as FDP_UIT.1 Data exchange integrity in the PP, PP-Module, functional package or ST.

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. The third role is a subject that requests verification of the evidence of origin.

EXAMPLE 2

Subject that requests evidence of origin: a recipient or a third party such as an arbiter.

Subject to which the evidence is provided: A notary

The PP, PP-Module, functional package or ST author must specify the conditions that must be met to be able to verify the validity of the evidence.

EXAMPLE 3

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.

In most cases, the identity of the recipient will be the identity of the user who received the transmission. In some instances, the PP, PP-Module, functional package or ST author does not want the user identity to be exported. In that case, the PP, PP-Module, functional package or 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.

In addition to (or instead of) the user identity, a PP, PP-Module, functional package or ST author might be more concerned about the time the information was transmitted.

EXAMPLE 4

For example, requests for proposals must be transmitted before a certain date in order to be considered.

In such instances, these requirements can be customized to provide a timestamp indication (time of origin).

D.2.2 FCO_NRO.1 Selective proof of origin

D.2.2.1 Operations

In FCO_NRO.1.1, the PP, PP-Module, functional package or ST author should fill in the types of information subject to the evidence of origin function.

EXAMPLE

An example of the type of information is "electronic mail messages"

In FCO_NRO.1.1, the PP, PP-Module, functional package or ST author should specify the user/subject who can request evidence of origin.

In FCO_NRO.1.1, the PP, PP-Module, functional package or ST author, dependent on the selection, should specify the third parties that can request evidence of origin.

EXAMPLE 1

A third party could be an arbiter, judge, or legal body.

In FCO_NRO.1.2, the PP, PP-Module, functional package or ST author should fill in the list of the attributes that shall be linked to the information;

EXAMPLE 2

Attributes include originator identity, time of origin, and location of origin.

In FCO_NRO.1.2, the PP, PP-Module, functional package or 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.

In FCO_NRO.1.3, the PP, PP-Module, functional package or ST author should specify the user/subject who can verify the evidence of origin.

In FCO_NRO.1.3, the PP, PP-Module, functional package or ST author should fill in the list of limitations under which the evidence can be verified.

EXAMPLE

An example of a limitation is "the evidence can only be verified within a 24-hour time interval."

An assignment of "immediate" or "indefinite" is acceptable.

In FCO_NRO.1.3, the PP, PP-Module, functional package or ST author, dependent on the selection, should specify the third parties that can verify the evidence of origin.

D.2.3 FCO_NRO.2 Enforced proof of origin

D.2.3.1 Operations

In FCO_NRO.2.1, the PP, PP-Module, functional package or ST author should fill in the types of information subject to the evidence of origin function.

EXAMPLE

electronic mail messages.

In FCO_NRO.2.2, the PP, PP-Module, functional package or 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.

In FCO_NRO.2.2, the PP, PP-Module, functional package or 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.

In FCO_NRO.2.3, the PP, PP-Module, functional package or ST author should specify the user/subject who can verify the evidence of origin.

In FCO_NRO.2.3, the PP, PP-Module, functional package or ST author should fill in the list of limitations under which the evidence can be verified.

EXAMPLE

For example, the evidence can only be verified within a 24-hour time interval.

An assignment of "immediate" or "indefinite" is acceptable.

In FCO_NRO.2.3, the PP, PP-Module, functional package or ST author, dependent on the selection, should specify the third parties that can verify the evidence of origin.

EXAMPLE 2

A third party could be an arbiter, judge, or legal body.

D.3 Non-repudiation of receipt (FCO_NRR)

D.3.1 User application notes

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 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.

EXAMPLE 1

An example of a receipt is a digital signature

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.

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, PP-Module, functional package or ST author should consider including integrity requirements such as FDP_UIT.1 Data exchange integrity in the PP, PP-Module, functional package or ST.

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). 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.

EXAMPLE 2

A recipient or subject could be a notary.

The PP, PP-Module, functional package or 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.

In most cases, the identity of the recipient will be the identity of the user who received the transmission. In some instances, the PP, PP-Module, functional package or ST author does not want the user identity to be exported. In that case, the PP, PP-Module, functional package or 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.

In addition to (or instead of) the user identity, a PP, PP-Module, functional package or ST author might be more concerned about the time the information was received.

EXAMPLE 3

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 customized to provide a timestamp indication (time of receipt).

D.3.2 FCO_NRR.1 Selective proof of receipt

D.3.2.1 Operations

In FCO_NRR.1.1, the PP, PP-Module, functional package or ST author should fill in the types of information subject to the evidence of receipt function, for example, electronic mail messages.

In FCO_NRR.1.1, the PP, PP-Module, functional package or ST author should specify the user/subject who can request evidence of receipt.

In FCO_NRR.1.1, the PP, PP-Module, functional package or ST author, dependent on the selection, should specify the third parties that can request evidence of receipt.

EXAMPLE

A third party could be an arbiter, judge, or legal body.

In FCO_NRR.1.2, the PP, PP-Module, functional package or 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.

In FCO_NRR.1.2, the PP, PP-Module, functional package or 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.

In FCO_NRR.1.3, the PP, PP-Module, functional package or ST author should specify the user/subjects who can verify the evidence of receipt.

In FCO_NRR.1.3, the PP, PP-Module, functional package or 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.

In FCO_NRR.1.3, the PP, PP-Module, functional package or ST author, dependent on the selection, should specify the third parties that can verify the evidence of receipt.

D.3.3 FCO_NRR.2 Enforced proof of receipt

D.3.3.1 Operations

In FCO_NRR.2.1, the PP, PP-Module, functional package or ST author should fill in the types of information subject to the evidence of receipt function,

EXAMPLE 1

For example, electronic mail messages.

In FCO_NRR.2.2, the PP, PP-Module, functional package or ST author should fill in the list of the attributes that shall be linked to the information;

EXAMPLE 2

For example, recipient identity, time of receipt, and location of receipt.

In FCO_NRR.2.2, the PP, PP-Module, functional package or 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.

In FCO_NRR.2.3, the PP, PP-Module, functional package or ST author should specify the user/subjects who can verify the evidence of receipt.

In FCO_NRR.2.3, the PP, PP-Module, functional package or ST author should fill in the list of limitations under which the evidence can be verified. An assignment of “immediate” or “indefinite” is acceptable.

EXAMPLE

For example, the evidence can only be verified within a 24-hour time interval.

In FCO_NRR.2.3, the PP, PP-Module, functional package or 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.

Annex E
(normative)

Class FCS: Cryptographic support- application notes

E.1 General information

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.

The FCS: Cryptographic support class is composed of four families: Cryptographic key management (FCS_CKM), Cryptographic operation (FCS_COP), Random bit generation (FCS_RBG), and Generation of random numbers (FCS_RNG).

The Cryptographic key management (FCS_CKM) family addresses the management aspects of cryptographic keys; the Cryptographic operation (FCS_COP) family is concerned with the operational use of those cryptographic keys; the Random bit generation (FCS_RBG) family provides requirements for generating random bits; and the Generation of random numbers (FCS_RNG) is concerned with ensuring that random numbers meet defined quality metrics.

For each cryptographic key generation method implemented by the TOE, if any, the PP, PP-Module, functional package or ST author should select the FCS_CKM.1 Cryptographic key generation component.

For each cryptographic key distribution method implemented by the TOE, if any, the PP, PP-Module, functional package or ST author should select the FCS_CKM.2 Cryptographic key distribution.

For each cryptographic key access method implemented by the TOE, if any, the PP, PP-Module, functional package or ST author should select the FCS_CKM.3 Cryptographic key access.

For each cryptographic key derivation method implemented by the TOE, if any, the PP, PP-Module, functional package or ST author should select the FCS_CKM.5 Cryptographic key derivation.

For each cryptographic key destruction method implemented by the TOE, if any, the PP, PP-Module, functional package or ST author should select the FCS_CKM.6 Timing and event of cryptographic key destruction component.

For each cryptographic operation (such as digital signature, data encryption, key agreement, secure hash, etc.) performed by the TOE, if any, the PP, PP-Module, functional package or ST author should select the FCS_COP.1 Cryptographic operation component.

For each deterministic random bit generation service implemented by the TOE, if any, the PP, PP-Module, functional package or ST author should select the FCS_RBG.1 Random bit generation (RBG) component.

For each external seeding source used by the TOE, if any, the PP, PP-Module, functional package or ST author should select the FCS_RBG.2 Random bit generation (external seeding)component.

For each internal seeding source (single) used by the TOE, if any, the PP, PP-Module, functional package or ST author should select the FCS_RBG.3 Random bit generation (internal seeding – single source) component.

Where internal seeding source (multiple) is to be specified, the PP, PP-Module, functional package or ST author should select the FCS_RBG.4 Random bit generation (internal seeding – multiple sources) component.

For cases where the TOE combines entropy sources, the FCS_RBG.5 Random bit generation (combining entropy sources) component should be specified by PP, PP-Module, functional package or ST author.

For each random bit generation service implemented by the TOE, the PP, PP-Module, functional package or ST author should specify the FCS_RBG.6 Random bit generation service component.

For each random number generation service implemented by the TOE, the PP, PP-Module, functional package or ST author should specify the FCS_RNG.1 Random number generation component.

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.

E.2 Cryptographic key management (FCS_CKM)

E.2.1 User application notes

Cryptographic keys must be managed throughout their lifetime. The typical events in the lifecycle of a cryptographic key include but are not limited to: key generation or derivation, distribution, entry, storage, access, and destruction.

EXAMPLE 1

- backup
- escrow
- archive
- recovery

The inclusion of other stages is dependent on the key management strategy being implemented, as the TOE is not always involved in all of the key life-cycle phases.

EXAMPLE 2

The TOE may only generate and distribute cryptographic keys.

This family is intended to support the cryptographic key lifecycle and consequently defines requirements for the following activities: cryptographic key generation, cryptographic key derivation, 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.

If Security audit data generation (FAU_GEN) is included in the PP, PP-Module, functional package or 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).

Typically, random numbers are used to generate cryptographic keys. If this is the case, then FCS_CKM.1 Cryptographic key generation should be used instead of the component FIA_SOS.2 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 should be used.

E.2.2 Evaluator notes

Evaluators should refer to ISO/IEC 15408-1:20XX, Annex B.4 for information in regard to the use of standards specified in FCS_CKM.5.

FCS_CKM.5 has a dependency on FCS_CKM.6, The dependency should be understood as the dependency of two directions, 1) destruction of key derivation key, and 2) destruction of derived keys. Evaluators should keep in mind that the dependency of two directions has to be fulfilled, and should also consider any intermediate values (such as those from key establishment) that should be destroyed in order to preserve the confidentiality of the key.

E.2.3 FCS_CKM.1 Cryptographic key generation

E.2.3.1 Component rationale and application notes

This component requires the cryptographic key sizes and method used to generate cryptographic keys to be specified, this may be in accordance with an assigned standard. It should be used to specify the cryptographic key sizes and the method 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 may be common or different for the various entities and may be either the input to or the output from the method.

EXAMPLE

An example of a method is an algorithm.

E.2.3.2 Operations

In FCS_CKM.1.1, the PP, PP-Module, functional package or ST author should specify the cryptographic key generation algorithm to be used.

In FCS_CKM.1.1, the PP, PP-Module, functional package or 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.

In FCS_CKM.1.1, the PP, PP-Module, functional package or 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 organizational standards.

E.2.4 FCS_CKM.2 Cryptographic key distribution

E.2.4.1 Component rationale and application notes

This component requires the method used to distribute cryptographic keys to be specified, this may be in accordance with an assigned standard. See ISO/IEC 15408-1 for information on using standards in PPs, PP-Modules, functional packages and STs.

E.2.4.2 Operations

In FCS_CKM.2.1 the PP, PP-Module, functional package or ST author should specify the cryptographic key distribution method to be used.

In FCS_CKM.2.1 the PP, PP-Module, functional package or 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 organizational standards.

E.2.5 FCS_CKM.3 Cryptographic key access

E.2.5.1 Component rationale and application notes

This component is intended to allow the specification of requirements on the usage of keys outside the TOE (backup, archival, escrow, recovery, etc.) and requires the methods used to access cryptographic keys be specified, this may be in accordance with an assigned standard.

FCS_CKM.3 Cryptographic key access is not intended to postulate the access control on cryptographic keys.

E.2.5.2 Operations

In FCS_CKM.3.1, the PP, PP-Module, functional package or ST author should specify the type of cryptographic key access being used.

EXAMPLE

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.

In FCS_CKM.3.1, the PP, PP-Module, functional package or ST author should specify the cryptographic key access method to be used.

In FCS_CKM.3.1, the PP, PP-Module, functional package or 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 organizational standards.

E.2.6 FCS_CKM.5 Cryptographic key derivation

E.2.6.1 Component rationale and application notes

This component requires the specification of the methods and parameters associated with the key derivation for a specified type of key.

FCS_CKM.5 has a dependency on FCS_CKM.6, The dependency should be understood as the dependency of two directions, 1) destruction of key derivation key, and 2) destruction of derived keys. PP, PP-Module, functional package and ST authors should keep in mind that the dependency of two directions has to be fulfilled, and should also consider any intermediate values (such as those from key establishment) that should be destroyed in order to preserve the confidentiality of the key.

E.2.7 FCS_CKM.6 Timing and event of cryptographic key destruction

E.2.7.1 Component rationale and application notes

This component requires the list of keys, including any keying material and the method used to destroy cryptographic keys to be specified, this can be in accordance with an assigned standard.

The purpose of the destruction of cryptographic keys and keying material is to prevent their recovery in consideration of threats related to their compromise.

NOTE Keying material includes keys and initialization vectors necessary to establish and maintain cryptographic keying relationships

E.2.7.2 Operations

In FCS_CKM.6.1, the PP, PP-Module, functional package or ST author provides a list of cryptographic keys and keying material that should be destroyed under certain circumstances.

In FCS_CKM.6.2, the PP, PP-Module, functional package or ST author provides the cryptographic key destruction method and the standards specifying the cryptographic key destruction method.

In FCS_CKM.6.1, the PP, PP-Module, functional package or ST author selects the circumstances of the destruction of key or key material.

E.3 Cryptographic operation (FCS_COP)

E.3.1 User application notes

A cryptographic operation may have cryptographic mode(s) of operation associated with it. If this is the case, then the cryptographic mode(s) shall be specified.

EXAMPLE

Examples of cryptographic modes of operation are cipher block chaining, output feedback mode, electronic code book mode, and cipher feedback mode.

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.

If Security audit data generation (FAU_GEN) Security audit data generation is included in the PP, PP-Module, functional package or 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.

When specifying cryptographic operations, the PP, PP-Module, functional package or ST author should perform due diligence in order to have confidence that the specified cryptographic operations are appropriate for the selected assurance requirements and in consideration of the technology types, environment and use cases of the TOE.

NOTE In some cases, certification bodies can apply policies in regard to the selection of cryptographic operations. (See ISO/IEC 18045 A.5 n).

E.3.2 FCS_COP.1 Cryptographic operation

E.3.2.1 Component rationale and application notes

This component requires the cryptographic algorithm and key size used to perform specified cryptographic operation(s) which can be based on an assigned standard.

The dependencies to FCS_RBG.1 or FCS_RNG.1 will be required for cryptographic algorithm operations which internally generate random numbers.

EXAMPLE 1

DSA signature generation, ECDSA signature generation, RSASSA-PSS signature generation.

The dependencies to FCS_RBG.1 or FCS_RNG.1 may not be necessary for deterministic cryptographic algorithm operations.

EXAMPLE 2

AES encryption / decryption in ECB mode.

E.3.2.2 Operations

In FCS_COP.1.1, the PP, PP-Module, functional package or ST author specifies 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.

In FCS_COP.1.1, the PP, PP-Module, functional package or ST author should specify the cryptographic algorithm to be used.

EXAMPLE

Examples of typical cryptographic algorithms include, but are not limited to, DES, RSA and IDEA.

In FCS_COP.1.1, the PP, PP-Module, functional package or 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.

In FCS_COP.1.1, the PP, PP-Module, functional package or 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, these may include standards from international, national, industry or organizational standards.

E.4 Random bit generation (FCS_RBG)

E.4.1 User application notes

When specifying random bit generation methods, the PP, PP-Module, functional package or ST author should perform due diligence in order to have confidence that the specifications are appropriate for the selected assurance requirements and in consideration of the technology types, environment and use cases of the TOE.

NOTE In some cases, certification bodies can apply policies in regard to the selection of random bit generators. (See ISO/IEC 18045 A.5 n).

E.4.2 FCS_RBG.1 Random bit generation (RBG)

E.4.2.1 Component rationale and application notes

For FCS_RBG.1, these dependencies shall always be met.

NOTE ISO/IEC 15408-1:20XX 8.3 item c) allowing a justification to be provided if a dependency is not met is not allowed for this component.

In the RBG State Update Table the ST author shall include a row for initialization (Source1). Other rows are optional, depending on the noise sources supported by the TSF. The identifier values identify the specific source, so there should be a row for every unique source, and if the same source is used for more than one update type then the same identifier is given.

If reseeding is not feasible, the TSF will uninstantiate RBGs (and instantiate a new RBG), rather than produce output that is of insufficient quality. The listed standards should specify the reseed interval, and procedure for uninstantiating and reseeding. The ‘Condition’ selection allows the PP Author to require application-specific conditions for reseeding.

“Uninstantiate” means that the internal state of the DRBG is no longer available for use.

In the ‘Condition’ selection, “on demand” means, that an interface to reseed is presented as a TSFI.

EXAMPLE

An example of an interface is an API call.

Health tests for the RBG are specified in FPT_TST.1.

E.4.3 FCS_RBG.2 Random bit generation (external seeding)

E.4.3.1 Component rationale and application notes

For this component, the interface to obtain the entropy noise source can be used multiple times to provide input. For instance, if the input length is 128 bits, it could be used twice to gather 256 bits. In this instance, the 128 bits would not be provided to the DRBG, since the DRBG can only be instantiated once, rather a function would gather the 128 bits twice and provide the DRBG with 256 bits of entropy noise source.

This component does not describe requirements on seed quality: it is the responsibility of the operational environment to define their requirement in this regard and to ensure that it is met by the external source.

Guidance in the introduction to PP, PP-Module, functional package or ST authors should address protection from modification and disclosure of the value from the external noise source, as well as the leaking of any pertinent information (e.g., internal state) regarding the RBG.

E.4.4 FCS_RBG.3 Random bit generation (internal seeding – single source)

E.4.4.1 Component rationale and application notes

If an ST Author wishes to use multiple internal noise sources, they iterate this requirement for each noise source being used by the TSF.

Hardware-based noise sources are sources whose primary function is noise generation, such as ring oscillators, diodes, and thermal noise. While software is used to collect the noise from these hardware sources, these are not software-based. Software-based noise sources are those sources that have some other primary function and the noise is a byproduct of their normal operation. Examples of software-based noise sources are user or system-based events, reading the least significant bits from an event timer, etc.

Hardware-based noise sources may be stochastically modeled, in which case the amount of entropy is well understood. Software-based noise sources are usually less well understood and therefore will typically take a more conservative approach, gathering larger numbers of bits than required and then performing a compression function to derive the final output. Software-based noise sources often rely on an entropy estimator.

E.5 Generation of random numbers (FCS_RNG)

Previous calls for contributions for the application notes for FCS_RNG have been made.

In the absence of any contribution, the editors have drawn from PP-0084B (https://www.commoncriteriaportal.org/files/ppfiles/pp0084b_pdf.pdf)

The editor requests careful review by SMEs.

E.5.1 User application notes

When specifying random number generation methods, the PP, PP-Module, functional package or ST author should perform due diligence in order to have confidence that the specifications are appropriate for the selected assurance requirements and in consideration of the technology types, environment and use cases of the TOE.

NOTE In some cases, certification bodies can apply policies in regard to the selection of random bit generators. (See ISO/IEC 18045 A.5 n).

E.5.2 FCS_RNG.1 Random number generation

E.5.2.1 Component rationale and application notes

The ST writer shall perform the missing operation appropriate for cryptographic application of the random numbers in the elements FCS_RNG.1.1 and FCS_RNG_1.2. The ST writer shall perform the selections for specification of the security capabilities provided by the random number generator of the TOE.

NOTE Some users of FCS_RNG may find The National Institute of Standards and Technology (NIST) Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015 and NIST Special Publication 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation, January 2018 useful.

The evaluation of the random number generator shall follow a recognized methodology,

EXAMPLE

An example of a recognized methodology is AIS31.

E.5.2.2 Operations

In FCS_RNG.1 .1 the PP, PP-Module, functional package or ST author should specify the list of security capabilities.

EXAMPLE 1

Examples of security capabilities include

- A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.
- If a total failure of the entropy source occurs while the RNG is being operated, the RNG [selection: prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy].
- The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.
- The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.
- The online test procedure checks the quality of the raw random number sequence. It is triggered [selection: externally, at regular intervals, continuously, applied upon specified internal events]. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.
- Failure or severe degradation of the noise source shall be detectable.
- Continuous tests or other mechanisms in the entropy source shall protect against producing output during malfunctions.

NOTE In the case of a PP, PP-Module or functional package, FCS_RNG.1 .1 could be completed with a more restrictive language such as:

- assignment: list of additional security capabilities.

In FCS_RNG.1.2 the PP, PP-Module, functional package or ST author should make the appropriate selection in regard to the quality metric.

EXAMPLE 2

Examples of quality metrics include

- Test procedure A [assignment: additional standard test suites] does not distinguish the internal random numbers from output sequences of an ideal RNG.
  NOTE The assignment for additional standard statistical test suite may be empty.
- The average Shannon entropy per internal random bit exceeds 0.99751.
- each output bit is independent of all other output bits,

NOTE In the case of a PP, PP-Module or functional package, FCS_RNG.1 .2 could be completed with a more restrictive language such as:

- [selection:
  - full entropy output,

- [assignment: bias and entropy rate of the output]]

EXAMPLE 3

In the case of a hybrid deterministic RNG, the following is an example:

FCS_RNG.1.1/HD

The TSF shall provide a hybrid deterministic55 random number generator that implements: [selection: CTR_DRBG, Hash_DRBG, HMAC_DRBG] as defined in NIST Special Publication 800-90A.

FCS_RNG.1.2/HD

The TSF shall provide [selection: bits, octets of bits, numbers [assignment: format of the numbers]] that meet [assignment: security bits].

Annex F
(normative)

Class FDP: User data protection- application notes

F.1 General information

This class contains families specifying requirements 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.

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.

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 shall adequately cover these three objectives in the PP, PP-Module, functional package or ST.

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, PP-Module, functional package or ST author whether these operations are described as “read” and/or “write” operations, or as more complex operations such as “update the database”.

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 policy 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 authorization, to change the attributes of the information.

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.

EXAMPLE 1

For example, a goal of having user-imposed (and user-defined) controls on information flow (such as. 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.

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.

EXAMPLE 2

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.

A set of SFRs 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.

When building a PP, PP-Module, functional package or 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.

The requirements in the FDP: User data protection class are defined in terms of a set of SFRs that will implement a SFP. Since a TOE may implement multiple SFPs simultaneously, the PP, PP-Module, functional package or ST author shall 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 SFP. This allows the author to easily indicate the scope for operations such as objects covered, operations covered, authorized users, etc.

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, PP-Module, functional package or ST to account for different policies if so desired.

The key to selecting components from this family is to have a well-defined set of TOE security objectives 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 functionality. 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).

The following steps are guidance on how this class is applied in the construction of a PP, PP-Module, functional package or 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 shall 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 of the TOE (FDP_ITC) and Export from the TOE (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.

F.2 Access control policy (FDP_ACC)

F.2.1 User application notes

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.

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 Export from the TOE (FDP_ETC). 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."

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.

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.

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.

This family provides a PP, PP-Module, functional package or 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, PP-Module, functional package or 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, PP-Module, functional package or 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.

F.2.2 FDP_ACC.1 Subset access control

F.2.2.1 Component rationale and application notes

The terms object and subject refer to generic elements in the TOE. For a policy to be implementable, the entities shall 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 TOE these generic terms (subject, object) shall be refined.

EXAMPLE

files, registers, ports, daemons, open calls, etc.

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.

F.2.2.2 Operations

In FDP_ACC.1.1, the PP, PP-Module, functional package or ST author should specify a uniquely named access control SFP to be enforced by the TSF.

In FDP_ACC.1.1, the PP, PP-Module, functional package or ST author should specify the list of subjects, objects, and operations among subjects and objects covered by the SFP.

F.2.3 FDP_ACC.2 Complete access control

F.2.3.1 Component rationale and application notes

This component requires that all possible operations on objects, that are included in the SFP, are covered by an access control SFP.

The PP, PP-Module, functional package or ST author shall demonstrate that each combination of objects and subjects is covered by an access control SFP.

F.2.3.2 Operations

In FDP_ACC.2.1, the PP, PP-Module, functional package or ST author should specify a uniquely named access control SFP to be enforced by the TSF.

In FDP_ACC.2.1, the PP, PP-Module, functional package or 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.3 Access control functions (FDP_ACF)

F.3.1 User application notes

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.

This family provides a PP, PP-Module, functional package or ST author the capability to describe the rules for access control. This results in a TOE 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 authorized administrator. This family also provides the PP, PP-Module, functional package or 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 authorization to access an object.

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.

A variety of acceptable access control functionality may be specified in this family.

EXAMPLE

- Access control lists (ACLs)
- Time-based access control specifications
- Origin-based access control specifications
- Owner-controlled access control attributes

F.3.2 FDP_ACF.1 Security attribute based access control

F.3.2.1 Component rationale and application notes

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 such as Access Control Lists (ACLs)). This component allows the PP, PP-Module, functional package or 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.

EXAMPLE

Examples of the attributes that a PP, PP-Module, functional package or ST author might assign are:

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.

A time attribute can be used to specify that access will be authorized during certain times of the day, during certain days of the week, or during a certain calendar year.

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.

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.

This component also provides requirements for the access control security functions to be able to explicitly authorize or deny access to an object based upon security attributes. This could be used to provide privilege, access rights, or access authorizations within the TOE. Such privileges, rights, or authorizations could apply to users, subjects (representing users or applications), and objects.

F.3.2.2 Operations

In FDP_ACF.1.1, the PP, PP-Module, functional package or 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).

In FDP_ACF.1.1, the PP, PP-Module, functional package or 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, PP-Module, functional package or 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.

In FDP_ACF.1.2, the PP, PP-Module, functional package or 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 or granular access control functions.

EXAMPLE

General access control functions: typical permission bits

Granular access control: Access Control Lists (ACL)

In FDP_ACF.1.3, the PP, PP-Module, functional package or ST author should specify the rules, based on security attributes, that explicitly authorize access of subjects to objects that will be used to explicitly authorize 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 authorize 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, PP-Module, functional package or ST author should specify "none".

In FDP_ACF.1.4, the PP, PP-Module, functional package or 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, PP-Module, functional package or ST author should specify "none".

F.4 Data authentication (FDP_DAU)

F.4.1 User application notes

This family describes specific functions that can be used to authenticate "static" data.

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 the Non-repudiation of origin (FCO_NRO) family provides for non-repudiation of origin of information received during a data exchange.

F.4.2 FDP_DAU.1 Basic Data Authentication

F.4.2.1 Component rationale and application notes

This component may be satisfied by one-way hash functions to generate a hash value for a definitive document that may be used as verification of the validity or authenticity of its information content.

EXAMPLE

cryptographic checksum, fingerprint, message digest

F.4.2.2 Operations

In FDP_DAU.1.1, the PP, PP-Module, functional package or ST author should specify the list of objects or information types for which the TSF shall be capable of generating data authentication evidence.

In FDP_DAU.1.2, the PP, PP-Module, functional package or 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.

F.4.3 FDP_DAU.2 Data Authentication with Identity of Guarantor

F.4.3.1 Component rationale and application notes

This component additionally requires the ability to verify the identity of the user that provided the guarantee of authenticity

EXAMPLE

A trusted third party.

F.4.3.2 Operations

In FDP_DAU.2.1, the PP, PP-Module, functional package or ST author should specify the list of objects or information types for which the TSF shall be capable of generating data authentication evidence.

In FDP_DAU.2.2, the PP, PP-Module, functional package or 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.5 Export from the TOE (FDP_ETC)

F.5.1 User application notes

This family defines functions for TSF-mediated exporting of 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).

Export from the TOE (FDP_ETC) is concerned with limitations on export and association of security attributes with the exported user data.

This family, and the corresponding Import family Import from outside of the TOE (FDP_ITC), address how the TOE deals with user data transferred into and outside its control. In principle, this family is concerned with the TSF-mediated exporting of user data and its related security attributes.

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.

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.

F.5.2 FDP_ETC.1 Export of user data without security attributes

F.5.2.1 Component rationale and application notes

This component is used to specify the TSF-mediated exporting of user data without the export of its security attributes.

F.5.2.2 Operations

In FDP_ETC.1.1, the PP, PP-Module, functional package or 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.

F.5.3 FDP_ETC.2 Export of user data with security attributes

F.5.3.1 Component rationale and application notes

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.

EXAMPLE

On the same external media

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.

F.5.3.2 Operations

In FDP_ETC.2.1, the PP, PP-Module, functional package or 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.

In FDP_ETC.2.5, the PP, PP-Module, functional package or 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.6 Information flow control policy (FDP_IFC)

F.6.1 User application notes

This family covers the identification of information flow control SFPs; and, for each, specifies the scope of control of the SFP.

The components in this family are capable of identifying the information flow control SFPs to be enforced by the traditional Mandatory Access Control (MAC) 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 information flow control SFP will be defined by other families such as Information flow control functions (FDP_IFF) and Export from the TOE (FDP_ETC). 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.”

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.

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.

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 shall allow this flow before it is permitted to take place.

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.

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 authorization, to change the attributes of the information.

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.

The components in this family can be applied multiple times in a PP, PP-Module, functional package or 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.

F.6.2 FDP_IFC.1 Subset information flow control

F.6.2.1 Component rationale and application notes

This component requires that an information flow control policy apply to a subset of the possible operations in the TOE.

F.6.2.2 Operations

In FDP_IFC.1.1, the PP, PP-Module, functional package or ST author should specify a uniquely named information flow control SFP to be enforced by the TSF.

In FDP_IFC.1.1, the PP, PP-Module, functional package or 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, PP-Module, functional package or ST author.

EXAMPLE

It could specify users, machines, or processes.

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 shall be enforced before the specified information could flow to or from the object.

F.6.3 FDP_IFC.2 Complete information flow control

F.6.3.1 Component rationale and application notes

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.

The PP, PP-Module, functional package or ST author shall demonstrate that each combination of information flows and subjects is covered by an information flow control SFP.

F.6.3.2 Operations

In FDP_IFC.2.1, the PP, PP-Module, functional package or ST author should specify a uniquely named information flow control SFP to be enforced by the TSF.

In FDP_IFC.2.1, the PP, PP-Module, functional package or 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, PP-Module, functional package or ST author.

EXAMPLE

It could specify users, machines, or processes.

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 shall be enforced before the specified information could flow to or from the object.

F.7 Information flow control functions (FDP_IFF)

F.7.1 User application notes

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.

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.

In this family, the phrase “types of illicit information flows” is used. This phrase may be used to refer to the categorization of flows as “Storage Channels” or “Timing Channels”, or it can refer to improved categorizations reflective of the needs of a PP, PP-Module, functional package or ST author.

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, PP-Module, functional package or ST author should consider incorporating a privilege policy.

F.7.2 FDP_IFF.1 Simple security attributes

F.7.2.1 Component rationale and application notes

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.

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.

This component also provides requirements for the information flow control functions to be able to explicitly authorize 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.

F.7.2.2 Operations

In FDP_IFF.1.1, the PP, PP-Module, functional package or 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).

In FDP_IFF.1.1, the PP, PP-Module, functional package or 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.

EXAMPLE 1

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.

In FDP_IFF.1.2, the PP, PP-Module, functional package or 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.

In FDP_IFF.1.3, the PP, PP-Module, functional package or ST author should specify any additional information flow control SFP rules that the TSF is to enforce. This includes all rules of the SFP that are either not based on the security attributes of the information and the subject or rules that automatically modify the security attributes of information or subjects as a result of an access operation. An example for the first case is a rule of the SFP controlling a threshold value for specific types of information. This would for example be the case when the information flow SFP contains rules on access to statistical data where a subject is only allowed to access this type of information up to a specific number of accesses. An example for the second case would be a rule stating under which conditions and how the security attributes of a subject or object change as the result of an access operation. Some information flow policies for example may limit the number of access operations to information with specific security attributes. If there are no additional rules then the PP, PP-Module, functional package or ST author should specify “none”.

In FDP_IFF.1.4, the PP, PP-Module, functional package or ST author should specify the rules, based on security attributes, that explicitly authorize information flows. These rules are in addition to those specified in the preceding elements. They are included in FDP_IFF.1.4 as they are intended to contain exceptions to the rules in the preceding elements.

EXAMPLE 2

An example of rules to explicitly authorize 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, PP-Module, functional package or ST author should specify “none”.

In FDP_IFF.1.5, the PP, PP-Module, functional package or 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.1.5 as they are intended to contain exceptions to the rules in the preceding elements. An example of rules to explicitly deny 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, PP-Module, functional package or ST author should specify “none”.

F.7.3 FDP_IFF.2 Hierarchical security attributes

F.7.3.1 Component rationale and application notes

This component requires that the named information flow control SFP uses hierarchical security attributes that form a lattice.

It is important to note that the hierarchical relationship requirements identified in FDP_IFF.2.4 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.

FDP_IFF.2.6 phrases the requirements for the set of security attributes to form a lattice. A number of information flow policies defined in the literature and implemented in IT products are based on a set of security attributes that form a lattice. FDP_IFF.2.6 is specifically included to address this type of information flow policies.

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, PP-Module, functional package or ST author should iterate this component once for each of those SFPs. Otherwise a conflict might arise with the sub-items of FDP_IFF.2.4 since the required relationships will not exist.

F.7.3.2 Operations

In FDP_IFF.2.1, the PP, PP-Module, functional package or 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).

In FDP_IFF.2.1, the PP, PP-Module, functional package or 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.

In FDP_IFF.2.2, the PP, PP-Module, functional package or 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.

In FDP_IFF.2.3, the PP, PP-Module, functional package or ST author should specify any additional information flow control SFP rules that the TSF is to enforce. This includes all rules of the SFP that are either not based on the security attributes of the information and the subject or rules that automatically modify the security attributes of information or subjects as a result of an access operation. An example for the first case is a rule of the SFP controlling a threshold value for specific types of information.

EXAMPLE 1

This would for example be the case when the information flow SFP contains rules on access to statistical data where a subject is only allowed to access this type of information up to a specific number of accesses. An example for the second case would be a rule stating under which conditions and how the security attributes of a subject or object change as the result of an access operation.

Some information flow policies may limit the number of access operations to information with specific security attributes. If there are no additional rules then the PP, PP-Module, functional package or ST author should specify “none”.

In FDP_IFF.2.4, the PP, PP-Module, functional package or ST author should specify the rules, based on security attributes, that explicitly authorize information flows. These rules are in addition to those specified in the preceding elements. They are included in FDP_IFF.2.4 as they are intended to contain exceptions to the rules in the preceding elements.

EXAMPLE 2

An example of rules to explicitly authorize 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, PP-Module, functional package or ST author should specify "none".

In FDP_IFF.2.5, the PP, PP-Module, functional package or 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.2.5 as they are intended to contain exceptions to the rules in the preceding elements. An example of rules to explicitly deny 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, PP-Module, functional package or ST author should specify "none".

F.7.4 FDP_IFF.3 Limited illicit information flows

F.7.4.1 Component rationale and application notes

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.

For the specified illicit information flows, certain maximum capacities should be provided. In addition, a PP, PP-Module, functional package or ST author has the ability to specify whether the illicit information flows must be audited.

F.7.4.2 Operations

In FDP_IFF.3.1, the PP, PP-Module, functional package or 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).

In FDP_IFF.3.1, the PP, PP-Module, functional package or ST author should specify the types of illicit information flows that are subject to a maximum capacity limitation.

In FDP_IFF.3.1, the PP, PP-Module, functional package or ST author should specify the maximum capacity permitted for any identified illicit information flows.

F.7.5 FDP_IFF.4 Partial elimination of illicit information flows

F.7.5.1 Component rationale and application notes

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.

F.7.5.2 Operations

In FDP_IFF.4.1, the PP, PP-Module, functional package or 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).

In FDP_IFF.4.1, the PP, PP-Module, functional package or ST author should specify the types of illicit information flows which are subject to a maximum capacity limitation.

In FDP_IFF.4.1, the PP, PP-Module, functional package or ST author should specify the maximum capacity permitted for any identified illicit information flows.

In FDP_IFF.4.2, the PP, PP-Module, functional package or 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.

F.7.6 FDP_IFF.5 No illicit information flows

F.7.6.1 Component rationale and application notes

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, PP-Module, functional package or 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.

F.7.6.2 Operations

In FDP_IFF.5.1, the PP, PP-Module, functional package or 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).

F.7.7 FDP_IFF.6 Illicit information flow monitoring

F.7.7.1 Component rationale and application notes

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.

F.7.7.2 Operations

In FDP_IFF.6.1, the PP, PP-Module, functional package or 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).

In FDP_IFF.6.1, the PP, PP-Module, functional package or ST author should specify the types of illicit information flows that will be monitored for exceeding a maximum capacity.

In FDP_IFF.6.1, the PP, PP-Module, functional package or ST author should specify the maximum capacity above which illicit information flows will be monitored by the TSF.

F.8 Information retention control (FDP_IRC)

F.8.1 User application notes

While a great aspect of the elimination of the objects as required by FDP_IRC refers to the information stored within the object as a container, it also includes all attributes (also in the meaning of metadata) that may be associated with the object.

In this aspect, the focus of FDP_IRC differs from other components related to access or information flow control policies, such as FDP_IFF and FDP_IFC. More important, objects here are always considered in the context of selected activities that are performed on these objects. In contrast to residual information protection (FDP_RIP), FDP_IRC excludes objects from any access or information flow and deletes them, irreversibly and untraceably when they are no longer needed by a set of activities.

While it may not be completely clear, which objects to consider, it is essential that the list of objects is assigned by the PP, PP-Module, functional package or ST author at the very latest in order to allow for concrete tests. In any case the list of objects shall be derived from a structured analysis.

F.8.2 FDP_IRC.1 Information retention control

F.8.2.1 Component rationale and application notes

The Information erasure policy as defined in FDP_IRC.1 serves to protect all information that is contained in the assigned objects from being misused, regardless of whether the information is primary content or any kind of attribute. The policy covers combinations of objects and activities. The policy's coverage may be "complete" with respect to all the objects related to one or more activities, or it may address only some of objects related to one or more activities.

The term "promptly" in FDP_IRC.1 specifically refers to the fact that the objects shall be terminated in a manner that ensures that they cannot be accessed as before.

F.8.2.2 Operations

In FDP_IRC.1.1, the PP, PP-Module, functional package or ST author should specify a uniquely named information erasure policy to be enforced by the TSF.

In FDP_IRC.1.1, the PP, PP-Module, functional package or ST author should specify the list of objects that are required for the respective list of activities, e.g. "all message objects".

In FDP_IRC.1.1, the PP, PP-Module, functional package or ST author should specify the list of activities that the information erasure policy is concerned with, e.g. "all activities related to passing a message on, such as receiving a message, cryptographic handling of a message, sending a message".

In FDP_IRC.1.2, the PP, PP-Module, functional package or ST author should specify the list of objects that are required for the respective list of activities. This assignment shall be identical to the assigned objects in FDP_IRC.1.1.

F.9 Import from outside of the TOE (FDP_ITC)

F.9.1 User application notes

This family defines mechanisms for TSF-mediated importing of user data from outside the TOE 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).

Import from outside of the TOE (FDP_ITC) is concerned with limitations on import, user specification of security attributes, and association of security attributes with the user data.

This family, and the corresponding export family Export from the TOE (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.

EXAMPLE 1

A variety of activities might be involved here:

a) importing user data from an unformatted medium (such as., tape, scanner, video or audio 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.

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.

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.

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).

EXAMPLE 2

Cryptographic checksum

This family is concerned with TSF-mediated importing of 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 of the TOE (FDP_ITC) is only concerned with the interface to the import medium. Export from the TOE (FDP_ETC) is responsible for the other end point of the medium (the source).

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.

These import requirements may be handled by the TSF with or without human intervention, depending on the IT limitations and the organizational security policy. For example, if user data is received on a “confidential” channel, the security attributes of the objects will be set to “confidential”.

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.

F.9.2 FDP_ITC.1 Import of user data without security attributes

F.9.2.1 Component rationale and application notes

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 initialized within the TSF. It could also be the case that the PP, PP-Module, functional package or 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.

F.9.2.2 Operations

In FDP_ITC.1.1, the PP, PP-Module, functional package or 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 TOE. The user data that this function imports is scoped by the assignment of these SFPs.

In FDP_ITC.1.3, the PP, PP-Module, functional package or 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.

F.9.3 FDP_ITC.2 Import of user data with security attributes

F.9.3.1 Component rationale and application notes

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, PP-Module, functional package or ST author specifies the rules for import.

F.9.3.2 Operations

In FDP_ITC.2.1, the PP, PP-Module, functional package or 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 TOE. The user data that this function imports is scoped by the assignment of these SFPs.

In FDP_ITC.2.5, the PP, PP-Module, functional package or 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.10 Internal TOE transfer (FDP_ITT)

F.10.1 User application notes

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 from the TOE (FDP_ETC) and Import from outside of the TOE (FDP_ITC), which address TSF-mediated transfer of data to or from outside the TOE.

The requirements in this family allow a PP, PP-Module, functional package or 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.

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.

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.

F.10.2 FDP_ITT.1 Basic internal transfer protection

F.10.2.1 Operations

In FDP_ITT.1.1, the PP, PP-Module, functional package or ST author should specify the access control SFP(s) and/or information flow control SFP(s) covering the information being transferred.

In FDP_ITT.1.1, the PP, PP-Module, functional package or ST author should specify the types of transmission errors that the TSF should prevent occurring for user data while in transport. The options are disclosure, modification, loss of use.

F.10.3 FDP_ITT.2 Transmission separation by attribute

F.10.3.1 Component rationale and application notes

This component could, for example, be used to provide different forms of protection to information with different clearance levels.

One of the ways to achieve separation of data when it is transmitted is through the use of separate logical or physical channels.

F.10.3.2 Operations

In FDP_ITT.2.1, the PP, PP-Module, functional package or ST author should specify the access control SFP(s) and/or information flow control SFP(s) covering the information being transferred.

In FDP_ITT.2.1, the PP, PP-Module, functional package or ST author should specify the types of transmission errors that the TSF should prevent occurring for user data while in transport. The options are disclosure, modification, loss of use.

In FDP_ITT.2.2, the PP, PP-Module, functional package or ST author should specify the security attributes, the values of which the TSF will use to determine when to separate data that is being transmitted 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.

F.10.4 FDP_ITT.3 Integrity monitoring

F.10.4.1 Component rationale and application notes

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).

The PP, PP-Module, functional package or ST author has to specify the types of errors that must be detected. The PP, PP-Module, functional package or 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.

The PP, PP-Module, functional package or ST author must specify the actions that the TSF should take on detection of a failure.

EXAMPLE

For example: ignore the user data, request the data again, inform the authorized administrator, reroute traffic for other lines.

F.10.4.2 Operations

In FDP_ITT.3.1, the PP, PP-Module, functional package or 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.

In FDP_ITT.3.1, the PP, PP-Module, functional package or ST author should specify the type of possible integrity errors to be monitored during transmission of the user data.

In FDP_ITT.3.2, the PP, PP-Module, functional package or ST author should specify the action to be taken by the TSF when an integrity error is encountered.

EXAMPLE

An example is 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.10.5 FDP_ITT.4 Attribute-based integrity monitoring

F.10.5.1 Component rationale and application notes

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, PP-Module, functional package or ST author to specify actions to be taken upon detection of an integrity error.

EXAMPLE 1

This component could be used to provide different integrity error detection and action for information at different integrity levels.

The PP, PP-Module, functional package or ST author has to specify the types of errors that must be detected. The PP, PP-Module, functional package or 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.

The PP, PP-Module, functional package or ST author should specify the attributes (and associated transmission channels) that necessitate integrity error monitoring.

The PP, PP-Module, functional package or ST author must specify the actions that the TSF should take on detection of a failure.

EXAMPLE 2

For example: ignore the user data, request the data again, inform the authorized administrator, reroute traffic for other lines.

F.10.5.2 Operations

In FDP_ITT.4.1, the PP, PP-Module, functional package or 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.

In FDP_ITT.4.1, the PP, PP-Module, functional package or ST author should specify the type of possible integrity errors to be monitored during transmission of the user data.

In FDP_ITT.4.1, the PP, PP-Module, functional package or 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.

In FDP_ITT.4.2, the PP, PP-Module, functional package or 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.4.1 will be enforced as the actions are taken by the TSF.

F.11 Residual information protection (FDP_RIP)

F.11.1 User application notes

Residual information protection ensures that TSF-controlled resources when de-allocated from an object and before they are reallocated to another object are treated by the TSF in a way that it is not possible to reconstruct all or part of the data contained in the resource before it was de-allocated.

A TOE usually has a number of functions that potentially de-allocate resources from an object and potentially re-allocate those resources to objects. Some, but not all of those resources may have been used to store critical data from the previous use of the resource and for those resources FDP_RIP requires that they are prepared for reuse. Object reuse applies to explicit requests of a subject or user to release resources as well as implicit actions of the TSF that result in the de-allocation and subsequent re-allocation of resources to different objects.

EXAMPLE

Examples of explicit requests are the deletion or truncation of a file or the release of an area of main memory. Examples of implicit actions of the TSF are the de-allocation and re-allocation of cache regions.

The requirement for object reuse is related to the content of the resource belonging to an object, not all information about the resource or object that may be stored elsewhere in the TSF. As an example, to satisfy the FDP_RIP requirement for files as objects requires that all sectors that make up the file need to be prepared for re-use.

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.

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”.

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, PP-Module, functional package or ST author should make sure that the proper environmental objectives are in place to support operational user guidance to address off-line objects.

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.

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.

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, PP-Module, functional package or ST author.

F.11.2 FDP_RIP.1 Subset residual information protection

F.11.2.1 Component rationale and application notes

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.

F.11.2.2 Operations

In FDP_RIP.1.1, the PP, PP-Module, functional package or ST author should specify the event, allocation of the resource to or deallocation of the resource from, that invokes the residual information protection function.

In FDP_RIP.1.1, the PP, PP-Module, functional package or ST author should specify the list of objects subject to residual information protection.

F.11.3 FDP_RIP.2 Full residual information protection

F.11.3.1 Component rationale and application notes

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.

F.11.3.2 Operations

In FDP_RIP.2.1, the PP, PP-Module, functional package or 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.12 Rollback (FDP_ROL)

F.12.1 User application notes

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.

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.

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.

The rollback requirement is bounded by certain limits.

EXAMPLE

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.

F.12.2 FDP_ROL.1 Basic rollback

F.12.2.1 Component rationale and application notes

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.

F.12.2.2 Operations

In FDP_ROL.1.1, the PP, PP-Module, functional package or 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.

In FDP_ROL.1.1, the PP, PP-Module, functional package or ST author should specify the list of operations that can be rolled back.

In FDP_ROL.1.1, the PP, PP-Module, functional package or ST author should specify the information and/or list of objects that are subjected to the rollback policy.

In FDP_ROL.1.2, the PP, PP-Module, functional package or 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,

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.12.3 FDP_ROL.2 Advanced rollback

F.12.3.1 Component rationale and application notes

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.

F.12.3.2 Operations

In FDP_ROL.2.1, the PP, PP-Module, functional package or 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.

In FDP_ROL.2.1, the PP, PP-Module, functional package or ST author should specify the list of objects that are subjected to the rollback policy.

In FDP_ROL.2.2, the PP, PP-Module, functional package or 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,

EXAMPLE

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.13 Stored data confidentiality (FDP_SDC)

F.13.1 User application notes

This family provides requirements that address protection of user data confidentiality while the data is stored within memory areas protected by the TSF. The TSF provides access to the data in the memory through the specified interfaces only and prevents compromise of their information bypassing these interfaces. It complements the family Stored data integrity (FDP_SDI) which protects the user data from integrity errors while being stored in the memory.

F.13.2 Evaluator notes

In practice, the dependency to FCS_COP.1 may be satisfied by a PP, PP-Module, functional package or ST author by providing a rationale explaining an alternative method to cryptography is used in some dedicated cases.

F.13.3 FDP_SDC.1 Stored data confidentiality

F.13.3.1 Component rationale and application notes

In FDP_SDC.1 Stored data confidentiality, the PP, PP-Module, functional package or ST author specifies which user data is to be protected and in which type of memory the user data is requested to be protected. In the second selection the PP, PP-Module, functional package or ST author provides the memory type where the user data is to be protected.

F.13.3.2 Operations

In FDP_SDC.1.1 the PP, PP-Module, functional package or ST author shall select either “all user data” or provide a list of user data using the assignment below. In the second selection, the PP, PP-Module, functional package or ST author can specify either temporary memory, persistent memory or any memory. “Any memory” includes both temporary (volatile) and persistent (non-volatile) memory.

In FDP_SDC.1.1 the PP, PP-Module, functional package or ST author provides a list of the user data that is to be protected in memory.

F.13.4 FDP_SDC.2 Stored data confidentiality with dedicated method

F.13.4.1 Component rationale and application notes

FDP_SDC.2 Stored data confidentiality with dedicated method refines the FDP_SDC.1.1 element by allowing the PP, PP-Module, functional package or ST author to refine the list of user data using a variety of data characteristics.

F.13.4.2 Operations

The operations of selection and the first assignment are the same as that in FDP_SDC.1.

For the second assignment the PP, PP-Module, functional package or ST author provides the data characteristics. Data characteristics can include items such as data length (shorter or longer than a threshold), data type (binary, text, image, sound, video), and data representation (binary, vector, character, frame).

F.14 Stored data integrity (FDP_SDI)

F.14.1 User application notes

This family provides requirements that address protection of user data while it is stored within containers controlled by the TSF.

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 controlled by the TSF are also addressed by this family.

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).

This family differs from Internal TOE transfer (FDP_ITT) that protects the user data from integrity errors while being transferred within the TOE.

F.14.2 FDP_SDI.1 Stored data integrity monitoring

F.14.2.1 Component rationale and application notes

This component monitors data stored on media for integrity errors. The PP, PP-Module, functional package or ST author can specify different kinds of user data attributes that will be used as the basis for monitoring.

F.14.2.2 Operations

In FDP_SDI.1.1, the PP, PP-Module, functional package or ST author should specify the integrity errors that the TSF will detect.

In FDP_SDI.1.1, the PP, PP-Module, functional package or ST author should specify the user data attributes that will be used as the basis for the monitoring.

F.14.3 FDP_SDI.2 Stored data integrity monitoring and action

F.14.3.1 Component rationale and application notes

This component monitors data stored on media for integrity errors. The PP, PP-Module, functional package or ST author can specify which action should be taken in case an integrity error is detected.

F.14.3.2 Operations

In FDP_SDI.2.1, the PP, PP-Module, functional package or ST author should specify the integrity errors that the TSF will detect.

In FDP_SDI.2.1, the PP, PP-Module, functional package or ST author should specify the user data attributes that will be used as the basis for the monitoring.

In FDP_SDI.2.2, the PP, PP-Module, functional package or ST author should specify the actions to be taken in case an integrity error is detected.

F.15 Inter-TSF user data confidentiality transfer protection (FDP_UCT)

F.15.1 User application notes

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 unauthorized disclosure of user data in transit between the two end points. The end points may be a TSF or a user.

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.

F.15.2 FDP_UCT.1 Basic data exchange confidentiality

F.15.2.1 Component rationale and application notes

Depending on the access control or information flow policies the TSF is required to send or receive user data in a manner such that the confidentiality of the user data is protected.

F.15.2.2 Operations

In FDP_UCT.1.1, the PP, PP-Module, functional package or 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.

In FDP_UCT.1.1, the PP, PP-Module, functional package or ST author should specify whether this element applies to a mechanism that transmits or receives user data.

F.16 Inter-TSF user data integrity transfer protection (FDP_UIT)

F.16.1 User application notes

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.

This family defines the requirements for providing integrity for user data in transit; while Integrity of exported TSF data (FPT_ITI) handles TSF data.

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 of the TOE (FDP_ITC).

F.16.2 FDP_UIT.1 Data exchange integrity

F.16.2.1 Component rationale and application notes

Depending on the access control or information flow policies the TSF is required to send or receive user data in a manner such that modification of the user data is detected. There is no requirement for a TSF mechanism to attempt to recover from the modification.

F.16.2.2 Operations

In FDP_UIT.1.1, the PP, PP-Module, functional package or 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.

In FDP_UIT.1.1, the PP, PP-Module, functional package or ST author should specify whether this element applies to a TSF that is transmitting or receiving objects.

In FDP_UIT.1.1, the PP, PP-Module, functional package or ST author should specify whether the data should be protected from modification, deletion, insertion, or replay.

In FDP_UIT.1.2, the PP, PP-Module, functional package or ST author should specify whether the errors of the type: modification, deletion, insertion, or replay are detected.

F.16.3 FDP_UIT.2 Source data exchange recovery

F.16.3.1 Component rationale and application notes

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 TOE, 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.

EXAMPLE

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.

F.16.3.2 Operations

In FDP_UIT.2.1, the PP, PP-Module, functional package or 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.

In FDP_UIT.2.1, the PP, PP-Module, functional package or 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.

F.16.4 FDP_UIT.3 Destination data exchange recovery

F.16.4.1 Component rationale and application notes

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.

F.16.4.2 Operations

In FDP_UIT.3.1, the PP, PP-Module, functional package or 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.

In FDP_UIT.3.1, the PP, PP-Module, functional package or ST author should specify the list of integrity errors from which the receiving TSF, alone, is able to recover the original user data.

Annex G
(normative)

Class FIA: Identification and authentication- application notes

G.1 General Information

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.

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.

EXAMPLE

Security attributes include identity, groups, roles, security, or integrity levels.

The unambiguous identification of authorized users and the correct association of security attributes with users and subjects is critical to the enforcement of the security policies.

The Authentication failures (FIA_AFL) family addresses defining limits on repeated unsuccessful authentication attempts.

The Authentication proof of identity (FIA_API) family addresses defining the functionality provided by the TOE to prove its identity and to be verified by an external entity in the TOE IT environment.

The User attribute definition (FIA_ATD) family addresses the definition of user attributes that are used in the enforcement of the SFRs.

The Specification of secrets (FIA_SOS) family addresses the generation and verification of secrets that satisfy a defined metric.

The User authentication (FIA_UAU) family addresses verifying the identity of a user.

The User identification (FIA_UID) family addresses determining the identity of a user.

The User-subject binding (FIA_USB) family addresses the correct association of security attributes for each authorized user.

G.2 Authentication failures (FIA_AFL)

G.2.1 User application notes

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.

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.

G.2.2 FIA_AFL.1 Authentication failure handling

G.2.2.1 Component rationale and application notes

The PP, PP-Module, functional package or ST author may define the number of unsuccessful authentication attempts or may choose to let the TOE developer or the authorized 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.

The PP, PP-Module, functional package or ST author could specify a list of actions that the TSF shall take in the case of authentication failure. An authorized administrator could also be allowed to manage the events, if deemed opportune by the PP, PP-Module, functional package or 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.

In order to prevent denial of service, TOEs usually ensure that there is at least one user account that cannot be disabled.

Further actions for the TSF can be stated by the PP, PP-Module, functional package or ST author, including rules for re-enabling the user session establishment process, or sending an alarm to the administrator.

EXAMPLE

Examples of these actions are: until a specified time has lapsed, until the authorized administrator re-enables the terminal/account, a time related to failed previous attempts (every time the attempt fails, the disabling time is doubled).

G.2.2.2 Operations

In FIA_AFL.1 Authentication failure handling, the PP, PP-Module, functional package or 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.

In FIA_AFL.1 Authentication failure handling, the PP, PP-Module, functional package or 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.

In FIA_AFL.1 Authentication failure handling, if the assignment of a positive integer is selected, the PP, PP-Module, functional package or ST author should specify the default number (positive integer) of unsuccessful authentication attempts that, when met or surpassed, will trigger the events.

In FIA_AFL.1 Authentication failure handling, if an administrator configurable positive integer is selected, the PP, PP-Module, functional package or 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.

In FIA_AFL.1.2, the PP, PP-Module, functional package or ST author should select whether the event of meeting or surpassing the defined number of unsuccessful authentication attempts shall trigger an action by the TSF.

In FIA_AFL.1.2, the PP, PP-Module, functional package or ST author should specify the actions to be taken in case the threshold is met or surpassed, as selected. 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.3 Authentication proof of identity (FIA_API)

G.3.1 User application notes

The other families of the Class FIA describe only the authentication verification of users' identity performed by the TOE and do not describe the functionality of the user to prove their identity. The family FIA_API allows the specification the functionality allowing a TOE to prove its own identity.

G.3.2 FIA_API.1 Authentication proof of identity

Editors' Note:

Since no contributions were received for this text, the editors have proposed the text below. The editors request that SMEs review carefully.

If no comments are received during the commenting period for this draft, the editors' proposal will be accepted in the next draft.

G.3.2.1 Component rationale and application notes

FIA_API.1 Authentication proof of identity allows the specification of the authentication mechanism used to support proving the identity of the TOE to external entities.

G.3.2.2 Operations

The first assignment is where a PP, PP-Module, functional package or ST author specifies the authentication mechanism to be used.

EXAMPLE

Examples of such an authentication method is "an Authentication Mechanism based on Triple-DES" and "Chip Authentication Protocol according to TR-03110"

The second assignment allows the PP, PP-Module, functional package or ST author to specify to what the proof of identity is associated with. This can be an object, authorized user or a role.

Editors' Note:

Editors observe that in many STs using this component the second completion is for "TOE" which is neither an object, authorized user or a role. Should FIA_API.1.1 be updated to allow for the specification of TOE in the assignment?

The editors' welcome comments on this point. If no comments are received then then text will remain unaltered in the next draft.

G.4 User attribute definition (FIA_ATD)

G.4.1 User application notes

All authorized users may have a set of security attributes, other than the user's identity, that are used to enforce the SFRs. This family defines the requirements for associating user security attributes with users as needed to support the TSF in making security decisions.

There are dependencies on the individual security policy (SFP) definitions. These individual definitions should contain the listing of attributes that are necessary for policy enforcement.

G.4.2 FIA_ATD.1 User attribute definition

G.4.2.1 Component rationale and application notes

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.

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.

G.4.2.2 Operations

In FIA_ATD.1.1, the PP, PP-Module, functional package or ST author should specify the security attributes that are associated to an individual user.

EXAMPLE

An example of such a list is {“clearance”, “group identifier”, “rights”}.

G.5 Specification of secrets (FIA_SOS)

G.5.1 User application notes

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.

A secret can be generated outside the TOE.

EXAMPLE

An example of a secret generated outside of the TOE could be one that is selected by the user and introduced in the TOE.

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.

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.

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.

G.5.2 FIA_SOS.1 Verification of secrets

G.5.2.1 Component rationale and application notes

Secrets can be generated by the user. This component ensures that those user generated secrets can be verified to meet a certain quality metric.

G.5.2.2 Operations

In FIA_SOS.1.1, the PP, PP-Module, functional package or 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.

EXAMPLE

quality metrics could include a description of the alphanumeric structure of acceptable secrets and/or the space size that acceptable secrets must meet.

G.5.3 FIA_SOS.2 TSF Generation of secrets

G.5.3.1 Component rationale and application notes

This component allows the TSF to generate secrets for specific functions such as authentication by means of passwords.

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.

G.5.3.2 Operations

In FIA_SOS.2.1, the PP, PP-Module, functional package or 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.

EXAMPLE

quality metrics could include a description of the alphanumeric structure of acceptable secrets and/or the space size that acceptable secrets must meet.

In FIA_SOS.2.2, the PP, PP-Module, functional package or 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.6 User authentication (FIA_UAU)

G.6.1 User application notes

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.

G.6.2 FIA_UAU.1 Timing of authentication

G.6.2.1 Component rationale and application notes

This component requires that the PP, PP-Module, functional package or 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.

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 or FIA_UID.2 User identification before any action with the appropriate assignments.

G.6.2.2 Operations

In FIA_UAU.1.1, the PP, PP-Module, functional package or 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.

EXAMPLE

Such an action might include the request for help on the login procedure.

G.6.3 FIA_UAU.2 User authentication before any action

G.6.3.1 Component rationale and application notes

This component requires that a user is authenticated before any other TSF-mediated action can take place on behalf of that user.

G.6.4 FIA_UAU.3 Unforgeable authentication

G.6.4.1 Component rationale and application notes

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.

This component may be useful only with authentication mechanisms that are based on authentication data that cannot be shared. It is impossible for a TSF to detect or prevent the sharing of passwords outside the control of the TSF.

~~EXAMPLE~~

~~An example of authentication data that cannot be shared is biometrics~~

Editors' Note

Is this a good example? Editors' suggest that replay attacks could be "sharing" biometrics.

If no comments are received, then this example will be deleted in the next draft

G.6.4.2 Operations

In FIA_UAU.3.1, the PP, PP-Module, functional package or ST author should specify whether the TSF will detect, prevent, or detect and prevent forging of authentication data.

In FIA_UAU.3.2, the PP, PP-Module, functional package or ST author should specify whether the TSF will detect, prevent, or detect and prevent copying of authentication data.

G.6.5 FIA_UAU.4 Single-use authentication mechanisms

G.6.5.1 Component rationale and application notes

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.

EXAMPLE

Single-use authentication data include single-use passwords, encrypted time-stamps, and/or random numbers from a secret lookup table.

The PP, PP-Module, functional package or ST author can specify to which authentication mechanism(s) this requirement applies.

G.6.5.2 Operations

In FIA_UAU.4.1, the PP, PP-Module, functional package or 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".

G.6.6 FIA_UAU.5 Multiple authentication mechanisms

G.6.6.1 Component rationale and application notes

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.

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.

To allow anonymous users to interact with the TOE, a "none" authentication mechanism can be incorporated. The use of such access should be clearly explained in the rules of FIA_UAU.5.2.

G.6.6.2 Operations

In FIA_UAU.5.1, the PP, PP-Module, functional package or ST author should define the available authentication mechanisms.

EXAMPLE 1

Such a list could be: “none, password mechanism, biometric (retinal scan), S/key mechanism”.

In FIA_UAU.5.2, the PP, PP-Module, functional package or 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.

EXAMPLE 2

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.”

The PP, PP-Module, functional package or ST author might give the boundaries within which the authorized 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, PP-Module, functional package or ST author also might choose not to specify any boundaries but leave the authentication mechanisms and their rules completely up to the authorized administrator.

G.6.7 FIA_UAU.6 Re-authenticating

G.6.7.1 Component rationale and application notes

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.

EXAMPLE

A server application requesting that the TSF re-authenticate the client it is serving.

G.6.7.2 Operations

In FIA_UAU.6.1, the PP, PP-Module, functional package or 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.

The PP, PP-Module, functional package or ST author might give the boundaries within which the re-authentication should occur and leave the specifics to the authorized administrator.

EXAMPLE

“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.”

G.6.8 FIA_UAU.7 Protected authentication feedback

G.6.8.1 Component rationale and application notes

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.

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” for each password character provided, and not the original character.

Example

A “dummy” could be a star ”*” character.

G.6.8.2 Operations

In FIA_UAU.7 Protected authentication feedback, the PP, PP-Module, functional package or ST author should specify the feedback related to the authentication process that will be provided to the user.

EXAMPLE

A feedback assignment could be “the number of characters typed”, another type of feedback is “the authentication mechanism that failed the authentication”.

G.7 User identification (FIA_UID)

G.7.1 User application notes

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.

G.7.2 FIA_UID.1 Timing of identification

G.7.2.1 Component rationale and application notes

This component poses requirements for the user to be identified. The PP, PP-Module, functional package or ST author can indicate specific actions that can be performed before the identification takes place.

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.

G.7.2.2 Operations

In FIA_UID.1.1, the PP, PP-Module, functional package or 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.

G.7.3 FIA_UID.2 User identification before any action

G.7.3.1 Component rationale and application notes

In this component users will be identified. A user is not allowed by the TSF to perform any action before being identified.

G.8 User-subject binding (FIA_USB)

G.8.1 User application notes

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.

G.8.2 FIA_USB.1 User-subject binding

G.8.2.1 Component rationale and application notes

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.

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 is those subjects that serve multiple users. In such cases the user that created this subject is assumed to be the “owner”.

EXAMPLE

An example of a user is a server process.

G.8.2.2 Operations

In FIA_USB.1.1, the PP, PP-Module, functional package or ST author should specify a list of the user security attributes that are to be bound to subjects.

In FIA_USB.1.2, the PP, PP-Module, functional package or ST author should specify any rules that are to apply upon initial association of attributes with subjects, or “none”.

In FIA_USB.1.3, the PP, PP-Module, functional package or 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”.

Annex H
(normative)

Class FMT: Security management- application notes

H.1 General information

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.

In an environment where the TOE is made up of multiple physically separated parts, 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 authorization, where the behaviour might be impaired. In such situations, use of components from Internal TOE TSF data replication consistency (FPT_TRC) is advisable.

The FMT_LIM family provides requirements that allow the specification of a policy that limits the capabilities and the availability of TSF functions. This is useful when a PP, PP-Module, functional package or ST author needs to enforce design principles such as least privilege and attack surface minimization.

Note These, and other architectural and design principles along with appropriate evaluation considerations are discussed in ISO/IEC 19249, Information technology — Security techniques — Catalogue of architectural and design principles for secure products, systems, and applications.

H.2 Limited capabilities and availability (FMT_LIM)

H.2.1 User application notes

The functional requirements FMT_LIM.1 and FMT_LIM.2 assume that there are two types of mechanisms (limitation of capabilities and limitation of availability) which together shall provide protection in order to enforce the policy. This also allows that

a) the TSF is provided without restrictions in the product in its user environment but its capabilities are so limited that the policy is enforced or conversely

b) the TSF is designed with high functionality but is removed or disabled in the product in its user environment.

The combination of both requirements shall enforce the policy.

H.2.2 FMT_LIM.1 Limited capabilities

H.2.2.1 Component rationale and application notes

An example of a limited capability JTAG interface enablement, which could be either enabled or disabled.

H.2.2.2 Operations

In FMT_LIM.1.1, the PP, PP-Module, functional package or ST author should specify the limited capability policy.

H.2.3 FMT_LIM.2 Limited availability

H.2.3.1 Component rationale and application notes

EXAMPLE

An example of a limited availability is JTAG interface enablement, which could be either enabled or disabled before operational use of the TOE.

H.2.3.2 Operations

In FMT_LIM.2.1, the PP, PP-Module, functional package or ST author should specify the limited availability policy.

H.3 Management of functions in TSF (FMT_MOF)

H.3.1 User application notes

The TSF management functions enable authorized 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 or definition and update of auditing system controls, definition and update of per-user policy attributes, definition of known system access control labels, and control and management of user groups.

   EXAMPLE

   User security characteristics: unique identifiers associated with user names, user accounts, system entry parameters

   Auditing system controls: selection of audit events, management of audit trails, audit trail analysis, and audit report generation

   User policy attributes: user clearance

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.

NOTE 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, PP-Module, functional package or ST author to ensure that adequate functions will be provided to manage the TOE in a secure fashion.

The TSF might contain functions that can be controlled by an administrator. For example, the auditing functions could be switched off, the time synchronization could be switchable, and/or the authentication mechanism could be modifiable.

H.3.2 FMT_MOF.1 Management of security functions behaviour

H.3.2.1 Component rationale and application notes

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.

EXAMPLE

modifying the behaviour of the security functions is changing of authentication mechanisms.

H.3.2.2 Operations

In FMT_MOF.1.1, the PP, PP-Module, functional package or ST author should select whether the role can determine the behaviour of, disable, enable, and/or modify the behaviour of the security functions.

In FMT_MOF.1.1, the PP, PP-Module, functional package or ST author should specify the functions that can be modified by the identified roles. Examples include auditing and time determination.

In FMT_MOF.1.1, the PP, PP-Module, functional package or 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.4 Management of security attributes (FMT_MSA)

H.4.1 User application notes

This family defines the requirements on the management of security attributes.

Security attributes affect the behaviour of the TSF.

EXAMPLE

Examples of 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, a specific authorized user (a user with explicitly given rights for this management) or inherit values according to a given policy/set of rules.

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.

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.

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.

H.4.2 FMT_MSA.1 Management of security attributes

H.4.2.1 Component rationale and application notes

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.

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.

H.4.2.2 Operations

In FMT_MSA.1.1, the PP, PP-Module, functional package or ST author should list the access control SFP(s) or the information flow control SFP(s) for which the security attributes are applicable.

In FMT_MSA.1.1, the PP, PP-Module, functional package or ST author should specify the operations that can be applied to the identified security attributes. The PP, PP-Module, functional package or 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.

In FMT_MSA.1.1, the PP, PP-Module, functional package or ST author should specify the security attributes that can be operated on by the identified roles. It is possible for the PP, PP-Module, functional package or ST author to specify that the default value such as default access-rights can be managed.

EXAMPLE 1

Examples of these security attributes are user-clearance, priority of service level, access control list, default access rights.

In FMT_MSA.1.1, the PP, PP-Module, functional package or 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.

In FMT_MSA.1.1, if selected, the PP, PP-Module, functional package or ST author should specify which other operations the role could perform.

EXAMPLE 2

An example of such an operation could be “create”.

H.4.3 FMT_MSA.2 Secure security attributes

H.4.3.1 Component rationale and application notes

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.

The definition of what “secure” means is not answered in this component but is left to the development of the TOE 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.

H.4.3.2 Operations

In FMT_MSA.2.1, the PP, PP-Module, functional package or ST author should specify the list of security attributes that require only secure values to be provided.

H.4.4 FMT_MSA.3 Static attribute initialization

H.4.4.1 Component rationale and application notes

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.

H.4.4.2 Operations

In FMT_MSA.3.1, the PP, PP-Module, functional package or ST author should list the access control SFP or the information flow control SFP for which the security attributes are applicable.

In FMT_MSA.3.1, the PP, PP-Module, functional package or ST author should select whether the default property of the access control attribute will be restrictive, permissive, or another property. Only one of these options may be chosen.

In FMT_MSA.3.1, if the PP, PP-Module, functional package or ST author selects another property, the PP, PP-Module, functional package or ST author should specify the desired characteristics of the default values.

In FMT_MSA.3.2, the PP, PP-Module, functional package or 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.4.5 FMT_MSA.4 Security attribute value inheritance

H.4.5.1 Component rationale and application notes

This component requires specification of the set of rules through which the security attribute inherits values and the conditions to be met for these rules to be applied.

H.4.5.2 Operations

In FMT_MSA.4.1, the PP, PP-Module, functional package or ST author specifies the rules governing the value that will be inherited by the specified security attribute, including the conditions that are to be met for the rules to be applied.

EXAMPLE

For example, if a new file or directory is created (in a multilevel filesystem), its label is the label at which the user is logged in at the time it is created.

H.5 Management of TSF data (FMT_MTD)

H.5.1 User application notes

This component imposes requirements on the management of TSF data. Examples of TSF data are the current time and the audit trail.

EXAMPLE

this family allows the specification of whom can read, delete, or create the audit trail.

H.5.2 FMT_MTD.1 Management of TSF data

H.5.2.1 Component rationale and application notes

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.

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.

H.5.2.2 Operations

In FMT_MTD.1.1, the PP, PP-Module, functional package or ST author should specify the operations that can be applied to the identified TSF data. The PP, PP-Module, functional package or 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, PP-Module, functional package or 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 that stores the TSF data remains in the TOE.

In FMT_MTD.1.1, the PP, PP-Module, functional package or ST author should specify the TSF data that can be operated on by the identified roles. It is possible for the PP, PP-Module, functional package or ST author to specify that the default value can be managed.

In FMT_MTD.1.1, the PP, PP-Module, functional package or 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.

In FMT_MTD.1.1, if selected, the PP, PP-Module, functional package or ST author should specify which other operations the role could perform.

EXAMPLE

An example of an operation is “create”.

H.5.3 FMT_MTD.2 Management of limits on TSF data

H.5.3.1 Component rationale and application notes

This component specifies limits on TSF data, and actions to be taken if these limits are exceeded. This component 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.

H.5.3.2 Operations

In FMT_MTD.2.1, the PP, PP-Module, functional package or 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.

In FMT_MTD.2.1, the PP, PP-Module, functional package or 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.

In FMT_MTD.2.2, the PP, PP-Module, functional package or ST author should specify the actions to be taken if the specified limit on the specified TSF data is exceeded.

EXAMPLE

An example of such a TSF action is that the authorized user is informed and an audit record is generated.

H.5.4 FMT_MTD.3 Secure TSF data

H.5.4.1 Component rationale and application notes

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.

The definition of what “secure” means is not answered in this component but is left to the development of the TOE and the resulting information in the guidance.

H.5.4.2 Operations

In FMT_MTD.3.1, the PP, PP-Module, functional package or ST author should specify what TSF data require only secure values to be accepted.

H.6 Revocation (FMT_REV)

H.6.1 User application notes

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

H.6.2 FMT_REV.1 Revocation

H.6.2.1 Component rationale and application notes

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.

H.6.2.2 Operations

In FMT_REV.1.1, the PP, PP-Module, functional package or ST author should specify which security attributes are to be revoked when a change is made to the associated object/subject/user/other resource.

In FMT_REV.1.1, the PP, PP-Module, functional package or 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.

In FMT_REV.1.1, the PP, PP-Module, functional package or 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.

In FMT_REV.1.1, the PP, PP-Module, functional package or ST author should, if additional resources is selected, specify whether the ability to revoke their security attributes shall be provided by the TSF.

In FMT_REV.1.2, the PP, PP-Module, functional package or 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.7 Security attribute expiration (FMT_SAE)

H.7.1 User application notes

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, audit attributes, etc.

EXAMPLE

An example of a certificate is key certificates such as ANSI X509.

H.7.2 FMT_SAE.1 Time-limited authorization

H.7.2.1 Operations

In FMT_SAE.1.1, the PP, PP-Module, functional package or ST author should provide the list of security attributes for which expiration is to be supported.

EXAMPLE

An example of such an attribute might be a user's security clearance.

In FMT_SAE.1.1, the PP, PP-Module, functional package or 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.

In FMT_SAE.1.2, the PP, PP-Module, functional package or 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, PP-Module, functional package or ST, the action “immediate revocation” should be specified.

H.8 Specification of Management Functions (FMT_SMF)

H.8.1 User application notes

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.

H.8.2 FMT_SMF.1 Specification of Management Functions

H.8.2.1 Component rationale and application notes

This component specifies the management functions to be provided.

PP, PP-Module, functional package or ST authors should consult the “Management” subclauses for components included in their PP, PP-Module, functional package or ST to provide a basis for the management functions to be listed via this component.

H.8.2.2 Operations

In FMT_SMF.1.1, the PP, PP-Module, functional package or 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.9 Security management roles (FMT_SMR)

H.9.1 User application notes

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.

This family requires that information be maintained to identify whether a user is authorized to use a particular security-relevant administrative function.

Some management actions can be performed by users, others only by designated people within the organization. This family allows the definition of different roles, such as owner, auditor, administrator, daily-management.

The roles as used in this family are security related roles. Each role can encompass an extensive set of capabilities or can be a single right. This family defines the roles. The capabilities of the role are defined in Limited capabilities and availability (FMT_LIM), Management of security attributes (FMT_MSA) and Management of TSF data (FMT_MTD).

EXAMPLE 1

Set of capabilities: root in UNIX

Single right: right to read a single object such as the helpfile.

Some type of roles might be mutually exclusive.

EXAMPLE 2

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.

H.9.2 FMT_SMR.1 Security roles

H.9.2.1 Component rationale and application notes

This component specifies the different roles that the TSF should recognize. Often the system distinguishes between the owner of an entity, an administrator, and other users.

H.9.2.2 Operations

In FMT_SMR.1.1, the PP, PP-Module, functional package or ST author should specify the roles that are recognized by the system. These are the roles that users could occupy with respect to security.

EXAMPLE

Examples of roles are: owner, auditor, and administrator.

H.9.3 FMT_SMR.2 Restrictions on security roles

H.9.3.1 Component rationale and application notes

This component specifies the different roles that the TSF should recognize, and conditions on how those roles could be managed. Often the system distinguishes between the owner of an entity, an administrator, and other users.

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.

H.9.3.2 Operations

In FMT_SMR.2.1, the PP, PP-Module, functional package or ST author should specify the roles that are recognized by the system. These are the roles that users could occupy with respect to security.

EXAMPLE 1

Examples of roles are: owner, auditor, and administrator.

In FMT_SMR.2.3, the PP, PP-Module, functional package or ST author should specify the conditions that govern role assignment.

EXAMPLE2

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”.

H.9.4 FMT_SMR.3 Assuming roles

H.9.4.1 Component rationale and application notes

This component specifies that an explicit request must be given to assume the specific role.

H.9.4.2 Operations

In FMT_SMR.3.1, the PP, PP-Module, functional package or ST author should specify the roles that require an explicit request to be assumed.

EXAMPLE

Examples of roles are: owner, auditor, and administrator.

Annex I
(normative)

Class FPR: Privacy- application notes

I.1 General Information

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.

In the components of this class there is flexibility as to whether or not authorized users are covered by the required security functionality.

EXAMPLE 1

a PP, PP-Module, functional package or ST author might consider it appropriate not to require protection of the privacy of users against a suitably authorized user.

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.

EXAMPLE 2

If authorized users are not allowed to see the user identity (perhaps because of 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, PP-Module, functional package or ST, where the fact that a particular security relevant event has occurred is more important than knowing who was responsible for it.

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.

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.

EXAMPLE 3

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.

All families assume that a user does not explicitly perform an action that discloses the user's own identity.

EXAMPLE 4

The TSF is not expected to screen the user name in electronic messages or databases.

All families in this class have components that are scoped through operations. These operations allow the PP, PP-Module, functional package or ST author to state the cooperating users/subjects to which the TSF must be resistant.

EXAMPLE 5

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”.

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 editors' have added the following note since it seemed pertinent in this section of the application notes. If no comments to the contrary are made then this note will remain in the next draft.

NOTE The reader's attention is drawn to ISO/IEC TS 19608:2018 Guidance for developing security and privacy functional requirements based on ISO/IEC 15408. ISO/IEC TS 19608:2018 provides guidance for:

- selecting and specifying security functional requirements (SFRs) from ISO/IEC 15408-2 to protect Personally Identifiable Information (PII);
- the procedure to define both privacy and security functional requirements in a coordinated manner; and
- developing privacy functional requirements as extended components based on the privacy principles defined in ISO/IEC 29100 through the paradigm described in ISO/IEC 15408-2.

I.2 Anonymity (FPR_ANO)

I.2.1 User application notes

Anonymity ensures that a subject may use a resource or service without disclosing its user identity.

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, PP-Module, functional package or ST author with a means to identify the set of users that cannot see the identity of someone performing certain actions.

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 user's identity, not on the protection of the subject identity; hence, the identity of the subject is not protected from disclosure.

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.

The interpretation of "determine" should be taken in the broadest sense of the word.

The Components leveling and description distinguishes between the users and an authorized user. An authorized user is often excluded from the component, and therefore allowed to retrieve a user's identity. However, there is no specific requirement that an authorized 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 authorized user can see the identity of anyone performing any action.

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, PP-Module, functional package or ST author wants to address all subjects/operations, the words "all subjects and all operations" could be provided.

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.

EXAMPLE

Potential hostile users or subjects include providers, system operators, communication partners and users, who smuggle malicious parts (including malware) into systems. All of these users can investigate usage patterns (such as which users used which services) and misuse this information.

I.2.2 FPR_ANO.1 Anonymity

I.2.2.1 Component rationale and application notes

This component ensures that the identity of a user is protected from disclosure. There may be instances, however, that a given authorized user can determine who performed certain actions. This component gives the flexibility to capture either a limited or total privacy policy.

I.2.2.2 Operations

In FPR_ANO.1.1, the PP, PP-Module, functional package or ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP, PP-Module, functional package or 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.

EXAMPLE 1

A set of users could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_ANO.1.1, the PP, PP-Module, functional package or 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.

EXAMPLE 2

An example of an object is “the voting application”.

I.2.3 FPR_ANO.2 Anonymity without soliciting information

I.2.3.1 Component rationale and application notes

This component is used to ensure that the TSF is not allowed to know the identity of the user.

I.2.3.2 Operations

In FPR_ANO.2.1, the PP, PP-Module, functional package or ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP, PP-Module, functional package or 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.

EXAMPLE 1

A set of users could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_ANO.2.1, the PP, PP-Module, functional package or 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.

EXAMPLE 2

“the voting application”.

In FPR_ANO.2.2, the PP, PP-Module, functional package or ST author should identify the list of services which are subject to the anonymity requirement, for example, “the accessing of job descriptions”.

In FPR_ANO.2.2, the PP, PP-Module, functional package or 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.3 Pseudonymity (FPR_PSE)

I.3.1 User application notes

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.

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.

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.

A way to use the reference is by being able to obtain the original user identity.

EXAMPLE 1

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, PP-Module, functional package or ST author might want to incorporate FPR_PSE.2 Reversible pseudonymity to describe those services.

Another usage of the reference is as an alias for a user.

EXAMPLE 2

A user who does not wish to be identified, can provide an account to which the resource utilization 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, PP-Module, functional package or ST author might wish to incorporate FPR_PSE.3 Alias pseudonymity to specify the rules to which the reference must conform.

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.

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).

It should be realized 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.

The intent is that the TSF not reveal any information that would compromise the identity of the user.

EXAMPLE 3

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.

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.

EXAMPLE 4

Potential hostile users include providers, system operators, communication partners and users, who smuggle malicious parts (including malware) 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 authorization without identification, especially for anonymous payment ("Digital Cash"). This helps providers to obtain their payment in a secure way while maintaining customer anonymity.

I.3.2 FPR_PSE.1 Pseudonymity

I.3.2.1 Component rationale and application notes

This component provides the user protection against disclosure of identity to other users. The user will remain accountable for its actions.

I.3.2.2 Operations

In FPR_PSE.1.1, the PP, PP-Module, functional package or ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP, PP-Module, functional package or 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.

EXAMPLE 1

A set of users could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_PSE.1.1, the PP, PP-Module, functional package or 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.

EXAMPLE 2

"the accessing of job offers".

Note "objects" includes any other attributes that might enable another user or subject to derive the actual identity of the user.

In FPR_PSE.1.2, the PP, PP-Module, functional package or ST author should identify the (one or more) number of aliases the TSF is able to provide.

In FPR_PSE.1.2, the PP, PP-Module, functional package or ST author should identify the list of subjects to whom the TSF is able to provide an alias.

In FPR_PSE.1.3, the PP, PP-Module, functional package or 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.

In FPR_PSE.1.3, the PP, PP-Module, functional package or ST author should identify the metric to which the TSF-generated or user-generated alias should conform.

I.3.3 FPR_PSE.2 Reversible pseudonymity

I.3.3.1 Component rationale and application notes

In this component, the TSF shall ensure that under specified conditions the user identity related to a provided reference can be determined.

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.

EXAMPLE

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."

I.3.3.2 Operations

In FPR_PSE.2.1, the PP, PP-Module, functional package or ST author should specify the set of users and/or subjects against which the TSF must provide protection.

EXAMPLE 1

Even if the PP, PP-Module, functional package or 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).

In FPR_PSE.2.1, the PP, PP-Module, functional package or 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.

EXAMPLE 2

"The accessing of job offers".

NOTE "objects" includes any other attributes that might enable another user or subject to derive the actual identity of the user.

In FPR_PSE.2.2, the PP, PP-Module, functional package or ST author should identify the (one or more) number of aliases the TSF, is able to provide.

In FPR_PSE.2.2, the PP, PP-Module, functional package or ST author should identify the list of subjects to whom the TSF is able to provide an alias.

In FPR_PSE.2.3, the PP, PP-Module, functional package or 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.

In FPR_PSE.2.3, the PP, PP-Module, functional package or ST author should identify the metric to which the TSF-generated or user-generated alias should conform.

In FPR_PSE.2.4, the PP, PP-Module, functional package or ST author should select whether the authorized user and/or trusted subjects can determine the real user name.

In FPR_PSE.2.4, the PP, PP-Module, functional package or ST author should identify the list of conditions under which the trusted subjects and authorized 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.

In FPR_PSE.2.4, the PP, PP-Module, functional package or ST author should identify the list of trusted subjects that can obtain the real user name under a specified condition.

EXAMPLE

A notary or special authorized user.

I.3.4 FPR_PSE.3 Alias pseudonymity

I.3.4.1 Component rationale and application notes

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.

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.

I.3.4.2 Operations

In FPR_PSE.3.1, the PP, PP-Module, functional package or ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP, PP-Module, functional package or 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.

EXAMPLE 1

A set of users could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_PSE.3.1, the PP, PP-Module, functional package or 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.

EXAMPLE 2

"the accessing of job offers".

NOTE "objects" includes any other attributes which might enable another user or subject to derive the actual identity of the user.

In FPR_PSE.3.2, the PP, PP-Module, functional package or ST author should identify the (one or more) number of aliases the TSF is able to provide.

In FPR_PSE.3.2, the PP, PP-Module, functional package or ST author should identify the list of subjects to whom the TSF is able to provide an alias.

In FPR_PSE.3.3, the PP, PP-Module, functional package or 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.

In FPR_PSE.3.3, the PP, PP-Module, functional package or ST author should identify the metric to which the TSF-generated or user-generated alias should conform.

In FPR_PSE.3.4, the PP, PP-Module, functional package or 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.4 Unlinkability (FPR_UNL)

I.4.1 User application notes

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.

The requirements for unlinkability are intended to protect the user identity against the use of profiling of the operations.

EXAMPLE 1

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.

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.

Unlinkability requires that different operations cannot be related. This relationship can take several forms.

EXAMPLE 2

The user associated with the operation, or the terminal which initiated the action, or the time the action was executed.

The PP, PP-Module, functional package or ST author can specify what kind of relationships are present that must be countered.

Possible applications include the ability to make multiple use of a pseudonym without creating a usage pattern that might disclose the user's identity.

EXAMPLE 3

Potential hostile subjects and users include providers, system operators, communication partners and users, who smuggle malicious parts, (including malware) into systems, they do not operate but want to get information about. All of these attackers can investigate (such as which users used which services) and misuse this information.

Unlinkability protects users from linkages, which could be drawn between several actions of a customer.

EXAMPLE 4

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.

I.4.2 FPR_UNL.1 Unlinkability

I.4.2.1 Component rationale and application notes

This component ensures that users cannot link different operations in the system and thereby obtain information.

I.4.2.2 Operations

In FPR_UNL.1.1, the PP, PP-Module, functional package or ST author should specify the set of users and/or subjects against which the TSF must provide protection.

EXAMPLE 1

Even if the PP, PP-Module, functional package or 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.

EXAMPLE 2

A set of users could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_UNL.1.1, the PP, PP-Module, functional package or ST author should identify the list of operations which should be subjected to the unlinkability requirement.

EXAMPLE 3

"Sending email".

In FPR_UNL.1.1, the PP, PP-Module, functional package or 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.

In FPR_UNL.1.1, the PP, PP-Module, functional package or ST author should identify the list of relations which should be protected against.

EXAMPLE

"Originate from the same IP address".

I.5 Unobservability (FPR_UNO)

I.5.1 User application notes

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.

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.

A number of techniques can be applied to implement unobservability.

EXAMPLE

Examples of techniques to provide unobservability are:

a) Allocation of information impacting unobservability: Unobservability relevant information (such as. 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 (such as Internet and Radio frequencies, including Ethernet, Bluetooth, WiFi and Near-field communication bands), 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 (such as 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.

Sometimes, users should not see the use of a resource, but an authorized user must be allowed to see the use of the resource in order to perform his duties. In such cases, the FPR_UNO.4 Authorized user observability could be used, which provides the capability for one or more authorized users to see the usage.

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.

Unobservability of communications may be an important factor in many areas, such as the enforcement of constitutional rights, organizational policies, or in defense related applications.

I.5.2 FPR_UNO.1 Unobservability

I.5.2.1 Component rationale and application notes

This component requires that the use of a function or resource cannot be observed by unauthorized users.

I.5.2.2 Operations

In FPR_UNO.1.1, the PP, PP-Module, functional package or ST author should specify the list of users and/or subjects against which the TSF must provide protection.

EXAMPLE 1

Even if the PP, PP-Module, functional package or 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.

EXAMPLE 2

A set of users could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_UNO.1.1, the PP, PP-Module, functional package or 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.

EXAMPLE 3

Reading and writing to the object.

In FPR_UNO.1.1, the PP, PP-Module, functional package or ST author should identify the list of objects which are covered by the unobservability requirement.

EXAMPLE 4

A specific mail server or ftp site.

In FPR_UNO.1.1, the PP, PP-Module, functional package or ST author should specify the set of protected users and/or subjects whose unobservability information will be protected.

EXAMPLE 5

“Users accessing the system through the internet”.

I.5.3 FPR_UNO.2 Allocation of information impacting unobservability

I.5.3.1 Component rationale and application notes

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.

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.

EXAMPLE

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).

I.5.3.2 Operations

In FPR_UNO.2.1, the PP, PP-Module, functional package or ST author should specify the list of users and/or subjects against which the TSF must provide protection. For example, even if the PP, PP-Module, functional package or 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.

EXAMPLE 1

A set of users could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_UNO.2.1, the PP, PP-Module, functional package or 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

EXAMPLE 2

Reading and writing to the object.

In FPR_UNO.2.1, the PP, PP-Module, functional package or 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.

In FPR_UNO.2.1, the PP, PP-Module, functional package or ST author should specify the set of protected users and/or subjects whose unobservability information will be protected.

EXAMPLE 3

“Users accessing the system through the internet”.

In FPR_UNO.2.2, the PP, PP-Module, functional package or ST author should identify which privacy related information should be distributed in a controlled manner.

EXAMPLE 4

This information could include: IP address of subject, IP address of object, time, used encryption keys.

In FPR_UNO.2.2, the PP, PP-Module, functional package or 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.

EXAMPLE 5

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”.

I.5.4 FPR_UNO.3 Unobservability without soliciting information

I.5.4.1 Component rationale and application notes

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.

I.5.4.2 Operations

In FPR_UNO.3.1, the PP, PP-Module, functional package or ST author should identify the list of services which are subject to the unobservability requirement.

EXAMPLE 1

"The accessing of job descriptions".

In FPR_UNO.3.1, the PP, PP-Module, functional package or ST author should identify the list of subjects from which privacy related information should be protected when the specified services are provided.

In FPR_UNO.3.1, the PP, PP-Module, functional package or ST author should specify the privacy related information that will be protected from the specified subjects.

EXAMPLE 2

Examples of privacy related information include the identity of the subject that used a service and the quantity of a service that has been used such as memory resource utilization.

I.5.5 FPR_UNO.4 Authorized user observability

I.5.5.1 Component rationale and application notes

This component is used to require that there will be one or more authorized users with the rights to view the resource utilization. Without this component, this review is allowed, but not mandated.

I.5.5.2 Operations

In FPR_UNO.4.1, the PP, PP-Module, functional package or ST author should specify the set of authorized users for which the TSF must provide the capability to observe the resource utilization. A set of authorized users, for example, could be a group of authorized users which can operate under the same role or can all use the same process(es).

In FPR_UNO.4.1, the PP, PP-Module, functional package or ST author should specify the set of resources and/or services that the authorized user must be able to observe.

Annex J
(normative)

Class FPT: Protection of the TSF- application notes

J.1 General information

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF 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.

From the point of view of this class, regarding to the TSF there are three significant elements:

a) The TSF's implementation, which executes and implements the mechanisms that enforce the SFRs.

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

c) The external entities that the TSF may interact with in order to enforce the SFRs.

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) TOE emanation (FPT_EMS), which addresses potential leakage of information from the TOE via emanations.

b) 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.

c) TSF initialization (FPT_INI), which addresses the initialization of the TOE into a correct and secure operational state.

d) Internal TOE TSF data transfer (FPT_ITT), which addresses protection of TSF data when it is transmitted between physically-separated parts of the TOE.

e) TSF physical protection (FPT_PHP), which provides an authorized user with the ability to detect external attacks on the parts of the TOE that comprise the TSF.

f) 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 another trusted IT product.

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 synchronization 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 another trusted IT product.

k) Testing of external entities (FPT_TEE) and TSF self-test (FPT_TST), which provide an authorized user with the ability to verify the correct operation of the external entities interacting with the TSF to enforce the SFRs, and the integrity of the TSF data and TSF itself.

J.2 FPT_EMS TOE emanation

J.2.1 User application notes

This family defines the requirements for the TOE to be able to prevent or mitigate attacks against data stored in and used by the TOE where the attack is based on external observable physical phenomena of the TOE.

EXAMPLE

Examples of such attacks are analysis of TOE's electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc.

FPT_EMS.1.1 Limit of Emissions requires the TOE to not emit intelligible emissions enabling access to TSF data or user data.

J.2.2 FPT_EMS.1 TOE emanation

J.2.2.1 Component rationale and application notes

FPT_EMS.1.1 Table found as part of the FPT_EMS.1.1 Limit of Emissions element shall be completed by the PP, PP-Module, functional package or ST author. Each row, which can be identified using the "Identifier", provides a set of assignments for completing the SFR, allowing the PP, PP-Module, functional package or ST author to specify the requirements for TOE emanation protection for various different combinations of emissions, interfaces, TSF data and user data.

EXAMPLE

Types of emission can include audio frequencies and radio frequencies.

Types of interfaces can include physical ports, I.C. boundaries, and electronic components.

J.3 Fail secure (FPT_FLS)

J.3.1 User application notes

The requirements of this family ensure that the TOE will always enforce its SFRs in the event of certain types of failures in the TSF.

J.3.2 FPT_FLS.1 Failure with preservation of secure state

J.3.2.1 Component rationale and application notes

The term "secure state" refers to a state in which the TSF data are consistent and the TSF continues correct enforcement of the SFRs.

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, PP-Module, functional package or ST author should specify those situations in which audit is desired and feasible.

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 initialization or resetting of the TSF.

J.3.2.2 Operations

In FPT_FLS.1.1, the PP, PP-Module, functional package or 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 SFRs.

J.4 TSF initialization (FPT_INI)

J.4.1 User application notes

This family defines the functional requirements for the initialization of the TSF. By a dedicated function of the TOE that ensures that the initialization of the TSF results in a correct and secure operational state. This can cover code/data that are stored and executed from non-modifiable memory at boot time, the immutable root-of-trust, and other one-time programmable (OTP) values such as versions and identifiers.

J.4.2 FPT_INI.1 TSF initialization

J.4.2.1 Operations

In FPT_INI.1.2 the PP, PP-Module, functional package or ST author should list the properties and the elements to which they apply, using the assignment table format in the element.

EXAMPLE

Properties could include authenticity, integrity, correct version and elements to which the properties apply could include TSF or user firmware, software or data.

In FPT_INI.1.3 the PP, PP-Module, functional package or ST author uses the selections and assignments to describe the behaviour of the TOE initialization function in the case that errors or other failures are encountered during the initialization.

FPT_INI.1.4 the PP, PP-Module, functional package or ST author uses the assignment to describe the methods by which the TOE initialization function interacts with the TSF.

J.5 Availability of exported TSF data (FPT_ITA)

J.5.1 User application notes

This family defines the rules for the prevention of loss of availability of TSF data moving between the TSF and another trusted IT product. This data could be TSF critical data such as passwords, keys, audit data, or TSF executable code.

This family is used in a distributed context where the TSF is providing TSF data to another 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.

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.

J.5.2 FPT_ITA.1 Inter-TSF availability within a defined availability metric

J.5.2.1 Operations

In FPT_ITA.1.1, the PP, PP-Module, functional package or ST author should specify the types of TSF data that are subject to the availability metric.

In FPT_ITA.1.1, the PP, PP-Module, functional package or ST should specify the availability metric for the applicable TSF data.

In FPT_ITA.1.1, the PP, PP-Module, functional package or ST author should specify the conditions under which availability must be ensured.

EXAMPLE

There must be a connection between the TOE and another trusted IT product.

J.6 Confidentiality of exported TSF data (FPT_ITC)

J.6.1 User application notes

This family defines the rules for the protection from unauthorized disclosure of TSF data moving between the TSF and another trusted IT product.

EXAMPLE

Examples of this data are TSF critical data such as passwords, keys, audit data, or TSF executable code.

This family is used in a distributed context where the TSF is providing TSF data to another 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.

J.6.2 Evaluator notes

Confidentiality of TSF Data during transmission is necessary to protect such information from disclosure.

EXAMPLE

Some possible implementations that could provide confidentiality include the use of cryptographic algorithms as well as spread spectrum techniques.

J.6.3 FPT_ITC.1 Inter-TSF confidentiality during transmission

J.6.3.1 Component rationale and application notes

This component is used when it is necessary to make the requirement for confidentiality of TSF data when being transmitted from the TSF to another trusted IT product.

J.7 Integrity of exported TSF data (FPT_ITI)

J.7.1 User application notes

This family defines the rules for the protection, from unauthorized modification, of TSF data during transmission between the TSF and another trusted IT product.

EXAMPLE

Examples of this data are TSF critical data such as passwords, keys, audit data, or TSF executable code.

This family is used in a distributed context where the TSF is exchanging TSF data with another trusted IT product. Note that a requirement that addresses modification, detection, or recovery at another trusted IT product cannot be specified, as the mechanisms that another 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 another trusted IT product can use.

J.7.1.1 Evaluator notes

In the FPT_ITI.2 component some possible means of satisfying this requirement involves the use of cryptographic functions or some form of checksum.

J.7.2 FPT_ITI.1 Inter-TSF detection of modification

J.7.2.1 Component rationale and application notes

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 another trusted IT product can request the TOE's TSF to retransmit data when modification has been detected or respond to such types of request.

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.

J.7.2.2 Operations

In FPT_ITI.1.1, the PP, PP-Module, functional package or ST should specify the modification metric that the detection mechanism must satisfy. This modification metric shall specify the desired strength of the modification detection.

In FPT_ITI.1.2, the PP, PP-Module, functional package or 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”.

J.7.3 FPT_ITI.2 Inter-TSF detection and correction of modification

J.7.3.1 Component rationale and application notes

This component should be used in situations where it is necessary to detect or correct modifications of TSF critical data.

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 or to mechanisms that are well known in the public literature.

EXAMPLE

Attack reference: “only 1 in 1000 random messages will be accepted”.

Well known mechanism: “the strength must be conformant to the strength offered by Secure Hash Algorithm”.

The approach taken to correct modification might be done through some form of error correcting checksum.

J.7.3.2 Operations

In FPT_ITI.2.1, the PP, PP-Module, functional package or ST should specify the modification metric that the detection mechanism must satisfy. This modification metric shall specify the desired strength of the modification detection.

In FPT_ITI.2.2, the PP, PP-Module, functional package or ST should specify the actions to be taken if a modification of TSF data has been detected.

EXAMPLE

An example of an action is: “ignore the TSF data and request the originating trusted product to send the TSF data again”.

In FPT_ITI.2.3, the PP, PP-Module, functional package or ST author should define the types of modification from which the TSF should be capable of recovering.

J.8 Internal TOE TSF data transfer (FPT_ITT)

J.8.1 User application notes

This family provides requirements that address protection of TSF data when it is transferred between separate parts of a TOE across an internal channel.

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.

J.8.2 Evaluator notes

One practical mechanism available to a TSF to provide this protection is cryptographically-based.

J.8.3 FPT_ITT.1 Basic internal TSF data transfer protection

J.8.3.1 Operations

In FPT_ITT.1.1, the PP, PP-Module, functional package or ST author should specify the desired type of protection to be provided from the choices: disclosure, modification.

J.8.4 FPT_ITT.2 TSF data transfer separation

J.8.4.1 Component rationale and application notes

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.

J.8.4.2 Operations

In FPT_ITT.2.1, the PP, PP-Module, functional package or ST author should specify the desired type of protection to be provided from the choices: disclosure, modification.

J.8.5 FPT_ITT.3 TSF data integrity monitoring

J.8.5.1 Operations

In FPT_ITT.3.1, the PP, PP-Module, functional package or ST author should specify the desired type of modification that the TSF shall be able to detect. The PP, PP-Module, functional package or ST author should select from: modification of data, substitution of data, re-ordering of data, deletion of data, or any other integrity errors.

In FPT_ITT.3.1, if the PP, PP-Module, functional package or 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.

In FPT_ITT.3.2, the PP, PP-Module, functional package or ST author should specify the action to be taken when an integrity error is identified.

J.9 TSF physical protection (FPT_PHP)

J.9.1 User application notes

TSF physical protection components refer to restrictions on unauthorized physical access to the TSF, and to the deterrence of, and resistance to, unauthorized physical modification, or substitution of the TSF.

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.

EXAMPLE 1

Examples of physical tampering scenarios include mechanical attack, radiation, changing the temperature.

It is acceptable for the functions that are available to an authorized 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 authorized users. As the TSF may not be “operational” during those modes, it may not be able to provide normal enforcement for authorized user access. The physical implementation of a TOE might consist of several structures. 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.

EXAMPLE 2

Examples of structures include an outer shielding, cards, and chips.

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. Nevertheless, a PP, PP-Module, functional package or 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, PP-Module, functional package or ST author should include appropriate requirements in the list of audit events.

NOTE Inclusion of these requirements may have implications on the hardware design and its interface to the software.

EXAMPLE 3

Examples of a hardware-based detection system is one based on breaking a circuit and lighting a light emitting diode (LED) if the circuit is broken when a button is pressed by the authorized user.

J.9.2 FPT_PHP.1 Passive detection of physical attack

J.9.2.1 Component rationale and application notes

FPT_PHP.1 Passive detection of physical attack should be used when threats from unauthorized 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 authorized 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_LIM.1 should be considered to specify who can make use of that capability, and how they can make use of that capability. If this is done by non-IT mechanisms such as physical inspection. management functions are not required.

J.9.3 FPT_PHP.2 Notification of physical attack

J.9.3.1 Component rationale and application notes

FPT_PHP.2 Notification of physical attack should be used when threats from unauthorized 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. 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.

J.9.3.2 Operations

In FPT_PHP.2.3, the PP, PP-Module, functional package or ST author should provide a list of TSF devices/elements for which active detection of physical tampering is required.

In FPT_PHP.2.3, the PP, PP-Module, functional package or 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_LIM.1 family) included in the PP, PP-Module, functional package or ST.

J.9.4 FPT_PHP.3 Resistance to physical attack

J.9.4.1 Component rationale and application notes

For some forms of tampering, it is necessary that the TSF not only detects the tampering, but actually resists it or delays the attacker.

This component should be used when TSF devices and TSF elements are expected to operate in an environment where a physical tampering of the internals of a TSF device or TSF element itself is a threat.

EXAMPLE

Physical tampering includes observation, analysis, or modification.

J.9.4.2 Operations

In FPT_PHP.3.1, the PP, PP-Module, functional package or 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 sub setting 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.

EXAMPLE

An example of policy protection:

with a confidentiality policy, it would be acceptable to physically disable the device so that the protected information may not be retrieved.

In FPT_PHP.3.1, the PP, PP-Module, functional package or 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.10 Trusted recovery (FPT_RCV)

J.10.1 User application notes

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.

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.

EXAMPLE

Failures that must be generally anticipated include the following:

a) Unmaskable action failures that always result in a system crash (such as 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 (such as parity errors, disk head crash, persistent read/write failure caused by misaligned disk heads, worn-out magnetic coating, dust on the disk surface, loss of Internet connection).

c) Discontinuity of operation caused by erroneous administrative action or lack of timely administrative action (such as unexpected shutdowns by turning off power, ignoring the exhaustion of critical resources, inadequate installed configuration).

NOTE 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.

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.

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 authorized 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 SFRs.

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.

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.

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 self-testing component may be argued away.

J.10.1.1 Evaluator notes

In FPT_RCV.1, it is acceptable for the functions that are available to an authorized user for trusted recovery to be available only in a maintenance mode. Controls should be in place to limit access during maintenance to authorized users.

In FPT_RCV.2 It is acceptable for the functions that are available to an authorized user for trusted recovery to be available only in a maintenance mode. Controls should be in place to limit access during maintenance to authorized users.

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.

It is assumed that the robustness of the automated recovery mechanisms will be verified.

In FPT_RCV.3 It is acceptable for the functions that are available to an authorized user for trusted recovery to be available only in a maintenance mode. Controls should be in place to limit access during maintenance to authorized users.

It is assumed that the evaluators will verify the robustness of the automated recovery mechanisms.

J.10.2 FPT_RCV.1 Manual recovery

J.10.2.1 Component rationale and application notes

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.

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.

J.10.2.2 Operations

In FPT_RCV.1.1, the PP, PP-Module, functional package or ST author should specify the list of failures or service discontinuities following which the TOE will enter a maintenance mode.

EXAMPLE

Power failure, audit storage exhaustion, any failure or discontinuity.

J.10.3 FPT_RCV.2 Automated recovery

J.10.3.1 Component rationale and application notes

Automated recovery is considered to be more useful than manual recovery, as it allows the machine to operate in an unattended fashion.

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.

J.10.3.2 Operations

In FPT_RCV.2.1, the PP, PP-Module, functional package or ST author should specify the list of failures or service discontinuities following which the TOE will need to enter a maintenance mode.

EXAMPLE

Power failure, audit storage exhaustion.

In FPT_RCV.2.2, the PP, PP-Module, functional package or ST author should specify the list of failures or other discontinuities for which automated recovery must be possible.

J.10.4 FPT_RCV.3 Automated recovery without undue loss

J.10.4.1 Component rationale and application notes

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.

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 under the control of the TSF. 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.

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 under the control of the TSF.

J.10.4.2 Operations

In FPT_RCV.3.1, the PP, PP-Module, functional package or ST author should specify the list of failures or service discontinuities following which the TOE will need to enter a maintenance mode.

EXAMPLE

Power failure, audit storage exhaustion.

In FPT_RCV.3.2, the PP, PP-Module, functional package or ST author should specify the list of failures or other discontinuities for which automated recovery must be possible.

In FPT_RCV.3.3, the PP, PP-Module, functional package or ST author should provide a quantification for the amount of loss of TSF data or objects that is acceptable.

J.10.5 FPT_RCV.4 Function recovery

J.10.5.1 Component rationale and application notes

Function recovery requires that if there should be some failure in the TSF, that certain functions in the TSF should either complete successfully or recover to a secure state.

J.10.5.2 Operations

In FPT_RCV.4.1, the PP, PP-Module, functional package or ST author should specify a list the functions and failure scenarios. In the event that any of the identified failure scenarios happen, the functions that have been specified must either complete successfully or recover to a consistent and secure state.

J.11 Replay detection (FPT_RPL)

J.11.1 User application notes

This family addresses detection of replay for various types of entities and subsequent actions to correct.

J.11.2 FPT_RPL.1 Replay detection

J.11.2.1 Component rationale and application notes

The entities included here are those that can be involved in replay detection.

EXAMPLE

Messages, service requests, service responses, or sessions.

J.11.2.2 Operations

In FPT_RPL.1.1, the PP, PP-Module, functional package or ST author should provide a list of identified entities for which detection of replay should be possible.

EXAMPLE

Messages, service requests, service responses, and user sessions.

In FPT_RPL.1.2, the PP, PP-Module, functional package or 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.12 State synchrony protocol (FPT_SSP)

J.12.1 User application notes

Distributed TOEs may give rise to greater complexity than monolithic TOEs through the potential for differences in state between parts of the TOE, and through delays in communication. In most cases, synchronization 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.

State synchrony protocol (FPT_SSP) establishes the requirement for certain critical functions of the TSF to use a trusted protocol. State synchrony protocol (FPT_SSP) ensures that two distributed parts of the TOE, such as hosts, have synchronized their states after a security-relevant action.

Some states may never be synchronized, or the transaction cost may be too high for practical use.

EXAMPLE 1

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 TOEs. Indeterminacy and state synchrony are related, and the same solution may apply. It is futile to design for indeterminate states; the PP, PP-Module, functional package or ST author should express other requirements in such cases.

EXAMPLE 2

Raise an alarm, audit the event.

J.12.2 FPT_SSP.1 Simple trusted acknowledgement

J.12.2.1 Component rationale and application notes

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.

J.12.3 FPT_SSP.2 Mutual trusted acknowledgement

J.12.3.1 Component rationale and application notes

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.

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)

J.13.1 User application notes

This family addresses requirements for a reliable time stamp function within a TOE.

It is the responsibility of the PP, PP-Module, functional package or 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.

J.13.2 FPT_STM.1 Reliable time stamps

J.13.2.1 Component rationale and application notes

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)

J.14.1 User application notes

In a distributed or composite environment, a TOE may need to exchange TSF data with another trusted IT Product.

EXAMPLE

the SFP-attributes associated with data, audit information, identification information.

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.

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.

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.

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.

J.14.2 FPT_TDC.1 Inter-TSF basic TSF data consistency

J.14.2.1 Component rationale and application notes

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.

EXAMPLE

The TSF data of two different systems may have different conventions internally. For the TSF data to be used properly (such as 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.

J.14.2.2 Operations

In FPT_TDC.1.1, the PP, PP-Module, functional package or 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.

In FPT_TDC.1.2, the PP, PP-Module, functional package or ST should assign the list of interpretation rules to be applied by the TSF.

J.15 Testing of external entities (FPT_TEE)

J.15.1 User application notes

This family defines requirements for the testing of one or more external entities by the TSF. These external entities are not human users, and they can include combinations of software and/or hardware interacting with the TOE.

EXAMPLE

Examples of the types of tests that may be run are:

a) tests for the presence of a firewall, and possibly whether it is correctly configured;

b) tests of some of the properties of the operating system that an application TOE runs on;

c) tests of some of the properties of the IC that a smart card OS TOE runs on (such as the random number generator).

Note The external entity may “lie” about the test results, either on purpose or because it is not working correctly.

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 also in this family.

J.15.2 Evaluator notes

The tests of external entities should be sufficient to test all of the characteristics of them upon which the TSF relies.

For FPT_TEE.1 Testing of external entities, 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 authorized users.

J.15.3 FPT_TEE.1 Testing of external entities

J.15.3.1 Component rationale and application notes

This component is not intended to be applied to human users.

This component provides support for the periodic testing of properties related to external entities upon which the TSF's operation depends, by requiring the ability to periodically invoke testing functions.

The PP, PP-Module, functional package or ST author may refine the requirement to state whether the function should be available in off-line, on-line or maintenance mode.

J.15.3.2 Operations

In FPT_TEE.1.1, the PP, PP-Module, functional package or ST author should specify when the TSF will run the testing of external entities, during initial start-up, periodically during normal operation, at the request of an authorized user, or under other conditions. 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, the testing of external entities may delay the normal operation of a TOE.

In FPT_TEE.1.1, the PP, PP-Module, functional package or ST author should specify the properties of the external entities to be checked by the tests.

EXAMPLE 1

Examples of these properties may include configuration or availability properties of a directory server supporting some access control part of the TSF.

In FPT_TEE.1.1, the PP, PP-Module, functional package or ST author should, if other conditions are selected, specify the frequency with which the testing of external entities will be run.

EXAMPLE 2

An example of this other frequency or condition may be to run the tests each time a user requests to initiate a session with the TOE. For instance, this could be the case of testing a directory server before its interaction with the TSF during the user authentication process.

In FPT_TEE.1.2, the PP, PP-Module, functional package or ST author should specify what are the action(s) that the TSF shall perform when the testing fails.

EXAMPLE 3

Examples of these action(s), illustrated by a directory server instance, may include to connect to an alternative available server or otherwise to look for a backup server.

J.16 Internal TOE TSF data replication consistency (FPT_TRC)

J.16.1 User application notes

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.

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 synchronization protocol. If a particular protocol is necessary for a PP, PP-Module, functional package or ST, it can be specified through refinement.

It can be impossible to synchronize some states, or the cost of such synchronization can be too high.

EXAMPLE

Examples of this situation are communication channel and encryption key revocations.

Indeterminate states can also occur; if a specific behaviour is desired, it should be specified via refinement.

J.16.2 FPT_TRC.1 Internal TSF consistency

J.16.2.1 Operations

In FPT_TRC.1.2, the PP, PP-Module, functional package or ST author should specify the list of functions dependent on TSF data replication consistency.

J.17 TSF self-test (FPT_TST)

J.17.1 User application notes

The family defines the requirements for the self-testing of the TSF with respect to some expected correct operation.

EXAMPLE

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 authorized 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.

The requirements of this family are also needed to detect the corruption of TSF data and TSF itself (i.e. TSF executable code or TSF hardware component) 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.

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.

The term “correct operation of the TSF” refers primarily to the operation of the TSF and the integrity of the TSF data.

J.17.2 Evaluator notes

For FPT_TST.1 TSF testing, it is acceptable for the functions that are available to the authorized 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 authorized users.

J.17.3 FPT_TST.1 TSF testing

J.17.3.1 Component rationale and application notes

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.

J.17.3.2 Operations

In FPT_TST.1.1, the PP, PP-Module, functional package or 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 authorized user, at other conditions. In the case of the latter option, the PP, PP-Module, functional package or ST author should also assign what those conditions are via the following assignment.

In FPT_TST.1.1, the PP, PP-Module, functional package or 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.

In FPT_TST.1.1, the PP, PP-Module, functional package or ST author should, if selected, specify the conditions under which the self-test should take place.

In FPT_TST.1.1, the PP, PP-Module, functional package or ST author should, if selected, specify the list of parts of the TSF that will be subject to TSF self-testing.

In FPT_TST.1., the PP, PP-Module, functional package or ST author should specify whether data integrity is to be verified for all TSF data, or only for selected data.

In FPT_TST.1., the PP, PP-Module, functional package or ST author should, if selected, specify the list of TSF data that will be verified for integrity.

In FPT_TST.1., the PP, PP-Module, functional package or ST author should specify whether TSF integrity is to be verified for all TSF, or only for selected TSF.

In FPT_TST.1., the PP, PP-Module, functional package or ST author should, if selected, specify the list of TSF that will be verified for integrity.

Annex K
(normative)

Class FRU: Resource utilization- application notes

K.1 General information

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 monopolized by lower priority tasks. The family Resource Allocation provides limits on the use of available resources, therefore preventing users from monopolizing the resources.

K.2 Fault tolerance (FRU_FLT)

K.2.1 User application notes

This family provides requirements for the availability of capabilities even in the case of failures.

EXAMPLE 1

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.

EXAMPLE 2

The PP, PP-Module, functional package or ST author could specify 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

Because the TOE can only continue its correct operation if the SFRs are 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.

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).

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 monopolizing).

K.2.2 FRU_FLT.1 Degraded fault tolerance

K.2.2.1 Component rationale and application notes

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.

EXAMPLE

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.

K.2.2.2 Operations

In FRU_FLT.1.1, the PP, PP-Module, functional package or ST author should specify the list of TOE capabilities the TOE will maintain during and after a specified failure.

In FRU_FLT.1.1, the PP, PP-Module, functional package or ST author should specify the list of types 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.2.3 FRU_FLT.2 Limited fault tolerance

K.2.3.1 Component rationale and application notes

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.

EXAMPLE

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.

K.2.3.2 Operations

In FRU_FLT.2.1, the PP, PP-Module, functional package or ST author should specify the list of types 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.3 Priority of service (FRU_PRS)

K.3.1 User application notes

The requirements of this family allow the TSF to control the use of resources under the control of the TSF by users and subjects such that high priority activities under the control of the TSF 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.

This family could be applicable to several types of resources.

EXAMPLE

Processing capacity, and communication channel capacity.

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.

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.

K.3.2 FRU_PRS.1 Limited priority of service

K.3.2.1 Component rationale and application notes

This component defines priorities for a subject, and the resources for which this priority will be used. If some subject attempts to act 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.

K.3.2.2 Operations

In FRU_PRS.1.2, the PP, PP-Module, functional package or ST author should specify the list of controlled resources for which the TSF enforces priority of service

EXAMPLE

Resources such as processes, disk space, memory, bandwidth.

K.3.3 FRU_PRS.2 Full priority of service

K.3.3.1 Component rationale and application notes

This component defines priorities for a subject. All shareable resources under the control of the TSF will be subjected to the Priority of Service mechanism. If some subject attempts to take action on a shareable TSF 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.4 Resource allocation (FRU_RSA)

K.4.1 User application notes

The requirements of this family allow the TSF to control the use of resources under the control of the TSF by users and subjects such that unauthorized denial of service will not take place by means of monopolization of resources by other users or subjects.

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.

EXAMPLE 1

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.

In general, these functions will be implemented through the use of attributes assigned to users and resources.

The objective of these components is to ensure a certain amount of fairness among the users and subjects.

EXAMPLE 2

A single user should not allocate all the available space

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.

EXAMPLE 3

Main memory or CPU cycles.

In those instances, the components allow that the resource allocation be on the level of subjects.

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.

K.4.2 FRU_RSA.1 Maximum quotas

K.4.2.1 Component rationale and application notes

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.

K.4.2.2 Operations

In FRU_RSA.1.1, the PP, PP-Module, functional package or ST author should specify the list of controlled resources for which maximum resource allocation limits are required.

EXAMPLE

Examples of controlled resources include processes, disk space, memory, and bandwidth.

If all resources under the control of the TSF need to be included, the words “all TSF resources” may be specified.

In FRU_RSA.1.1, the PP, PP-Module, functional package or 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.

In FRU_RSA.1.1, the PP, PP-Module, functional package or ST author should select whether the maximum quotas are applicable to any given time (simultaneously), or over a specific time interval.

K.4.3 FRU_RSA.2 Minimum and maximum quotas

K.4.3.1 Component rationale and application notes

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.

K.4.3.2 Operations

In FRU_RSA.2.1, the PP, PP-Module, functional package or ST author should specify the controlled resources for which maximum and minimum resource allocation limits are required.

If all resources under the control of the TSF need to be included, the words “all TSF resources” can be specified.

In FRU_RSA.2.2, the PP, PP-Module, functional package or ST author specifies the controlled resources for which a minimum allocation limit needs to be set.

If all resources under the control of the TSF need to be included the words “all TSF resources” can be specified.

EXAMPLE

Examples of controlled resources include processes, disk space, memory, and bandwidth.

In FRU_RSA.2.1, the PP, PP-Module, functional package or 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.

In FRU_RSA.2.1, the PP, PP-Module, functional package or ST author should select whether the maximum quotas are applicable to any given time (simultaneously), or over a specific time interval.

In FRU_RSA.2.2, the PP, PP-Module, functional package or ST author selects whether the minimum quotas apply to individual users, to a defined group of users, or subjects or any combination of these.

In FRU_RSA.2.2, the PP, PP-Module, functional package or ST author selects whether the minimum quotas are applicable to any given time (simultaneously), or over a specific time interval.

Annex L
(normative)

Class FTA: TOE access- application notes

L.1 General information

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.

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.

L.2 Limitation on scope of selectable attributes (FTA_LSA)

L.2.1 User application notes

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.

EXAMPLE 1

Time-of-day, day-of-week.

This family provides the capability for a PP, PP-Module, functional package or ST author to specify requirements for the TSF to place limits on the domain of an authorized user's security attributes based on an environmental condition.

EXAMPLE 2

A user could be allowed to establish a “secret session” during normal business hours but outside those hours the same user could be constrained to only establishing “unclassified sessions”.

The identification of relevant constraints on the domain of selectable attributes may be achieved through the use of the selection operation. These constraints may 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.

EXAMPLE 3

Examples of attributes that could be used to limit the session security attributes are:

- The method of access can be used to specify in which type of environment the user will be operating (such as file transfer protocol, terminal, vtam).
- 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.
- 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.

L.2.2 FTA_LSA.1 Limitation on scope of selectable attributes

L.2.2.1 Operations

In FTA_LSA.1.1, the PP, PP-Module, functional package or ST author specifies the set of session security attributes that are to be constrained.

EXAMPLE 1

Examples of these session security attributes are user clearance level, integrity level and roles.

In FTA_LSA.1.1, the PP, PP-Module, functional package or ST author specifies the set of attributes that can be used to determine the scope of the session security attributes.

EXAMPLE 2

Examples of such attributes are user identity, originating location, time of access, and method of access.

L.3 Limitation on multiple concurrent sessions (FTA_MCS)

L.3.1 User application notes

This family defines how many sessions a user may have at the same time (concurrent sessions). This number of concurrent sessions may either be set for a group of users or for each individual user.

L.3.2 FTA_MCS.1 Basic limitation on multiple concurrent sessions

L.3.2.1 Component rationale and application notes

This component allows the system to limit the number of sessions in order to effectively use the resources of the TOE.

L.3.2.2 Operations

In FTA_MCS.1.2, the PP, PP-Module, functional package or ST author specifies the default number of maximum concurrent sessions to be used.

L.3.3 FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions

L.3.3.1 Component rationale and application notes

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.

L.3.3.2 Operations

In FTA_MCS.2.1, the PP, PP-Module, functional package or ST author specifies the rules that determine the maximum number of concurrent sessions.

EXAMPLE

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”.

In FTA_MCS.2.2, the PP, PP-Module, functional package or ST author specifies the default number of maximum concurrent sessions to be used.

L.4 Session locking and termination (FTA_SSL)

L.4.1 User application notes

This family defines requirements for the TSF to provide the capability for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

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 or terminate the session. To reactivate the terminal, an event specified by the PP, PP-Module, functional package or ST author, such as the user re-authentication must occur.

A user is considered inactive, if he/she has not provided any stimulus to the TOE for a specified period of time.

PP, PP-Module, functional package or ST authors consider whether FTP_TRP.1 Trusted path should be included. In that case, the function “session locking” must be included in the operation in FTP_TRP.1 Trusted path.

L.4.2 FTA_SSL.1 TSF-initiated session locking

L.4.2.1 Component rationale and application notes

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.

If display devices are overwritten, the replacement contents need not be static (i.e. “screen savers” are permitted).

This component allows the PP, PP-Module, functional package or ST author to specify what events will unlock the session. These events may be related to the terminal, the user, or time.

EXAMPLE

Examples of events include

- Terminal related: a fixed set of keystrokes to unlock the session.
- User related: reauthentication.
- Time related: after 15 minutes.

L.4.2.2 Operations

In FTA_SSL.1.1, the PP, PP-Module, functional package or ST author specifies the interval of user inactivity that will trigger the locking of an interactive session. If so desired the PP, PP-Module, functional package or ST author could, through the assignment, specify that the time interval is left to the authorized 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.

In FTA_SSL.1.2, the PP, PP-Module, functional package or ST author specifies the event(s) that should occur before the session is unlocked.

EXAMPLE

Examples of such an event are: “user re-authentication” or “user enters unlock key-sequence”.

L.4.3 FTA_SSL.2 User-initiated locking

L.4.3.1 Component rationale and application notes

FTA_SSL.2 User-initiated locking, provides the capability for an authorized user to lock and unlock his/her own interactive session. This would provide authorized users with the ability to effectively block further use of their active sessions without having to terminate the active session.

If devices are overwritten, the replacement contents need not be static (i.e. “screen savers” are permitted).

L.4.3.2 Operations

In FTA_SSL.2.2, the PP, PP-Module, functional package or ST author specifies the event(s) that must occur before the session is unlocked.

EXAMPLE

Examples of such an event are: “user re-authentication”, or “user enters unlock key-sequence”.

L.4.4 FTA_SSL.3 TSF-initiated termination

L.4.4.1 Component rationale and application notes

FTA_SSL.3 TSF-initiated termination, requires that the TSF terminate an interactive user session after a period of inactivity.

The PP, PP-Module, functional package or ST author should be aware that a session may continue after the user terminated his/her activity. This requirement would terminate this background subject after a period of inactivity of the user without regard to the status of the subject.

EXAMPLE

An example of a continuing session after a user terminated activity is background processing.

L.4.4.2 Operations

In FTA_SSL.3.1, the PP, PP-Module, functional package or ST author specifies the interval of user inactivity that will trigger the termination of an interactive session. If so desired, the PP, PP-Module, functional package or ST author could, through the assignment, specify that the interval is left to the authorized 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.4.5 FTA_SSL.4 User-initiated termination

L.4.5.1 Component rationale and application notes

FTA_SSL.4 User-initiated termination, provides the capability for an authorized user to terminate his/her interactive session.

The PP, PP-Module, functional package or ST author should be aware that a session could continue after the user terminated his/her activity.

EXAMPLE

An example of a continuing session after a user terminated activity is background processing.

This requirement would allow the user to terminate this background subject without regard to the status of the subject.

L.5 TOE access banners (FTA_TAB)

L.5.1 User application notes

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.

L.5.2 FTA_TAB.1 Default TOE access banners

L.5.2.1 Component rationale and application notes

This component requires that there is an advisory warning regarding the unauthorized use of the TOE. A PP, PP-Module, functional package or ST author could refine the requirement to include a default banner.

L.6 TOE access history (FTA_TAH)

L.6.1 User application notes

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 could 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.

L.6.2 FTA_TAH.1 TOE access history

L.6.2.1 Component rationale and application notes

This family can provide authorized users with information that could indicate the possible misuse of their user account.

This component requests that the user is presented with the information. The user should be able to review the information but is not forced to do so.

EXAMPLE

A user might create scripts that ignore this information and start other processes.

L.6.2.2 Operations

In FTA_TAH.1.1, the PP, PP-Module, functional package or ST author selects 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, and/or location.

In FTA_TAH.1.2, the PP, PP-Module, functional package or ST author selects 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, and/or location.

EXAMPLE

Method of access: ftp.

Location: terminal 50.

L.7 TOE session establishment (FTA_TSE)

L.7.1 User application notes

This family defines requirements to deny a 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, ranges of time or combinations of parameters.

EXAMPLE 1

Security attribute: identity, clearance level, integrity level, membership in a role.

Ranges of time: time-of-day, day-of-week, calendar dates.

This family provides the capability for the PP, PP-Module, functional package or ST author to specify requirements for the TOE to place constraints on the ability of an authorized user to establish a session with the TOE. The identification of relevant constraints can be achieved through the use of the selection operation.

EXAMPLE 2

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.

This capability is particularly relevant in situations where authorization or login may take place at a different location from where TOE access checks are performed.

c) 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.

L.7.2 FTA_TSE.1 TOE session establishment

L.7.2.1 Operations

In FTA_TSE.1.1, the PP, PP-Module, functional package or ST author specifies the attributes that can be used to restrict the session establishment.

EXAMPLE

Examples of possible attributes are user identity, originating location (such as no remote terminals), time of access (such as outside hours), or method of access (such as telnet).

Annex M
(normative)

Class FTP: Trusted path/channels- application notes

M.1 General information

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 another trusted IT product.

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 could be erroneous and could lead to a security breach.

M.2 Inter-TSF trusted channel (FTP_ITC)

M.2.1 User application notes

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.

EXAMPLE

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.

M.2.2 FTP_ITC.1 Inter-TSF trusted channel

M.2.2.1 Component rationale and application notes

This component is used when a trusted communication channel between the TSF and another trusted IT product is required.

M.2.2.2 Operations

In FTP_ITC.1.2, the PP, PP-Module, functional package or ST author must specify whether the local TSF, another trusted IT product, or both shall have the capability to initiate the trusted channel.

In FTP_ITC.1.3, the PP, PP-Module, functional package or ST author specifies the functions for which a trusted channel is required.

EXAMPLE

Examples of these functions may include transfer of user, subject, and/or object security attributes and ensuring consistency of TSF data.

M.3 Trusted channel protocol (FTP_PRO)

M.3.1 User application notes

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 protection of data transfers. In contrast with FTP_ITC or FTP_TRP, FTP_PRO is concerned with security details of the protocol used for a channel and provides a focus for protocol properties that might otherwise be split between a larger number of separate SFRs. It can improve clarity of a PP/ST by highlighting mechanisms within the protocol that may then be linked to cryptographic functions described in other SFRs (such as FCS_COP.1).

The components of FTP_PRO are not hierarchical but are intended to be used together to separately specify different aspects of a trusted channel, such as its confidentiality and integrity protection features.

There is no dependency from FTP_PRO.2 to FTP_PRO.3 because any mechanisms for security of the shared secret establishment will be part of the mechanism described in FTP_PRO.2 itself.

In cases where some cryptographic operations used in the trusted channel protocol are performed outside the TOE, FTP_PRO.2 and/or FTP_PRO.3 might be omitted from an ST, and the ST author would then need to supply a rationale for the unsatisfied dependencies between FTP_PRO components.

Separate iterations of the relevant FTP_PRO components may be used for different channels where the completion of the SFR needs to be different for each channel. In general, each separate iteration will need to include all three components with its own dependencies' rationale.

M.3.2 FTP_PRO.1 Trusted channel protocol

M.3.2.1 Component rationale and application notes

Where values used in the completion of FTP_PRO operations have dependencies between different SFR elements, these need to be made clear in the instantiation of the SFR.

EXAMPLE

A table could be given in which the columns represent the relevant selections and assignments, and the rows define the valid combination of completion values.

M.3.2.2 Operations

In FTP_PRO.1.1, if selected, the PP, PP-Module, functional package or ST author should specify a trusted channel protocol and the defined protocol roles.

EXAMPLE 1

Examples of "defined protocol roles" would be 'client' or 'server' (TLS), 'initiator' or 'responder' (IKEv2/IPsec), 'Trust Center' (ZigBee) or 'Key Distribution Centre' (Kerberos).

In FTP_PRO.1.2 the first assignment is intended to state rules for when the trusted channel is required to be used by the TOE, such as mandating its use for communications with an audit server. This assignment may take the value 'None specified' (also with 'None specified' in the second assignment) if no specific uses of the channel are mandated for the TOE.

In FTP_PRO.1.5 the assignment is intended to state rules related to implementation of the protocol(s). It may take the value 'None specified' if no rules are required, or if the standards referenced in other elements of the SFR include the relevant rules and no specific evaluator check is required for the context in which the SFR is being used.

EXAMPLE 2

Rules include those for maximum packet sizes or rekey intervals

In FTP_PRO.1.6 the assignment is intended to state rules related to negotiable aspects of the protocol, when intending to narrow the options provided by the TOE compared to the standard that defines the protocol.

EXAMPLE 3

Specification of acceptable ciphersuites or acceptable older protocol versions.

The assignment may take the value 'None specified' if no rules are required. Where the assignment is completed with a list then that list specifies the only configurations permitted –

any other configuration would be a violation of the SFR. This element may be used to specify mandatory supported configurations without limiting the TOE to using these configurations by, for example, listing the required configurations with "(support required)" after each entry in the list and then including a final element which states that any other configuration permitted by the standard is allowed.

In FTP_PRO.1.3 the PP, PP-Module, functional package or ST author selects which entity is allowed to initiate the establishment of the trusted channel.

M.3.3 FTP_PRO.2 Trusted channel establishment

M.3.3.1 Component rationale and application notes

In FTP_PRO.2, the 'list of rules for carrying out the authentication' may be used to limit available parameters for the authentication mechanisms.

EXAMPLE

Rules might be stated for the format (e.g. FQDN or IP address, use of wildcards) or prioritization of identifiers when alternative sources of an identifier are available in the authentication data exchanged.

M.3.3.2 Operations

In FTP_PRO.2.2 the selection indicating the direction of the authentication should be chosen.

In FTP_PRO.2.1 The PP, PP-Module, functional package or ST author provides a list of key establishment mechanisms.

In FTP_PRO.2.2 the assignments include providing a list of authentication mechanisms used during the authentication and a list of rules used during the authentication.

M.3.4 FTP_PRO.3 Trusted channel data protection

M.3.4.1 Component rationale and application notes

The FTP_PRO.3 component addresses protection (confidentiality and integrity) of data in transit through a trusted channel.

M.3.4.2 Operations

The PP, PP-Module, functional package or ST author selects the attacks that are mitigated by the TSF.

The PP, PP-Module, functional package or ST author completes the assignment by specifying lists of encryption and integrity protection mechanisms.

EXAMPLE

Examples of integrity protection mechanism include protection of contents and file-system permissions of system files and directories; protection of processes against code injection, and protection against unsigned kernel extensions.

M.4 Trusted path (FTP_TRP)

M.4.1 User application notes

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.

M.4.2 FTP_TRP.1 Trusted path

M.4.2.1 Component rationale and application notes

This component is used when trusted communication between a user and the TSF is required, either for initial authentication purposes only or for additional specified user operations.

M.4.2.2 Operations

In FTP_TRP.1.1, the PP, PP-Module, functional package or ST author specifies whether the trusted path must be extended to remote and/or local users.

In FTP_TRP.1.1, the PP, PP-Module, functional package or ST author specifies whether the trusted path shall protect the data from modification, disclosure, and/or other types of integrity or confidentiality violation.

In FTP_TRP.1.1, if selected, the PP, PP-Module, functional package or ST author identifies any additional types of integrity or confidentiality violation against which the trusted path shall protect the data.

In FTP_TRP.1.2, the PP, PP-Module, functional package or ST author specifies whether the TSF, local users, and/or remote users are able to initiate the trusted path.

In FTP_TRP.1.3, the PP, PP-Module, functional package or ST author specifies whether the trusted path is to be used for initial user authentication and/or for other specified services.

In FTP_TRP.1.3, if selected, the PP, PP-Module, functional package or ST author identifies other services for which trusted path is required, if any.