Common Criteria for Information Technology Security Evaluation

Part 2 : Annexes

15 November 1998

Foreword

ISO (the International Organisation for Standardisation) and the IEC (the International Electrotechnical Commission) form the specialised system for worldwide standardisation. National bodies that are members of ISO or IEC participate in the development of international standards through technical committees established by the respective organisation to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organisations, 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. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75% of the national bodies casting a vote.

International Standard ISO/IEC 15408 was prepared by Joint Technical Committee ISO/IEC JTC 1, Information Technology, in collaboration with the Common Criteria Implementation Board, a joint entity composed of members of the Common Criteria Project Sponsoring Organisations. The identical text of ISO/IEC 15408 is published by the Common Criteria Project Sponsoring Organisations as *Common Criteria for Information Technology Security Evaluation.* Additional information on the Common Criteria Project and contact information on its Sponsoring Organisations is provided in Annex A of Part 1.

ISO/IEC 15408 consists of the following parts, under the general title *Evaluation Criteria for Information Technology Security*.

Part 1: Introduction and General Model

Part 2: Security Functional Requirements

Part 3: Security Assurance Requirements

This LEGAL NOTICE has been placed in all Parts of ISO/IEC 15408 by request:
The seven governmental organisations (collectively called "the Common Criteria Project Sponsoring Organisations") listed just below and identified more fully in Part 1 Annex A, as the joint holders of the copyright in the Common Criteria for Information Technology Security Evaluation, version 2.0, Parts 1 through 3 (called "CC 2.0"), hereby grant non-exclusive license to ISO/IEC to use CC 2.0 in the development of the ISO/IEC 15408 international standard. However, the Common Criteria Project Sponsoring Organisations retain the right to use, copy, distribute, or modify CC 2.0 as they see fit.

- Canada:	*Communications Security Establishment*
- France:	*Service Central de la Sécurité des Systèmes d'Information*
- Germany:	*Bundesamt für Sicherheit in der Informationstechnik*
- Netherlands:	*Netherlands National Communications Security Agency*
- United Kingdom:	*Communications-Electronics Security Group*
- United States:	*National Institute of Standards and Technology*
- United States:	*National Security Agency*

Table of Contents

Annex A
Security functional requirements application notes 1

A.1 Structure of the notes 1
A.1.1 Class structure 1
A.1.2 Family structure 2
A.1.3 Component structure 2
A.2 Dependency table 3

Annex B
Functional classes, families, and components 11

Annex C
Security audit (FAU) 13

C.1 Security audit automatic response (FAU_ARP) 15
C.2 Security audit data generation (FAU_GEN) 16
C.3 Security audit analysis (FAU_SAA) 19
C.4 Security audit review (FAU_SAR) 24
C.5 Security audit event selection (FAU_SEL) 26
C.6 Security audit event storage (FAU_STG) 27

Annex D
Communication (FCO) 31

D.1 Non-repudiation of origin (FCO_NRO) 32
D.2 Non-repudiation of receipt (FCO_NRR) 35

Annex E
Cryptographic support (FCS) 39

E.1 Cryptographic key management (FCS_CKM) 41
E.2 Cryptographic operation (FCS_COP) 44

Annex F
User data protection (FDP) 47

F.1 Access control policy (FDP_ACC) 51
F.2 Access control functions (FDP_ACF) 53
F.3 Data authentication (FDP_DAU) 56
F.4 Export to outside TSF control (FDP_ETC) 58
F.5 Information flow control policy (FDP_IFC) 60
F.6 Information flow control functions (FDP_IFF) 63
F.7 Import from outside TSF control (FDP_ITC) 69
F.8 Internal TOE transfer (FDP_ITT) 72
F.9 Residual information protection (FDP_RIP) 76

F.10 Rollback (FDP_ROL) 78
F.11 Stored data integrity (FDP_SDI) 80
F.12 Inter-TSF user data confidentiality transfer protection (FDP_UCT) 82
F.13 Inter-TSF user data integrity transfer protection (FDP_UIT) 83

Annex G
Identification and authentication (FIA) 85

G.1 Authentication failures (FIA_AFL) 87
G.2 User attribute definition (FIA_ATD) 89
G.3 Specification of secrets (FIA_SOS) 90
G.4 User authentication (FIA_UAU) 92
G.5 User identification (FIA_UID) 96
G.6 User-subject binding (FIA_USB) 97

Annex H
Security management (FMT) 99

H.1 Management of functions in TSF (FMT_MOF) 101
H.2 Management of security attributes (FMT_MSA) 103
H.3 Management of TSF data (FMT_MTD) 106
H.4 Revocation (FMT_REV) 108
H.5 Security attribute expiration (FMT_SAE) 109
H.6 Security management roles (FMT_SMR) 110

Annex I
Privacy (FPR) 113

I.1 Anonymity (FPR_ANO) 115
I.2 Pseudonymity (FPR_PSE) 117
I.3 Unlinkability (FPR_UNL) 122
I.4 Unobservability (FPR_UNO) 124

Annex J
Protection of the TOE Security Functions (FPT) 129

J.1 Underlying abstract machine test (FPT_AMT) 132
J.2 Fail secure (FPT_FLS) 134
J.3 Availability of exported TSF data (FPT_ITA) 135
J.4 Confidentiality of exported TSF data (FPT_ITC) 136
J.5 Integrity of exported TSF data (FPT_ITI) 137
J.6 Internal TOE TSF data transfer (FPT_ITT) 139
J.7 TSF physical protection (FPT_PHP) 141
J.8 Trusted recovery (FPT_RCV) 143
J.9 Replay detection (FPT_RPL) 146
J.10 Reference mediation (FPT_RVM) 147
J.11 Domain separation (FPT_SEP) 148
J.12 State synchrony protocol (FPT_SSP) 150
J.13 Time stamps (FPT_STM) 151

J.14 Inter-TSF TSF data consistency (FPT_TDC) 152
J.15 Internal TOE TSF data replication consistency (FPT_TRC) 153
J.16 TSF self test (FPT_TST) 154

Annex K
Resource utilisation (FRU) 157

K.1 Fault tolerance (FRU_FLT) 158
K.2 Priority of service (FRU_PRS) 160
K.3 Resource allocation (FRU_RSA) 161

Annex L
TOE access (FTA) 163

L.1 Limitation on scope of selectable attributes (FTA_LSA) 164
L.2 Limitation on multiple concurrent sessions (FTA_MCS) 165
L.3 Session locking (FTA_SSL) 166
L.4 TOE access banners (FTA_TAB) 168
L.5 TOE access history (FTA_TAH) 169
L.6 TOE session establishment (FTA_TSE) 170

Annex M
Trusted path/channels (FTP) 171

M.1 Inter-TSF trusted channel (FTP_ITC) 172
M.2 Trusted path (FTP_TRP) 173

List of Figures

Figure A.1 - Functional class structure 1
Figure A.2 - Functional family structure for application notes 2
Figure A.3 - Functional component structure 3
Figure C.1 - Security audit class decomposition 14
Figure D.1 - Communication class decomposition 31
Figure E.1 - Cryptographic support class decomposition 39
Figure F.1 - User data protection class decomposition 48
Figure F.2 - User data protection class decomposition (cont.) 49
Figure G.1 - Identification and authentication class decomposition 86
Figure H.1 - Security management class decomposition 100
Figure I.1 - Privacy class decomposition 113
Figure J.1 - Protection of the TOE Security Functions class decomposition 129
Figure J.2 - Protection of the TOE Security Functions class decomposition (Cont.) 130
Figure K.1 - Resource utilisation class decomposition 157
Figure L.1 - TOE access class decomposition 163
Figure M.1 - Trusted path/channels class decomposition 171

List of Tables

Table A.1 - Dependency table for functional components 4

Annex A

Security functional requirements application notes

This annex contains informative guidance for the families and components found in the main body of Part 2, which may be required by users, developers or evaluators to use the components. To facilitate finding the appropriate information, the presentation of the classes, families and components in this annex is similar to the main body of Part 2. The class, family, and component structures in this annex differ from those found in the main body of Part 2, as this annex is concerned with only those sections that are informative.

A.1 Structure of the notes

This section defines the content and presentation of the notes related to functional requirements of the CC.

A.1.1 Class structure

Figure A.1 below illustrates the functional class structure in this annex.

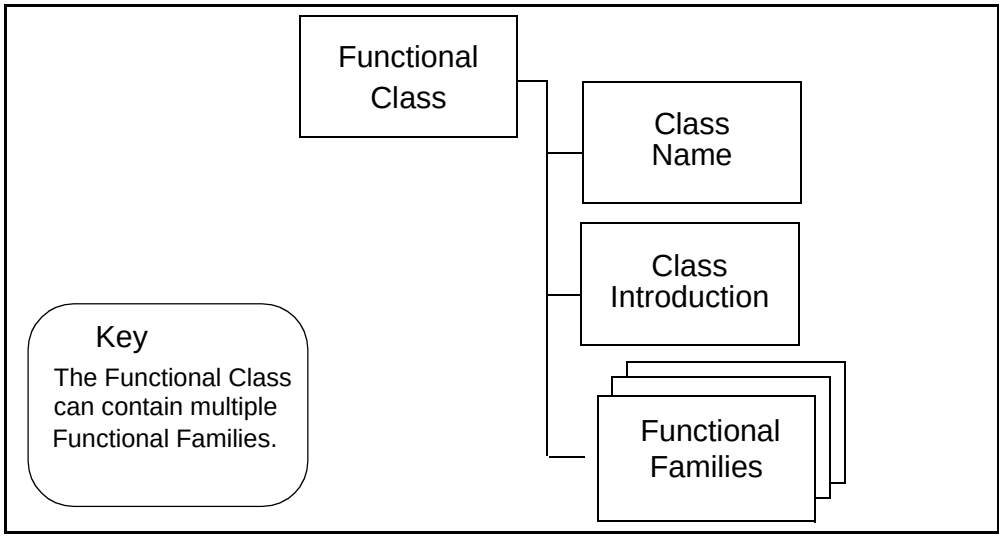

Figure A.1 - Functional class structure

A.1.1.1 Class name

This is the unique name of the class defined in Part 2 of the CC.

A.1.1.2 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 organisation of each class with the families in each class and the hierarchical relationship between components in each family.

A.1.2 Family structure

Figure A.2 illustrates the functional family structure for application notes in diagrammatic form.

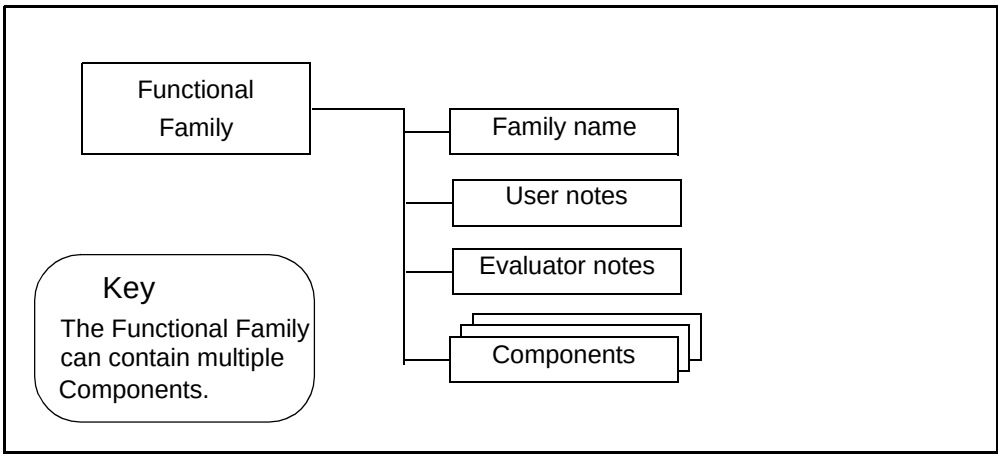

Figure A.2 - Functional family structure for application notes

A.1.2.1 Family name

This is the unique name of the family defined in Part 2 of the CC.

A.1.2.2 User notes

The *user notes* contain additional information that is of interest to potential users of the family, that is PP, ST and functional package authors, and developers of TOEs incorporating the functional components. The presentation is informative, and might cover warnings about limitations of use and areas where specific attention might be required when using the components.

A.1.2.3 Evaluator notes

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 sections are not mandatory and appear only if appropriate.

A.1.3 Component structure

Figure A.3 illustrates the functional component structure for the application notes.

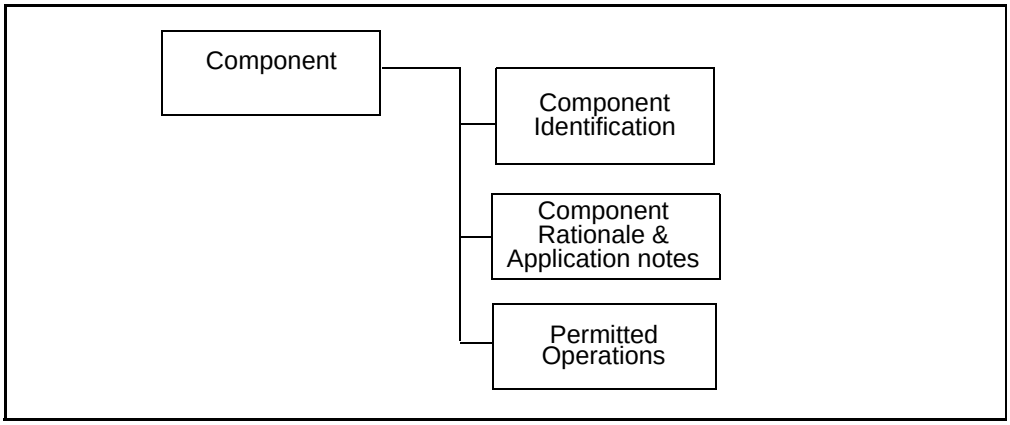

Figure A.3 - Functional component structure

A.1.3.1 Component identification

This is the unique name of the component defined in Part 2 of the CC.

A.1.3.2 Component rationale and application notes

Any specific information related to the component can be found in this section.

- The *rationale* contains the specifics of the rationale that refine the general statements on rationale for the specific level, and should only be used if level specific amplification is required.

- The *application notes* contain additional refinement in terms of narrative qualification as it pertains to a specific component. This refinement can pertain to user notes, and/ or evaluator notes as described in section A.1.2 of this annex. This refinement can be used to explain the nature of the dependencies (e.g. shared information, or shared operation).

This section is not mandatory and appears only if appropriate.

A.1.3.3 Permitted operations

This portion of each component contains advice relating to the permitted operations of the component.

This section is not mandatory and appears only if appropriate.

A.2 Dependency table

Table A.1 - Dependency table for functional components, shows their direct, indirect and optional dependencies. Each of the components that is a dependency of some functional component is allocated a column. Each functional component is allocated a row. The value in the table cell indicate whether the column label component is directly required (indicated by a cross ‘X’),

indirectly required (indicated by a dash '-'), or optionally required (indicated by a 'o') by the row label component. An example of a component with optional dependencies is FDP_ETC.1, which requires either FDP_ACC.1 or FDP_IFC.1 to be present. So if FDP_ACC.1 is present, FDP_IFC.1 is not necessary and vice versa. If no character is presented, the component is not dependent upon another component.

Table A.1 - Dependency table for functional components

	ADV_SPM.1	AGD_ADM.1	AVA_CCA.1	AVA_CCA.3	FAU_GEN.1	FAU_SAA.1	FAU_SAR.1	FAU_STG.1	FCS_CKM.1	FCS_CKM.2	FCS_CKM.4	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	FDP_UIT.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MOF.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMR.1	FPR_UNO.1	FPT_AMT.1	FPT_FLS.1	FPT_ITT.1	FPT_STM.1	FPT_TDC.1	FPT_TST.1	FTP_ITC.1	FTP_TRP.1
FAU_ARP.1					-	x																												-				
FAU_GEN.1																																		x				
FAU_GEN.2					x																		x											-				
FAU_SAA.1					x																													-				
FAU_SAA.2																							x															
FAU_SAA.3																																						
FAU_SAA.4																																						
FAU_SAR.1					x																													-				
FAU_SAR.2					-		x																											-				
FAU_SAR.3					-		x																											-				
FAU_SEL.1					x																		-					x	-					-				
FAU_STG.1					x																													-				
FAU_STG.2					x																													-				
FAU_STG.3					-			x																										-				
FAU_STG.4					x																													-				
FCO_NRO.1																							x															
FCO_NRO.2																							x															
FCO_NRR.1																							x															
FCO_NRR.2																							x															

Table A.1 - Dependency table for functional components

	ADV_SPM.1	AGD_ADM.1	AVA_CCA.1	AVA_CCA.3	FAU_GEN.1	FAU_SAA.1	FAU_SAR.1	FAU_STG.1	FCS_CKM.1	FCS_CKM.2	FCS_CKM.4	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	FDP_UIT.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MOF.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMR.1	FPR_UNO.1	FPT_AMT.1	FPT_FLS.1	FPT_ITT.1	FPT_STM.1	FPT_TDC.1	FPT_TST.1	FTP_ITC.1	FTP_TRP.1
FCS_CKM.1	-								-	o	x	o	-	-	-	-	-						-		-	x	-		-									
FCS_CKM.2	-								o	-	x	-	-	-	-	-	o						-		-	x	-		-									
FCS_CKM.3	-								o	-	x	-	-	-	-	-	o						-		-	x	-		-									
FCS_CKM.4	-								o	-	-	-	-	-	-	-	o						-		-	x	-		-									
FCS_COP.1	-								o	-	x	-	-	-	-	-	o						-		-	x	-		-									
FDP_ACC.1													-	x	-	-							-		-		-		-									
FDP_ACC.2													-	x	-	-							-		-		-		-									
FDP_ACF.1													x	-	-	-							-		-		x		-									
FDP_DAU.1																																						
FDP_DAU.2																							x															
FDP_ETC.1													o	-	o	-							-		-		-		-									
FDP_ETC.2													o	-	o	-							-		-		-		-									
FDP_IFC.1													-	-	-	x							-		-		-		-									
FDP_IFC.2													-	-	-	x							-		-		-		-									
FDP_IFF.1													-	-	x	-							-		-		x		-									
FDP_IFF.2													-	-	x	-							-		-		x		-									
FDP_IFF.3			x										-	-	x	-							-		-		-		-									
FDP_IFF.4			x										-	-	x	-							-		-		-		-									
FDP_IFF.5				x									-	-	x	-							-		-		-		-									
FDP_IFF.6			x										-	-	x	-							-		-		-		-									
FDP_ITC.1													o	-	o	-							-		-		x		-									
FDP_ITC.2													o	-	o	-							-		-		-		-						x		o	
FDP_ITT.1													o	-	o	-							-		-		-		-									

Table A.1 - Dependency table for functional components

	ADV_SPM.1	AGD_ADM.1	AVA_CCA.1	AVA_CCA.3	FAU_GEN.1	FAU_SAA.1	FAU_SAR.1	FAU_STG.1	FCS_CKM.1	FCS_CKM.2	FCS_CKM.4	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	FDP_UIT.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MOF.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMR.1	FPR_UNO.1	FPT_AMT.1	FPT_FLS.1	FPT_ITT.1	FPT_STM.1	FPT_TDC.1	FPT_TST.1	FTP_ITC.1	FTP_TRP.1
FDP_ITT.2													o	-	o	-							-		-		-		-									
FDP_ITT.3													o	-	o	-		x					-		-		-		-									
FDP_ITT.4													o	-	o	-			x				-		-		-		-									
FDP_RIP.1																																						
FDP_RIP.2																																						
FDP_ROL.1													o	-	o	-							-		-		-		-									
FDP_ROL.2													o	-	o	-							-		-		-		-									
FDP_SDI.1																																						
FDP_SDI.2																																						
FDP_UCT.1													o	-	o	-							-		-		-		-								o	o
FDP_UIT.1													o	-	o	-							-		-		-		-								o	o
FDP_UIT.2													o	-	o	-				x			-		-		-		-								x	
FDP_UIT.3													o	-	o	-				x			-		-		-		-								x	
FIA_AFL.1																						x	-															
FIA_ATD.1																																						
FIA_SOS.1																																						
FIA_SOS.2																																						
FIA_UAU.1																							x															
FIA_UAU.2																							x															
FIA_UAU.3																																						
FIA_UAU.4																																						
FIA_UAU.5																																						
FIA_UAU.6																																						

Table A.1 - Dependency table for functional components

	ADV_SPM.1	AGD_ADM.1	AVA_CCA.1	AVA_CCA.3	FAU_GEN.1	FAU_SAA.1	FAU_SAR.1	FAU_STG.1	FCS_CKM.1	FCS_CKM.2	FCS_CKM.4	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	FDP_UIT.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MOF.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_MTD.2	FMT_MTD.3	FMT_SMR.1	FPR_UNO.1	FPT_AMT.1	FPT_FLS.1	FPT_ITT.1	FPT_STM.1	FPT_TDC.1	FPT_TST.1	FTP_ITC.1	FTP_TRP.1
FIA_UAU.7																						x	-																	
FIA_UID.1																																								
FIA_UID.2																																								
FIA_USB.1																					x																			
FMT_MOF.1																							-								x									
FMT_MSA.1													o	-	o	-							-		-		-				x									
FMT_MSA.2	x												o	-	o	-							-		x		-				x									
FMT_MSA.3													-	-	-	-							-		x		-				x									
FMT_MTD.1																							-								x									
FMT_MTD.2																							-						x		x									
FMT_MTD.3	x																						-						x		-									
FMT_REV.1																							-								x									
FMT_SAE.1																							-								x					x				
FMT_SMR.1																							x																	
FMT_SMR.2																																								
FMT_SMR.3																							-								x									
FPR_ANO.1																																								
FPR_ANO.2																																								
FPR_PSE.1																																								
FPR_PSE.2																							x																	
FPR_PSE.3																																								
FPR_UNL.1																																								
FPR_UNO.1																																								

Table A.1 - Dependency table for functional components

	ADV_SPM.1	AGD_ADM.1	AVA_CCA.1	AVA_CCA.3	FAU_GEN.1	FAU_SAA.1	FAU_SAR.1	FAU_STG.1	FCS_CKM.1	FCS_CKM.2	FCS_CKM.4	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	FDP_UIT.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MOF.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMR.1	FPR_UNO.1	FPT_AMT.1	FPT_FLS.1	FPT_ITT.1	FPT_STM.1	FPT_TDC.1	FPT_TST.1	FTP_ITC.1	FTP_TRP.1
FPR_UNO.2																																						
FPR_UNO.3																														X								
FPR_UNO.4																																						
FPT_AMT.1																																						
FPT_FLS.1	X																																					
FPT_ITA.1																																						
FPT_ITC.1																																						
FPT_ITI.1																																						
FPT_ITI.2																																						
FPT_ITT.1																																						
FPT_ITT.2																																						
FPT_ITT.3																																	X					
FPT_PHP.1																							-	X					-									
FPT_PHP.2																							-	X					-									
FPT_PHP.3																																						
FPT_RCV.1	X	X																													-					X		
FPT_RCV.2	X	X																													-					X		
FPT_RCV.3	X	X																													-					X		
FPT_RCV.4	X																																					
FPT_RPL.1																																						
FPT_RVM.1																																						
FPT_SEP.1																																						
FPT_SEP.2																																						

Table A.1 - Dependency table for functional components

	ADV_SPM.1	AGD_ADM.1	AVA_CCA.1	AVA_CCA.3	FAU_GEN.1	FAU_SAA.1	FAU_SAR.1	FAU_STG.1	FCS_CKM.1	FCS_CKM.2	FCS_CKM.4	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	FDP_UIT.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MOF.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMR.1	FPR_UNO.1	FPT_AMT.1	FPT_FLS.1	FPT_ITT.1	FPT_STM.1	FPT_TDC.1	FPT_TST.1	FTP_ITC.1	FTP_TRP.1
FPT_SEP.3																																						
FPT_SSP.1																																	X					
FPT_SSP.2																																	X					
FPT_STM.1																																						
FPT_TDC.1																																						
FPT_TRC.1																																	X					
FPT_TST.1																															X							
FRU_FLT.1	-																															X						
FRU_FLT.2	-																															X						
FRU_PRS.1																																						
FRU_PRS.2																																						
FRU_RSA.1																																						
FRU_RSA.2																																						
FTA_LSA.1																																						
FTA_MCS.1																							X															
FTA_MCS.2																							X															
FTA_SSL.1																						X	-															
FTA_SSL.2																						X	-															
FTA_SSL.3																																						
FTA_TAB.1																																						
FTA_TAH.1																																						
FTA_TSE.1																																						
FTP_ITC.1																																						

Table A.1 - Dependency table for functional components

	ADV_SPM.1	AGD_ADM.1	AVA_CCA.1	AVA_CCA.3	FAU_GEN.1	FAU_SAA.1	FAU_SAR.1	FAU_STG.1	FCS_CKM.1	FCS_CKM.2	FCS_CKM.4	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	FDP_UIT.1	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_MOF.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMR.1	FPR_UNO.1	FPT_AMT.1	FPT_FLS.1	FPT_ITT.1	FPT_STM.1	FPT_TDC.1	FPT_TST.1	FTP_ITC.1	FTP_TRP.1
FTP_TRP.1																																						

Annex B

Functional classes, families, and components

The following Annexes C through M provide the application notes for the functional classes defined in the main body of this Part 2.

Annex C

Security audit (FAU)

CC audit families allow PP/ST authors the ability to define requirements for monitoring user activities and, in some cases, detecting real, potential, or imminent violations of the TSP. The TOE's security audit functions are defined to help monitor security-relevant events, and act as a deterrent against security violations. The requirements of the audit families refer to functions that include audit data protection, record format, and event selection, as well as analysis tools, violation alarms, and real-time analysis. The audit trail should be presented in human-readable format either directly (e.g. storing the audit trail in human-readable format) or indirectly (e.g. using audit reduction tools), or both.

While developing the security audit requirements, the PP/ST author should take note of the inter-relationships among the audit families and components. The potential exists to specify a set of audit requirements that comply with the family/component dependencies lists, while at the same time resulting in a deficient audit function (e.g. an audit function that requires all security relevant events to be audited but without the selectivity to control them on any reasonable basis such as individual user or object).

Audit requirements in a distributed environment:

The implementation of audit requirements for networks and other large systems may differ significantly from those needed for stand-alone systems. Larger, more complex and active systems require more thought concerning which audit data to collect and how this should be managed, due to lowered feasibility of interpreting (or even storing) what gets collected. The traditional notion of a time-sorted list or "trail" of audited events may not be applicable in a global asynchronous network with arbitrarily many events occurring at once.

Also, different hosts and servers on a distributed TOE may have differing naming policies and values. Symbolic names presentation for audit review may require a net-wide convention to avoid redundancies and "name clashes."

A multi-object audit repository, portions of which are accessible by a potentially wide variety of authorised users, may be required if audit repositories are to serve a useful function in distributed systems.

Finally, misuse of authority by authorised users should be addressed by systematically avoiding local storage of audit data pertaining to administrator actions.

Figure C.1 shows the decomposition of this class into its constituent components.

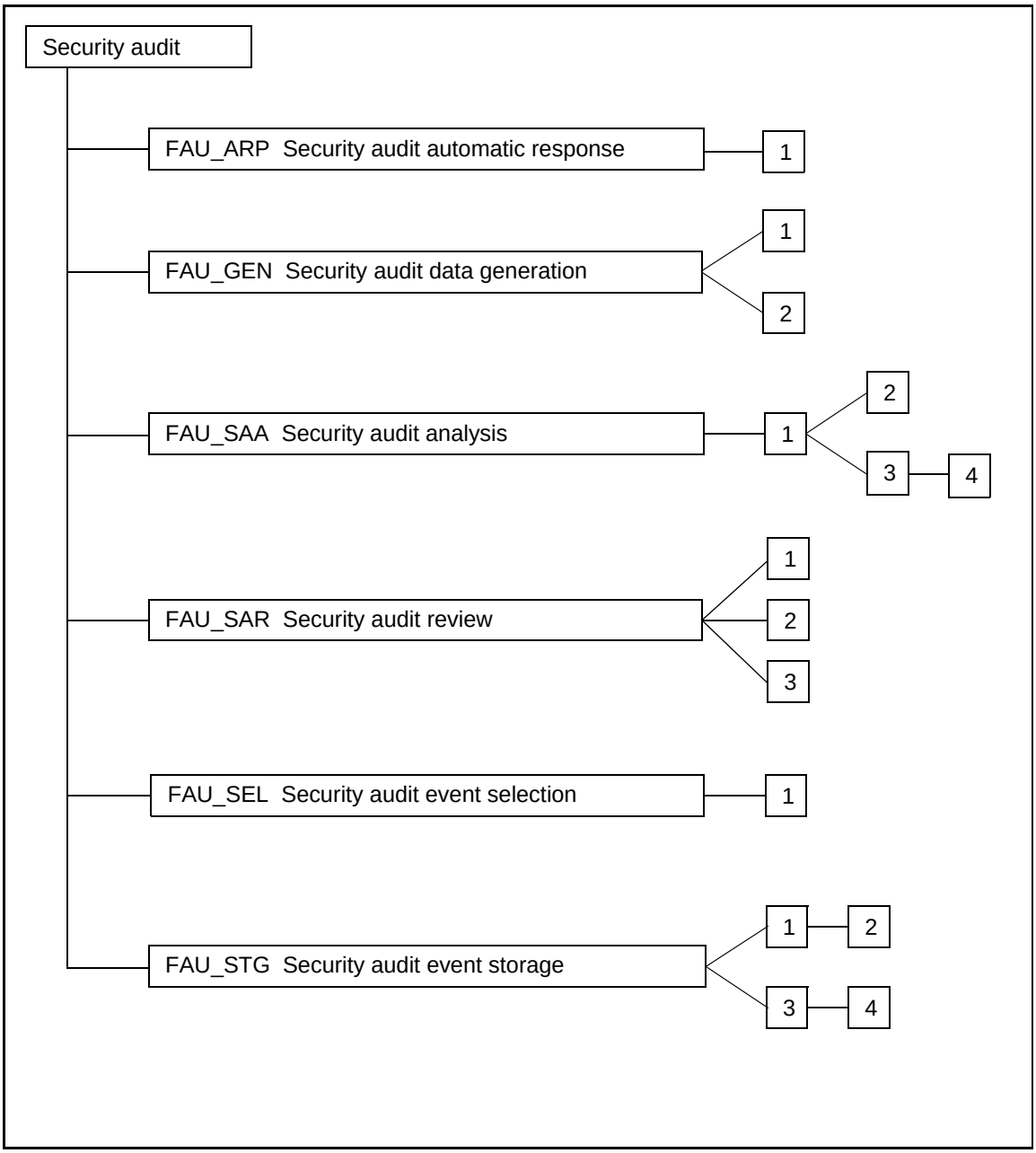

Figure C.1 - Security audit class decomposition

C.1 Security audit automatic response (FAU_ARP)

The Security audit automatic response family describes requirements for the handling of audit events. The requirement could include requirements for alarms or TSF action (automatic response). For example, the TSF could include the generation of real time alarms, termination of the offending process, disabling of a service, or disconnection or invalidation of a user account.

Application Notes

An audit event is defined to be an “potential security violation” if so indicated by the FAU_SAA components.

FAU_ARP.1 Security alarms

User application notes

An action should be taken for follow up action in the event of an alarm. This action can be to inform the authorised user, to present the authorised user with a set of possible containment actions, or to take corrective actions. The timing of the actions should be carefully considered by the PP/ST author.

Operations

Assignment:

In FAU_ARP.1.1 the PP/ST author should specify the actions to be taken in case of a potential security violation. An example of such a list is: “inform the authorised user, disable the subject that created the potential security violation.” It can also specify that the action to be taken can be specified by an authorised user.

C.2 Security audit data generation (FAU_GEN)

The Security audit data generation family includes requirements to specify the audit events that should 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 section developed in which the events to be audited for that functional area are listed. When the PP/ST author assembles the PP/ST, the items in the audit area are used to complete the variable in this component. Thus, the specification of what could be audited for a functional area is localised in that functional area.

The list of auditable events is entirely dependent on the other functional families within the PP/ST. Each family definition should therefore include a list of its family-specific auditable events. Each auditable event in the list of auditable events specified in the functional family should correspond to one of the levels of audit event generation specified in this family (i.e. minimal, basic, detailed). This provides the PP/ST author with information necessary to ensure that all appropriate auditable events are specified in the PP/ST. The following example shows how auditable events are to be specified in appropriate functional families:

> "The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
>
> a) Minimal: Successful use of the user security attribute administration functions;
>
> b) Basic: All attempted uses of the user security attribute administration functions;
>
> c) Basic: Identification of which user security attributes have been modified;
>
> d) Detailed: With the exception of specific sensitive attribute data items (e.g. passwords, cryptographic keys), the new values of the attributes should be captured."

For each functional component that is chosen, the auditable events that are indicated in that component, at and below the level indicated in FAU_GEN should be auditable. If, for example, in the previous example 'Basic' would be selected in FAU_GEN, the auditable events mentioned in a), b) and c) should be auditable.

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

A PP/ST author may decide to include other auditable events beyond those required for a given audit level. For example, the PP/ST may claim only minimal audit capabilities while including most of the basic capabilities because the few excluded capabilities conflict with other PP/ST constraints (e.g. because they require the collection of unavailable data).

Application Notes

The functionality that creates the auditable event should be specified in the PP or ST as a functional requirement.

The following are examples of the types of the events that should be defined as auditable within each PP/ST functional component:

a) Introduction of objects within the TSC into a subject's address space;

b) Deletion of objects;

c) Distribution or revocation of access rights or capabilities;

d) Changes to subject or object security attributes;

e) Policy checks performed by the TSF as a result of a request by a subject;

f) The use of access rights to bypass a policy check;

g) Use of Identification and Authentication functions;

h) Actions taken by an operator, and/or authorised user (e.g. suppression of a TSF protection mechanism as human-readable labels);

i) Import/export of data from/to removable media (e.g. printed output, tapes, diskettes).

FAU_GEN.1 Audit data generation

User application notes

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 by itself might be used when the TSP does not require that individual user identities be associated with audit events. This could be appropriate when the PP/ST also contains privacy requirements. If the user identity must be incorporated FAU_GEN.2 could be used in addition.

Evaluator application notes

There is a dependency on FPT_STM. If correctness of time is not an issue for this TOE, elimination of this dependency could be justified.

Operations

Selection:

For FAU_GEN.1.1b, the PP/ST author should select the level of auditable events called out in the audit section of other functional components included in the PP/ST. This level could be 'minimum', 'basic', 'detailed' or 'not specified'. If 'not specified' is selected, the PP/ST author should fill in all desired auditable events in FAU_GEN.1.1c, and this part of the element (item b) can be removed entirely.

Assignment:

For FAU_GEN.1.1c, the PP/ST author should assign a list of other specifically defined auditable events to be included in the list of auditable events. These events

could be auditable events of a functional requirement that are of higher audit level than requested in FAU_GEN.1.1b, as well as the events generated through the use of a specified Application Programming Interface (API).

For FAU_GEN.1.2b, the PP/ST author should assign, for each auditable events included in the PP/ST, a list of other audit relevant information to be included in audit event records.

FAU_GEN.2 User identity association

User 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. The user may want to keep his/her actions to himself/herself and not be identified by other persons (e.g. a site with job offers). Or it might be required in the Organisational Security Policy that the identity of the users must be protected. In those cases the objectives for audit and privacy might contradict each other. Therefore if this requirement is selected and privacy is important, inclusion of the component user pseudonimity might be considered. Requirements on determining the real user name based on its pseudonym are specified in the privacy class.

C.3 Security audit analysis (FAU_SAA)

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

The action to be performed by the TSF on detection of a possible imminent or potential violation is defined in FAU_ARP Security audit automatic response components.

Application Notes

For real-time analysis, audit data could be transformed into a useful format for automated treatment, but into a different useful format for delivery to authorised users for review.

FAU_SAA.1 Potential violation analysis

User application notes

This component is used to specify the set of auditable events whose occurrence or accumulated occurrence held to indicate a potential violation of the TSP, and any rules to be used to perform the violation analysis.

Operations

Assignment:

For FAU_SAA.1.2.a, the PP/ST author should identify the subset of defined auditable events whose occurrence or accumulated occurrence need to be detected as an indication of a potential violation of the TSP.

Assignment:

In FAU_SAA.1.2.b, the PP/ST author should specify any other rules that the TSF should use in its analysis of the audit trail. Those rules could include specific requirements to express the needs for the events to occur in a certain period of time (e.g. period of the day, duration).

FAU_SAA.2 Profile based anomaly detection

A *profile* is a structure that characterises the behaviour of users and/or subjects; it represents how the users/subjects interact with the TSF in a variety of ways. Patterns of usage are established with respect to the various types of activity the users/subjects engage in (e.g. patterns in exceptions raised, patterns in resource utilisation (when, which, how), patterns in actions performed). The ways in which the various types of activity are recorded in the profile (e.g. resource measures, event counters, timers) are referred to as *profile metrics*.

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/ST and the complexity of the required profile metrics.

This component is used to specify the set of auditable events whose occurrence or accumulated occurrence indicates a potential violation of the TSP, and any rules to be used to perform the violation analysis. This set of events or rules could be modified by the authorised user, through addition, modification or deletion of events or rules.

The PP/ST author should enumerate specifically what activity should be monitored and/or analysed by the TSF. The PP/ST author should also identify specifically what information pertaining to the activity is necessary to construct the usage profiles.

FAU_SAA.2 requires that the TSF maintain profiles of system usage. The word maintain implies that the anomaly detector is actively updating the usage profile based on new activity performed by the profile target members. It is important here that the metrics for representing user activity are defined by the PP/ST author. For example, there may be a thousand different actions an individual may be capable of performing, but the anomaly detector may choose to monitor a subset of that activity. Anomalous activity gets integrated into the profile just like non-anomalous activity (assuming the tool is monitoring those actions). Things that may have appeared anomalous four months ago, might over time become the norm (and vice-versa) as the user's work duties change. The TSF wouldn't be able to capture this notion if it filtered out anomalous activity from the profile updating algorithms.

Administrative notification should be provided such that the authorised user understands the significance of the suspicion rating.

The PP/ST author should define how to interpret suspicion ratings and the conditions under which anomalous activity is indicated to the FAU_ARP mechanism.

Operations

Assignment:

For FAU_SAA.2.1, the PP/ST author should specify the profile target group. A single PP/ST may include multiple profile target groups.

For FAU_SAA.2.3, the PP/ST author should specify conditions under which anomalous activity is reported by the TSF. Conditions may include the suspicion rating reaching a certain value, or be based on the type of anomalous activity observed.

FAU_SAA.3 Simple attack heuristics

User application notes

In practice, it is at best rare when an analysis tool can detect with certainty when a security violation is imminent. However, there do exist some system events that are so significant that they are always worthy of independent review. Example of such events include the deletion of a key TSF security data file (e.g. the password file) or activity such as a remote user attempting to gain administrative privilege. These events are referred to as *signature events* in that their occurrence in isolation from the rest of the system activity are indicative of intrusive activity.

The complexity of a given tool will depend greatly on the assignments defined by the PP/ST author in identifying the base set of signature events.

The PP/ST author should enumerate specifically what events should be monitored by the TSF in order to perform the analysis. The PP/ST author should identify specifically what information pertaining to the event is necessary to determine if the event maps to a signature event.

Administrative notification should be provided such that the authorised 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 (examples of other input data include network datagrams, resource/accounting data, or combinations of various system data).

The elements of FAU_SAA.3 do not require that the TSF implementing the immediate attack heuristics be the same TSF whose activity is being monitored. Thus, one can develop an intrusion detection component that operates independently of the system whose system activity is being analysed.

Operations

Assignment:

For FAU_SAA.3.1, the PP/ST author should identify a base subset of system events whose occurrence, in isolation from all other system activity, may indicate a violation of the TSP. These include events that by themselves indicate a clear violation to the TSP, or whose occurrence is so significant that they warrant actions.

In FAU_SAA.3.2, the PP/ST author should specify the information used to determine system activity. This information is the input data used by the analysis tool to determine the system activity that has occurred on the TOE. This data may include audit data, combinations of audit data with other system data, or may

consist of data other than the audit data. The PP/ST author should define precisely what system events and event attributes are being monitored within the input data.

FAU_SAA.4 Complex attack heuristics

User application notes

In practice, it is at best rare when an analysis tool can detect with certainty when a security violation is imminent. However, there do exist some system events that are so significant they are always worthy of independent review. Example of such events include the deletion of a key TSF security data file (e.g. the password file) or activity such as a remote user attempting to gain administrative privilege. These events are referred to as *signature events* in that their occurrence in isolation from the rest of the system activity are indicative of intrusive activity. Event sequences are an ordered set of signature events that might indicate intrusive activity.

The complexity of a given tool will depend greatly on the assignments defined by the PP/ST author in identifying the base set of signature events and event sequences.

The PP/ST author should define a base set of signature events and event sequences to be represented by the TSF. Additional signature events and event sequences may be defined by the system developer.

The PP/ST author should enumerate specifically what events should be monitored by the TSF in order to perform the analysis. The PP/ST author should identify specifically what information pertaining to the event is necessary to determine if the event maps to a signature event.

Administrative notification should be provided such that the authorised 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 (examples of other input data include network datagrams, resource/accounting data, or combinations of various system data). Levelling, therefore, requires the PP/ST author to specify the type of input data used to monitor system activity.

The elements of FAU_SAA.4 do not require that the TSF implementing the complex attack heuristics be the same TSF whose activity is being monitored. Thus, one can develop an intrusion detection component that operates independently of the system whose system activity is being analysed.

Operations

Assignment:

For FAU_SAA.4.1, the PP/ST author should identify a base set of list of sequences of system events whose occurrence are representative of known penetration scenarios. These event sequences represent known penetration scenarios. Each event represented in the sequence should map to a monitored system event, such

that as the system events are performed, they are bound (mapped) to the known penetration event sequences.

For FAU_SAA.4.1, the PP/ST author should identify a base subset of system events whose occurrence, in isolation from all other system activity, may indicate a violation of the TSP. These include events that by themselves indicate a clear violation to the TSP, or whose occurrence is so significant they warrant action.

In FAU_SAA.4.2, the PP/ST author should specify the information used to determine system activity. This information is the input data used by the analysis tool to determine the system activity that has occurred on the TOE. This data may include audit data, combinations of audit data with other system data, or may consist of data other than the audit data. The PP/ST author should define precisely what system events and event attributes are being monitored within the input data.

C.4 Security audit review (FAU_SAR)

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 that includes, for example, the ability to selectively review:

- the actions of one or more users (e.g. identification, authentication, TOE entry, and access control actions);
- the actions performed on a specific object or TOE resource;
- all of a specified set of audited exceptions; or
- actions associated with a specific TSP attribute.

Application Notes

The distinction between audit reviews is based on functionality. Audit review (only) encompasses the ability to view audit data. Selectable review is more sophisticated, and requires the ability to perform searches based on a single criterion or multiple criteria with logical (i.e. and/or) relations, sort audit data, filter audit data, before audit data are reviewed.

FAU_SAR.1 Audit review

User application notes

This component is used to specify that users and/or authorised users can read the audit records. These audit records will be provided in a manner appropriate to the user. There are different types of users (human users, machine users) that might have different needs.

The content of the audit records that can be viewed can be specified.

Operations

Assignment:

In FAU_SAR.1.1 the PP/ST author should specify the authorised users that can use this capability. If appropriate the PP/ST author may include security roles (see FMT_SMR.1 Security roles).

In FAU_SAR.1.1 the PP/ST author should specify the type of information the specified user is permitted to obtain from the audit records. Examples are “all”, “subject identity”, “all information belonging to audit records referencing this user”.

FAU_SAR.2 Restricted audit review

User application notes

This component specifies that any users not identified in FAU_SAR.1 will not be able to read the audit records.

FAU_SAR.3 Selectable audit review

User 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 (e.g. sort, filter).

Operations

Selection:

For FAU_SAR.3.1 the PP/ST author should select whether searches, sorting and/ or ordering can be performed by the TSF.

Assignment:

For FAU_SAR.3.1, the PP/ST author should assign the criteria, possibly with logical relations, to be used to select the audit data for review. The logical relations are intended to specify whether the operation can be on an individual attribute or a collection of attributes. An example of this assignment could be: "application, user account and/or location". In this case the operation could be specified using any combination of the three attributes: application, user account and location.

C.5 Security audit event selection (FAU_SEL)

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 FAU_GEN Security audit data generation family, but those events should be defined as being selectable in this component to be audited.

Application Notes

This family ensures that it is possible to keep the audit trail from becoming so large that it becomes useless, by defining the appropriate granularity of the selected security audit events.

FAU_SEL.1 Selective audit

User application notes

This component defines the criteria used for the selection of events to be audited. Those criteria could permit inclusion or exclusion of events from the set of auditable events, based on user attributes, subject attributes, objects attributes, or event types.

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 criteria for events to be audited.

The management function FMT_MTD.1 Management of TSF data will handle the rights of authorised users to query or modify the selections.

Operations

Selection:

For FAU_SEL.1.1a, the PP/ST author should select whether the security attributes upon which audit selectivity is based, is related to object identity, user identity, subject identity, host identity, or event type.

Assignment:

For FAU_SEL.1.1b, the PP/ST author should specify any additional attributes upon which audit selectivity is based.

C.6 Security audit event storage (FAU_STG)

The Security audit event storage family describes requirements for storing audit data for later use, including requirements controlling the loss of audit information due to system failure, attack and/ or exhaustion of storage space.

FAU_STG.1 Protected audit trail storage

User application notes

In a distributed environment, as the location of the audit trail is in the TSC, but not necessarily co-located with the function generating the audit data, the PP/ST author could request authentication of the originator of the audit record, or non-repudiation of the origin of the record prior storing this record in the audit trail.

The TSF will protect the audit trail from unauthorised deletion and modification. It is noted that in some systems the auditor (role) might not be authorised to delete the audit records for a certain period of time.

Operations

Selection:

In FAU_STG.1.2, the PP/ST author should specify whether the TSF shall prevent or only be able to detect modifications of the audit trail.

FAU_STG.2 Guarantees of audit data availability

User application notes

This component allows the PP/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 TSC, but not necessarily co-located with the function generating the audit data, the PP/ST author could request authentication of the originator of the audit record, or non-repudiation of the origin of the record prior storing this record in the audit trail.

Operations

Selection:

In FAU_STG.2.2, the PP/ST author should specify whether the TSF shall prevent or only be able to detect modifications of the audit trail.

In FAU_STG.2.3, the PP/ST author should specify the condition under which the TSF shall still be able to maintain a defined amount of audit data. This condition can be any one of the following: audit storage exhaustion, failure, attack.

Assignment:

In FAU_STG.2.3, the PP/ST author should specify the metric that the TSF must ensure with respect to the audit trail. This metric limits the data loss by enumerating the number of records that must be kept, or the time that records are guaranteed to be maintained. An example of the metric could be "100,000" indicating that 100,000 audit records can be stored.

FAU_STG.3 Action in case of possible audit data loss

User application notes

This component requires that actions will be taken when the audit trail exceeds certain pre-defined limits.

Operations

Assignment:

In FAU_STG.3.1, the PP/ST author should indicate the pre-defined limit. If the management functions indicate that this number might be changed by the authorised user, this value is the default value. The PP/ST author might choose to let the authorised user define this limit. In that case the assignment can be for example "an authorised user set limit".

In FAU_STG.3.1, the PP/ST author should specify actions that should be taken in case of imminent audit storage failure indicated by exceeding the threshold. Actions might include informing an authorised user.

FAU_STG.4 Prevention of audit data loss

User application notes

This component specifies the behaviour of the TOE if the audit trail is full: either audit records are ignored, or the TOE is frozen such that no auditable events can take place. The requirement also states that no matter how the requirement is instantiated, the authorised user with specific rights to this effect, can continue to generate auditable events (actions). The reason is that otherwise the authorised user could not even reset the system. Consideration should be given to the choice of the action to be taken by the TSF in the case of audit storage exhaustion, as ignoring events, which provides better availability of the TOE, will also permit actions to be performed without being recorded and without the user being accountable.

Operations

Selection:

In FAU_STG.4.1, the PP/ST author should select whether the TSF shall ignore auditable actions, or whether it should prevent auditable actions from happening,

or whether the oldest audit records should be overwritten when the TSF can no longer store audit records.

Assignment:

In FAU_STG.4.1, the PP/ST author should specify other actions that should be taken in case of audit storage failure, such as informing the authorised user.

Annex D

Communication (FCO)

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.

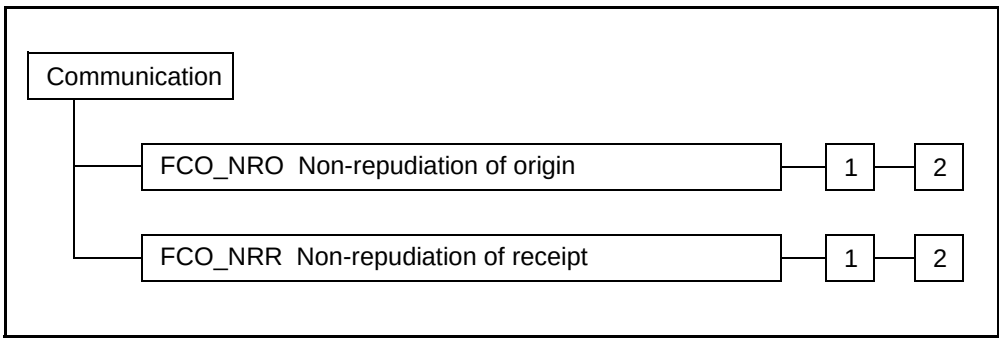

Figure D.1 - Communication class decomposition

Figure D.1 shows the decomposition of this class into its constituent components.

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 recognised that the term ‘proof’ might be interpreted in a legal sense to imply a form of mathematical rationale. The components in this class interpret the de-facto use of the word ‘proof’ in the context of ‘evidence’ that the TSF demonstrates the non-repudiated transport of types of information.

D.1 Non-repudiation of origin (FCO_NRO)

Non-repudiation of origin defines requirements to provide evidence to users/subjects about the identity of the originator of some information. The originator cannot successfully deny having sent the information because evidence of origin (e.g. digital signature) provides evidence of the binding between the originator and the information sent. The recipient or a third party can verify the evidence of origin. This evidence should not be forgeable.

User notes

If the information or the associated attributes are altered in any way, validation of the evidence of origin might fail. Therefore a PP/ST author should consider including integrity requirements such as FDP_UIT.1 Data exchange integrity in the PP/ST.

In non-repudiation there are several different roles involved, each of which could be combined in one or more subjects. The first role is a subject that requests evidence of origin (only in FCO_NRO.1 Selective proof of origin). The second role is the recipient and/or other subjects to which the evidence is provided (e.g. a notary). The third role is a subject that requests verification of the evidence of origin, for example, a recipient or a third party such as an arbiter.

The PP/ST author must specify the conditions that must be met to be able to verify the validity of the evidence. An example of a condition which could be specified is where the verification of evidence must occur within 24 hours. These conditions, therefore, allow the tailoring of the non-repudiation to legal requirements, such as being able to provide evidence for several years.

In most cases, the identity of the recipient will be the identity of the user who received the transmission. In some instances, the PP/ST author does not want the user identity to be exported. In that case the PP/ST author must consider whether it is appropriate to include this class, or whether the identity of the transport service provider or the identity of the host should be used.

In addition to (or instead of) the user identity, a PP/ST author might be more concerned about the time the information was transmitted. For example, requests for proposals must be transmitted before a certain date in order to be considered. In such instances, these requirements can be customised to provide a timestamp indication (time of origin).

FCO_NRO.1 Selective proof of origin

Operations

Assignment:

In FCO_NRO.1.1 the PP/ST author should fill in the types of information subject to the evidence of origin function, for example, electronic mail messages.

Selection:

In FCO_NRO.1.1 the PP/ST author should specify the user/subject who can request evidence of origin.

Assignment:

In FCO_NRO.1.1 the PP/ST author, dependent on the selection, should specify the third parties that can request evidence of receipt. A third party could be an arbiter, judge or legal body.

In FCO_NRO.1.2 the PP/ST author should fill in the list of the attributes that shall be linked to the information; for example, originator identity, time of origin, and location of origin.

In FCO_NRO.1.2 the PP/ST author should fill in the list of information fields within the information over which the attributes provide evidence of origin, such as the body of a message.

Selection:

In FCO_NRO.1.3 the PP/ST author should specify the user/subject who can verify the evidence of origin.

Assignment:

In FCO_NRO.1.3 the PP/ST author, dependent on the selection, should specify the third parties that can verify the evidence of origin.

In FCO_NRO.1.3 the PP/ST author should fill in the list of limitations under which the evidence can be verified. For example the evidence can only be verified within a 24 hour time interval. An assignment of 'immediate' or 'indefinite' is acceptable.

FCO_NRO.2 Enforced proof of origin

Operations

Assignment:

In FCO_NRO.2.1 the PP/ST author should fill in the types of information subject to the evidence of origin function, for example, electronic mail messages.

In FCO_NRO.2.2 the PP/ST author should fill in the list of the attributes that shall be linked to the information; for example, originator identity, time of origin, and location of origin.

In FCO_NRO.2.2 the PP/ST author should fill in the list of information fields within the information over which the attributes provide evidence of origin, such as the body of a message.

Selection:

In FCO_NRO.2.3 the PP/ST author should specify the user/subject who can verify the evidence of origin.

Assignment:

In FCO_NRO.2.3 the PP/ST author, dependent on the selection, should specify the third parties that can verify the evidence of origin. A third party could be an arbiter, judge or legal body.

In FCO_NRO.2.3 the PP/ST author should fill in the list of limitations under which the evidence can be verified. For example the evidence can only be verified within a 24 hour time interval. An assignment of ‘immediate’ or ‘indefinite’ is acceptable.

D.2 Non-repudiation of receipt (FCO_NRR)

Non-repudiation of receipt defines requirements to provide evidence to other users/subjects that the information was received by the recipient. The recipient cannot successfully deny having received the information because evidence of receipt (e.g. digital signature) provides evidence of the binding between the recipient attributes and the information. The originator or a third party can verify the evidence of receipt. This evidence should not be forgeable.

User notes

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/ST author should consider including integrity requirements such as FDP_UIT.1 Data exchange integrity in the PP/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, (e.g. a notary). The third role is a subject that requests verification of the evidence of receipt, for example, an originator or a third party such as an arbiter.

The PP/ST author must specify the conditions that must be met to be able to verify the validity of the evidence. An example of a condition which could be specified is where the verification of evidence must occur within 24 hours. These conditions, therefore, allow the tailoring of the non-repudiation to legal requirements, such as being able to provide evidence for several years.

In most cases, the identity of the recipient will be the identity of the user who received the transmission. In some instances, the PP/ST author does not want the user identity to be exported. In that case, the PP/ST author must consider whether it is appropriate to include this class, or whether the identity of the transport service provider or the identity of the host should be used.

In addition to (or instead of) the user identity, a PP/ST author might be more concerned about the time the information was received. For example, when an offer expires at a certain date, orders must be received before a certain date in order to be considered. In such instances, these requirements can be customised to provide a timestamp indication (time of receipt).

FCO_NRR.1 Selective proof of receipt

Operations

Assignment:

In FCO_NRR.1.1 the PP/ST author should fill in the types of information subject to the evidence of receipt function, for example, electronic mail messages.

Selection:

In FCO_NRR.1.1 the PP/ST author should specify the user/subject who can request evidence of receipt.

Assignment:

In FCO_NRR.1.1 the PP/ST author, dependent on the selection, should specify the third parties that can request evidence of receipt. A third party could be an arbiter, judge or legal body.

In FCO_NRR.1.2 the PP/ST author should fill in the list of the attributes that shall be linked to the information; for example, recipient identity, time of receipt, and location of receipt.

In FCO_NRR.1.2 the PP/ST author should fill in the list of information fields with the fields within the information over which the attributes provide evidence of receipt, such as the body a message.

Selection:

In FCO_NRR.1.3 the PP/ST author should specify the user/subjects who can verify the evidence of receipt.

Assignment:

In FCO_NRR.1.3 the PP/ST author, dependent on the selection, should specify the third parties that can verify the evidence of receipt.

In FCO_NRR.1.3 the PP/ST author should fill in the list of limitations under which the evidence can be verified. For example the evidence can only be verified within a 24 hour time interval. An assignment of 'immediate' or 'indefinite' is acceptable.

FCO_NRR.2 Enforced proof of receipt

Operations

Assignment:

In FCO_NRR.2.1 the PP/ST author should fill in the types of information subject to the evidence of receipt function, for example electronic mail messages.

In FCO_NRR.2.2 the PP/ST author should fill in the list of the attributes that shall be linked to the information; for example, recipient identity, time of receipt, and location of receipt.

In FCO_NRR.2.2 the PP/ST author should fill in the list of information fields with the fields within the information over which the attributes provide evidence of receipt, such as the body of a message.

Selection:

In FCO_NRR.2.3 the PP/ST author should specify the user/subjects who can verify the evidence of receipt.

Assignment:

In FCO_NRR.2.3 the PP/ST author, dependent on the selection, should specify the third parties that can verify the evidence of receipt. A third party could be an arbiter, judge or legal body.

In FCO_NRR.2.3 the PP/ST author should fill in the list of limitations under which the evidence can be verified. For example the evidence can only be verified within a 24 hour time interval. An assignment of ‘immediate’ or ‘indefinite’ is acceptable.

Annex E

Cryptographic support (FCS)

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

Figure E.1 shows the decomposition of this class into its constituent components.

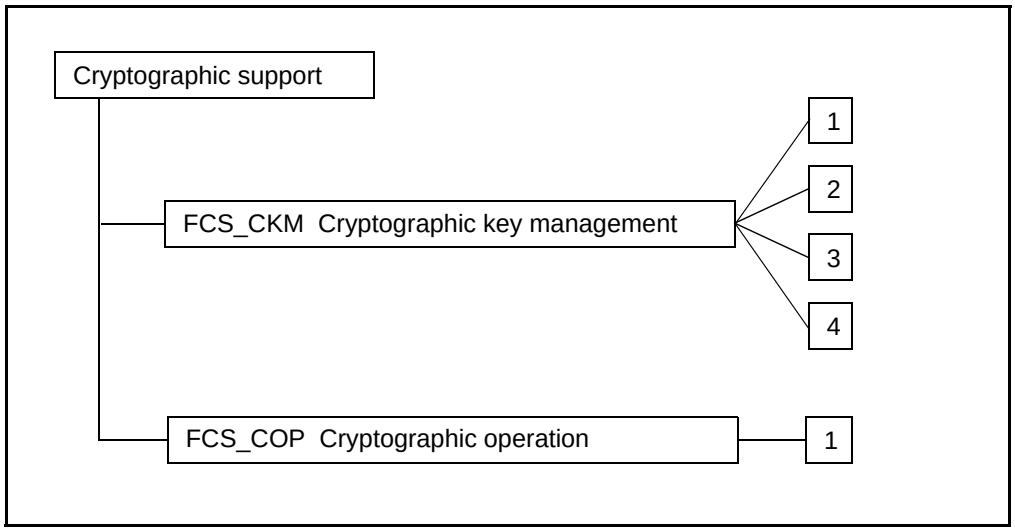

Figure E.1 - Cryptographic support class decomposition

For each cryptographic key generation method implemented by the TOE, if any, the PP/ST author should select the FCS_CKM.1 component.

For each cryptographic key distribution method implemented by the TOE, if any, the PP/ST author should select the FCS_CKM.2 component.

For each cryptographic key access method implemented by the TOE, if any, the PP/ST author should select the FCS_CKM.3 component.

For each cryptographic key destruction method implemented by the TOE, if any, the PP/ST author should select the FCS_CKM.4 component.

For each cryptographic operation (such as digital signature, data encryption, key agreement, secure hash, etc.) performed by the TOE, if any, the PP/ST author should select the FCS_COP.1 component.

Cryptographic functionality may be used to meet objectives specified in class FCO, and in families FDP_DAU, FDP_SDI, FDP_UCT, FDP_UIT, FIA_SOS, 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 should be used when cryptographic functionality of the TOE is sought by consumers.

E.1 Cryptographic key management (FCS_CKM)

User 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): generation, distribution, entry, storage, access (e.g. backup, escrow, archive, recovery) and destruction.

As a minimum, cryptographic keys should at least go through the following stages: generation, storage and destruction. The inclusion of other stages is dependent on the key management strategy being implemented, as the TOE need not be involved in all of the key life-cycle (e.g. the TOE may only generate and distribute cryptographic keys).

This family is intended to support the cryptographic key lifecycle and consequently defines requirements for the following activities: cryptographic key generation, cryptographic key distribution, cryptographic key access and cryptographic key destruction. This family should be included whenever there are functional requirements for the management of cryptographic keys.

If FAU_GEN Security Audit Data Generation is included in the PP/ST then, in the context of the events being audited:

a) The object attributes may include the assigned user for the cryptographic key, the user role, the cryptographic operation that the cryptographic key is to be used for, the cryptographic key identifier and the cryptographic key validity period.

b) The object value may include the values of cryptographic key(s) and parameters **excluding** any sensitive information (such as secret or private cryptographic keys).

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.

FCS_CKM.1 Cryptographic key generation

User application notes

This component requires the cryptographic key sizes and method used to generate cryptographic keys to be specified, this can be in accordance with an assigned standard. It should be used to specify the cryptographic key sizes and the method (e.g. algorithm) used to generate the cryptographic keys. Only one instance of the component is needed for the same method and multiple key sizes. The key size could be common or different for the various entities, and could be either the input to or the output from the method.

Operations

Assignment:

In FCS_CKM.1.1, the PP/ST author should specify the cryptographic key generation algorithm to be used.

In FCS_CKM.1.1, the PP/ST author should specify the cryptographic key sizes to be used. The key sizes specified should be appropriate for the algorithm and its intended use.

In FCS_CKM.1.1, the PP/ST author should specify the assigned standard that documents the method used to generate cryptographic keys. The assigned standard may comprise none, one or more actual standards publications, for example, from international, national, industry or organisational standards.

FCS_CKM.2 Cryptographic key distribution

User application notes

This component requires the method used to distribute cryptographic keys to be specified, this can be in accordance with an assigned standard.

Operations

Assignment:

In FCS_CKM.2.1, the PP/ST author should specify the cryptographic key distribution method to be used.

In FCS_CKM.2.1, the PP/ST author should specify the assigned standard that documents the method used to distribute cryptographic keys. The assigned standard may comprise none, one or more actual standards publications, for example, from international, national, industry or organisational standards.

FCS_CKM.3 Cryptographic key access

User application notes

This component requires the method used to access cryptographic keys be specified, this can be in accordance with an assigned standard.

Operations

Assignment:

In FCS_CKM.3.1, the PP/ST author should specify the type of cryptographic key access being used. Examples of types of cryptographic key access include (but are

not limited to) cryptographic key backup, cryptographic key archival, cryptographic key escrow and cryptographic key recovery.

In FCS_CKM.3.1, the PP/ST author should specify the cryptographic key access method to be used.

In FCS_CKM.3.1, the PP/ST author should specify the assigned standard that documents the method used to access cryptographic keys. The assigned standard may comprise none, one or more actual standards publications, for example, from international, national, industry or organisational standards.

FCS_CKM.4 Cryptographic key destruction

User application notes

This component requires the method used to destroy cryptographic keys be specified, this can be in accordance with an assigned standard.

Operations

Assignment:

In FCS_CKM.4.1, the PP/ST author should specify the key destruction method to be used to destroy cryptographic keys.

In FCS_CKM.4.1, the PP/ST author should specify the assigned standard that documents the method used to destroy cryptographic keys. The assigned standard may comprise none, one or more actual standards publications, for example, from international, national, industry or organisational standards.

E.2 Cryptographic operation (FCS_COP)

User notes

A cryptographic operation may have cryptographic mode(s) of operation associated with it. If this is the case, then the cryptographic mode(s) must be specified. Examples of cryptographic modes of operation are cipher block chaining, output feedback mode, electronic code book mode, and cipher feedback mode.

Cryptographic operations may be used to support one or more TOE security services. The 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 FAU_GEN Security audit data generation is included in the PP/ST then, in the context of the cryptographic operation events being audited:

a) The types of cryptographic operation may include digital signature generation and/or verification, cryptographic checksum generation for integrity and/or for verification of checksum, secure hash (message digest) computation, data encryption and/or decryption, cryptographic key encryption and/or decryption, cryptographic key agreement and random number generation.

b) The subject attributes may include subject role(s) and user(s) associated with the subject.

c) The object attributes may include the assigned user for the cryptographic key, user role, cryptographic operation the cryptographic key is to be used for, cryptographic key identifier, and the cryptographic key validity period.

FCS_COP.1 Cryptographic operation

User application notes

This component requires the cryptographic algorithm and key size used to perform specified cryptographic operation(s) which can be based on an assigned standard.

Operations

Assignment:

In FCS_COP.1.1, the PP/ST author should specify the cryptographic operations being performed. Typical cryptographic operations include digital signature generation and/or verification, cryptographic checksum generation for integrity and/or for verification of checksum, secure hash (message digest) computation, data encryption and/or decryption, cryptographic key encryption and/or

decryption, cryptographic key agreement and random number generation. The cryptographic operation may be performed on user data or TSF data.

In FCS_COP.1.1, the PP/ST author should specify the cryptographic algorithm to be used. Typical cryptographic algorithms include, but are not limited to, DES, RSA and IDEA.

In FCS_COP.1.1, the PP/ST author should specify the cryptographic key sizes to be used. The key sizes specified should be appropriate for the algorithm and its intended use.

In FCS_COP.1.1, the PP/ST author should specify the assigned standard that documents how the identified cryptographic operation(s) are performed. The assigned standard may comprise none, one or more actual standards publications, for example, from international, national, industry or organisational standards.

Annex F

User data protection (FDP)

This class contains families specifying requirements for TOE security functions and TOE security function policies related to protecting user data. This class differs from FIA and FPT in that FDP 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 does not explicitly deal with confidentiality, integrity, or availability, as all three are most often intertwined in the policy and mechanisms. However, the TOE security policy must adequately cover these three objectives in the PP/ST.

A final aspect of this class is that it specifies access control in terms of “operations”. An operation is defined as a specific type of access on a specific object. It depends on the level of abstraction of the PP/ST author whether these operations are described as “read” and/or “write” operations, or as more complex operations such as “update the database”.

The access control policies are policies that control access to the information container. The attributes represent attributes of the container. Once the information is out of the container, the accessor is free to modify that information, including writing the information into a different container with different attributes. By contrast, an information flow policies controls access to the information, independent of the container. The attributes of the information, which may be associated with the attributes of the container (or may not, as in the case of a multi-level database) stay with the information as it moves. The accessor does not have the ability, in the absence of an explicit authorisation, to change the attributes of the information.

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.

For example, one could imagine a goal of having user-imposed (and user-defined) controls on information flow (e.g. an automated implementation of the NO FOREIGN handling caveat). Such concepts could be handled as refinements of, or extensions to the FDP components.

Finally, it is important when looking at the components in FDP to remember that these components are requirements for functions that may be implemented by a mechanism that also serves or could serve another purpose. For example, it is possible to build an access control policy (FDP_ACC) that uses labels (FDP_IFF.1) as the basis of the access control mechanism.

A TOE security policy may encompass many security function policies (SFPs), each to be identified by the two policy oriented components FDP_ACC, and 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.

Figures F.1 and F.2 show the decomposition of this class into its constituent components.

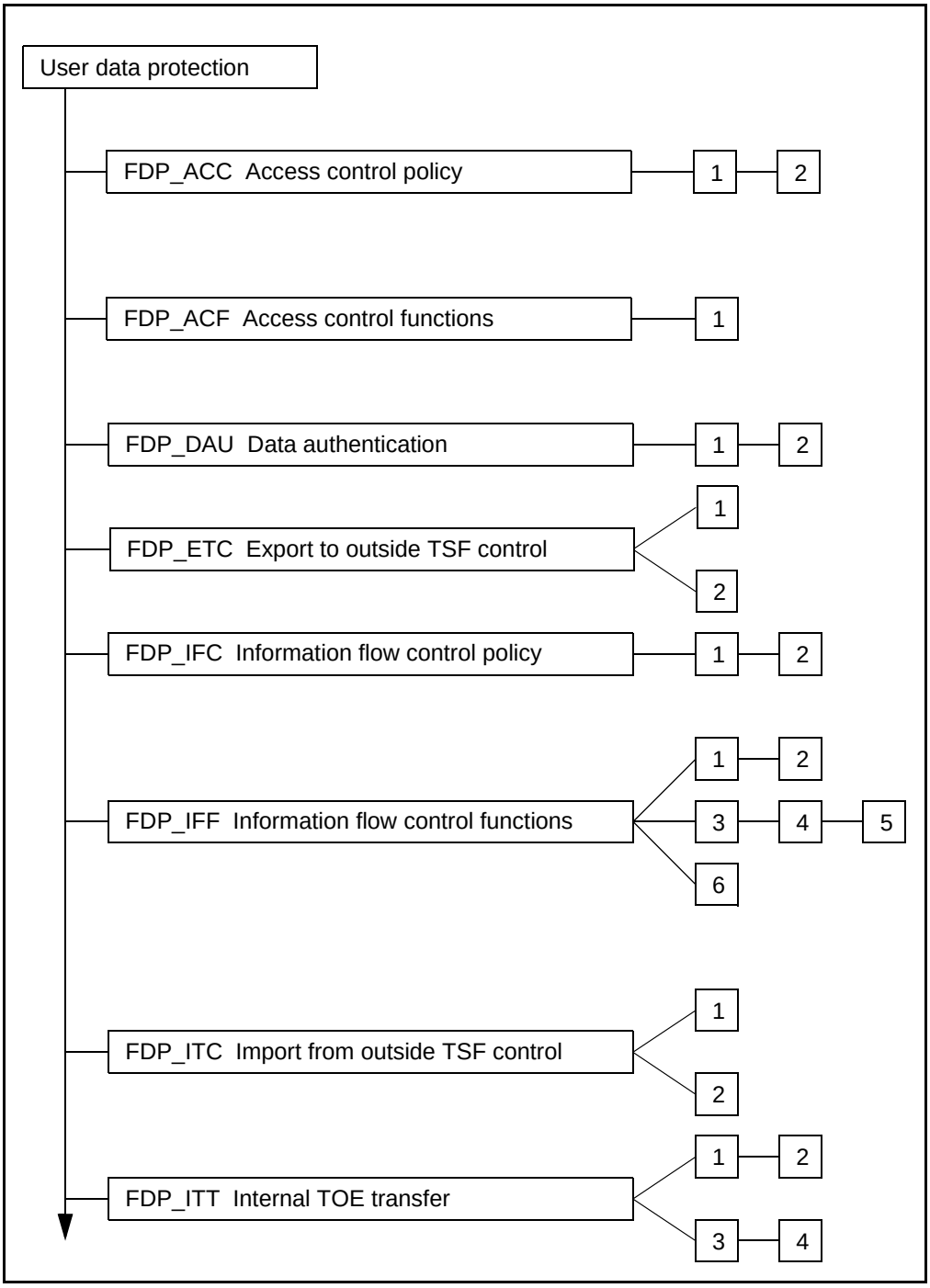

Figure F.1 - User data protection class decomposition

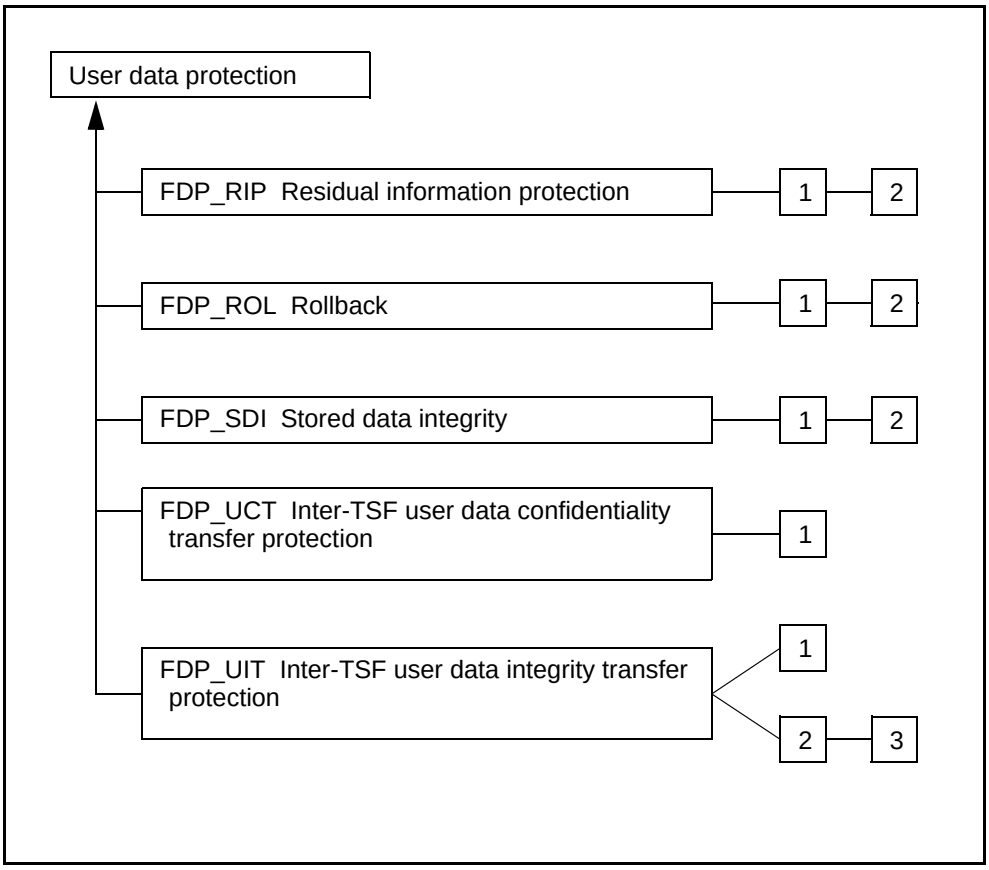

Figure F.2 - User data protection class decomposition (cont.)

When building a PP/ST using components from the FDP class, the following information provides guidance on where to look and what to select from the class.

The requirements in the FDP class are defined in terms of a security function (abbreviated SF) that will implement a SFP. Since a TOE may implement multiple SFPs simultaneously, the PP/ST author must specify the name for each SFP, so it can be referenced in other families. This name will then be used in each component selected to indicate that it is being used as part of the definition of requirements for that function. This allows the author to easily indicate the scope for operations such as objects covered, operations covered, authorised users, etc.

Each instantiation of a component can apply to only one SFP. Therefore if an SFP is specified in a component then this SFP will apply to all the elements in this component. The components may be instantiated multiple times within a PP/ST to account for different policies if so desired.

The key to selecting components from this family is to have a well defined TOE security policy to enable proper selection of the components from the two policy components; FDP_ACC and FDP_IFC. In FDP_ACC and FDP_IFC respectively, all access control policies and all information flow control policies are named. Furthermore the scope of control of these components in terms of the subjects, objects and operations covered by this security function. The names of these policies

are meant to be used throughout the remainder of the functional components that have an operation that calls for an assignment or selection of an “access control SFP” or an “information flow control SFP”. The rules that define the functionality of the named access control and information flow control SFPs will be defined in the FDP_ACF and FDP_IFF families (respectively).

The following steps are guidance on how this class is applied in the construction of a PP/ST:

a) Identify the policies to be enforced from the FDP_ACC, and 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 FDP_ACF and FDP_IFF families to address the named policy families from FDP_ACC and 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 Class FMT Security management and perform the operations. Refinements may be useful here to identify missing features, such as that some or all changes must be done via trusted path.

e) Identify any appropriate components from the Class FMT Security management for initial values for new objects and subjects.

f) Identify any applicable rollback components from the FDP_ROL family.

g) Identify any applicable residual information protection requirements from the FDP_RIP family.

h) Identify any applicable import or export components, and how security attributes should be handled during import and export, from the FDP_ITC and FDP_ETC families.

i) Identify any applicable internal TOE communication components from the FDP_ITT family.

j) Identify any requirements for integrity protection of stored information from the FDP_SDI.

k) Identify any applicable inter-TSF communication components from the FDP_UCT or FDP_UIT families.

F.1 Access control policy (FDP_ACC)

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.

User notes

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 FDP_ACF and FDP_RIP. The names of the access control SFPs defined in 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 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 Class FMT: Security management.

There are no audit requirements in 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/ST author the capability to specify several policies, for example, a fixed access control SFP to be applied to one scope of control, and a flexible access control SFP to be defined for a different scope of control. To specify more than one access control policy, the components from this family can be iterated multiple times in a PP/ST to different subsets of operations and objects. This will accommodate TOEs that contain multiple policies, each addressing a particular set of operations and objects. In other words, the PP/ST author should specify the required information in the ACC component for each of the access control SFPs that the TSF will enforce. For example, a TOE incorporating three access control SFPs, each covering only a subset of the objects, subjects, and operations within the TOE, will contain one FDP_ACC.1 Subset access control component for each of the three access control SFPs, necessitating a total of three FDP_ACC.1 components.

FDP_ACC.1 Subset access control

User application notes

The terms object and subject refer to generic elements in the TOE. For a policy to be implementable, the entities must be clearly identified. For a PP, the objects and operations might be expressed as types such as: named objects, data repositories, observe accesses, etc. For a

specific system these generic terms (subject, object) must be refined, e.g. files, registers, ports, daemons, open calls, etc.

This component specifies that the policy cover some well-defined set of operations on some subset of the objects. It places no constraints on any operations outside the set – including operations on objects for which other operations are controlled.

Operations

Assignment:

In FDP_ACC.1.1, the PP/ST author should specify a uniquely named access control SFP to be enforced by the TSF.

In FDP_ACC.1.1, the PP/ST author should specify the list of subjects, objects, and operations among subjects and objects covered by the SFP.

FDP_ACC.2 Complete access control

User application notes

This component requires that all possible operations on objects, that are included in the SFP, are covered by an access control SFP.

The PP/ST author must demonstrate that each combination of objects and subjects is covered by an access control SFP.

Operations

Assignment:

In FDP_ACC.2.1, the PP/ST author should specify a uniquely named access control SFP to be enforced by the TSF.

In FDP_ACC.2.1, the PP/ST author should specify the list of subjects and objects covered by the SFP. All operations among those subjects and objects will be covered by the SFP.

F.2 Access control functions (FDP_ACF)

This family describes the rules for the specific functions that can implement an access control policy named in FDP_ACC which also specifies the scope of control of the policy.

User notes

This family provides a PP/ST author the capability to describe the rules for access control. This results in a system where the access to objects will not change. An example of such an object is "Message of the Day", which is readable by all, and changeable only by the authorised administrator. This family also provides the PP/ST author with the ability to describe rules that provide for exceptions to the general access control rules. Such exceptions would either explicitly allow or deny authorisation to access an object.

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 SFs may be specified in this family such as:

- Access control lists (ACLs)
- Time-based access control specifications
- Origin-based access control specifications
- Owner-controlled access control attributes

FDP_ACF.1 Security attribute based access control

User application notes

This component provides requirements for a mechanism that mediates access control based on security attributes associated with subjects and objects. Each object and subject has a set of associated attributes, such as location, time of creation, access rights (e.g., Access Control Lists (ACLs)). This component allows the PP/ST author to specify the attributes that will be used for the access control mediation. This component allows access control rules, using these attributes, to be specified.

Examples of the attributes that a PP/ST author might assign are presented in the following paragraphs.

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 authorised 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 authorise or deny access to an object based upon security attributes. This could be used to provide privilege, access rights, or access authorisations within the TOE. Such privileges, rights, or authorisations could apply to users, subjects (representing users or applications), and objects.

Operations

Assignment:

In FDP_ACF.1.1, the PP/ST author should specify an access control SFP name that the TSF is to enforce. The name of the access control SFP, and the scope of control for that policy are defined in components from FDP_ACC.

In FDP_ACF.1.1, the PP/ST author should specify the security attributes and/or named groups of security attributes that the function will use in the specification of the rules. For example, such attributes may be things such as the user identity, subject identity, role, time of day, location, ACLs, or any other attribute specified by the PP/ST author. Named groups of security attributes can be specified to provide a convenient means to refer to multiple security attributes. Named groups could provide a useful way to associate "roles" defined in FMT_SMR Security management roles, 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/ST author should specify the SFP rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects. These rules specify when access is granted or denied. It can specify general access control functions (e.g. typical permission bits) or granular access control functions (e.g. ACLs).

In FDP_ACF.1.3, the PP/ST author should specify the rules, based on security attributes, that explicitly authorise access of subjects to objects that will be used to explicitly authorise access. These rules are in addition to those specified in FDP_ACF.1.1. They are included in FDP_ACF.1.3 as they are intended to contain exceptions to the rules in FDP_ACF.1.1. An example of rules to explicitly authorise access is based on a privilege vector associated with a subject that always grants access to objects covered by the access control SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify "none".

In FDP_ACF.1.4, the PP/ST author should specify the rules, based on security attributes, that explicitly deny access of subjects to objects. These rules are in addition to those specified in FDP_ACF.1.1. They are included in FDP_ACF.1.4 as they are intended to contain exceptions to the rules in FDP_ACF.1.1. An example of rules to explicitly deny access is based on a privilege vector associated with a subject that always denies access to objects covered by the access control

SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify “none”.

F.3 Data authentication (FDP_DAU)

This family describes specific functions that can be used to authenticate ‘static’ data.

User notes

Components in this family are to be used when there is a requirement for ‘static’ data authentication, i.e. where data is to be signed but not transmitted. (Note that the FCO_NRO family provides for non-repudiation of origin of information received during a data exchange.)

FDP_DAU.1 Basic data authentication

User application notes

This component may be satisfied by one-way hash functions (cryptographic checksum, fingerprint, message digest), to generate a hash value for a definitive document that may be used as verification of the validity or authenticity of its information content.

Operations

Assignment:

In FDP_DAU.1.1, the PP/ST author should specify the list of objects or information types for which the TSF shall be capable of generating data authentication evidence.

In FDP_DAU.1.2, the PP/ST author should specify the list of subjects that will have the ability to verify data authentication evidence for the objects identified in the previous element. The list of subjects could be very specific, if the subjects are known, or it could be more generic and refer to a “type” of subject such as an identified role.

FDP_DAU.2 Data authentication with identity of guarantor

User application notes

This component additionally requires the ability to verify the identity of the user that provided the guarantee of authenticity (e.g. a trusted third party).

Operations

Assignment:

In FDP_DAU.2.1, the PP/ST author should specify the list of objects or information types for which the TSF shall be capable of generating data authentication evidence.

In FDP_DAU.2.2, the PP/ST author should specify the list of subjects that will have the ability to verify data authentication evidence for the objects identified in the

previous element **as well as the identity of the user that created the data authentication evidence**.

F.4 Export to outside TSF control (FDP_ETC)

This family defines functions for exporting user data from the TOE such that its security attributes either can be explicitly preserved or can be ignored once it has been exported. Consistency of these security attributes are addressed by FPT_TDC Inter-TSF TSF data consistency.

FDP_ETC is concerned with limitations on export and association of security attributes with the exported user data.

User notes

This family, and the corresponding Import family FDP_ITC, address how the TOE deals with user data transferred into and outside its control. In principle this family is concerned with the export of user data and its related security attributes.

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.

FDP_ETC.1 Export of user data without security attributes

User application notes

This component is used to specify the export of user data without the export of its security attributes.

Operations

Assignment:

In FDP_ETC.1.1, the PP/ST author should specify the access control SFP(s) and/ or information flow control SFP(s) that will be enforced when exporting user data. The user data that this function exports is scoped by the assignment of these SFPs.

FDP_ETC.2 Export of user data with security attributes

User application notes

The user data is exported together with its security attributes. The security attributes are unambiguously associated with the user data. There are several ways of achieving this association. One way that this can be achieved is by physically collocating the user data and the security attributes (e.g. the same floppy), or by using cryptographic techniques such as secure signatures to associate the attributes and the user data. FTP_ITC Inter-TSF trusted channel could be used to

assure that the attributes are correctly received at the other Trusted IT Product while FPT_TDC Inter-TSF TSF data consistency can be used to make sure that those attributes are properly interpreted. Furthermore, FTP_TRP Trusted path could be used to make sure that the export is being initiated by the proper user.

Operations

Assignment:

In FDP_ETC.2.1, the PP/ST author should specify the access control SFP(s) and/ or information flow control SFP(s) that will be enforced when exporting user data. The user data that this function exports is scoped by the assignment of these SFPs.

In FDP_ETC.2.4, the PP/ST author should specify any additional exportation control rules or "none" if there are no additional exportation control rules. These rules will be enforced by the TSF in addition to the access control SFPs and/or information flow control SFPs selected in FDP_ETC.2.1.

F.5 Information flow control policy (FDP_IFC)

This family covers the identification of information flow control SFPs; and, for each, specifies the scope of control of the SFP.

Examples of security policies that might satisfy this objective are:

- Bell and La Padula Security model [B&L];
- Biba Integrity model [Biba];
- Non-Interference [Gogu1,Gogu2].

User notes

The components in this family are capable of identifying the information flow control SFPs to be enforced by the traditional Mandatory Access Control mechanisms that would be found in a TOE. However, they go beyond just the traditional MAC mechanisms and can be used to identify and describe non-interference policies and state-transitions. It further defines the subjects under control of the policy, the information under control of the policy, and operations which cause controlled information to flow to and from controlled subjects for each information flow control SFP in the TOE. The functionality that defines the rules of an information flow control SFP will be defined by other families such as FDP_IFF and FDP_RIP. The access control SFPs named here in 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 must 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 authorisation, 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/ST to different subsets of operations and objects. This will accommodate TOEs that contain multiple policies, each addressing a particular set of objects, subjects, and operations.

FDP_IFC.1 Subset information flow control

User application notes

This component requires that an information flow control policy apply to a subset of the possible operations in the TOE.

Operations

Assignment:

In FDP_IFC.1.1, the PP/ST author should specify a uniquely named information flow control SFP to be enforced by the TSF.

In FDP_IFC.1.1, the PP/ST author should specify the list of subjects, information, and operations which cause controlled information to flow to and from controlled subjects covered by the SFP. As mentioned above, the list of subjects could be at various levels of detail depending on the needs of the PP/ST author. It could specify users, machines, or processes for example. Information could refer to data such as email or network protocols, or more specific objects similar to those specified under an access control policy. If the information that is specified is contained within an object that is subject to an access control policy, then both the access control policy and information flow control policy must be enforced before the specified information could flow to or from the object.

FDP_IFC.2 Complete information flow control

User application notes

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/ST author must demonstrate that each combination of information flows and subjects is covered by an information flow control SFP.

Operations

Assignment:

In FDP_IFC.2.1, the PP/ST author should specify a uniquely named information flow control SFP to be enforced by the TSF.

In FDP_IFC.2.1, the PP/ST author should specify the list of subjects and information that will be covered by the SFP. All operations that cause that information to flow to and from subjects will be covered by the SFP. As mentioned above, the list of subjects could be at various levels of detail depending on the needs of the PP/ST author. It could specify users, machines, or processes for example. Information could refer to data such as email or network protocols, or more specific objects similar to those specified under an access control policy. If the information that is specified is contained within an object that is subject to an access control policy, then both the access control policy and information flow control policy must be enforced before the specified information could flow to or from the object.

F.6 Information flow control functions (FDP_IFF)

This family describes the rules for the specific functions that can implement the information flow control SFPs named in 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.

User notes

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 categorisation of flows as “Storage Channels” or “Timing Channels”, or it can refer to improved categorisations reflective of the needs of a PP/ST author.

The flexibility of these components allows the definition of a privilege policy within FDP_IFF.1 and FDP_IFF.2 to allow the controlled bypass of all or part of a particular SFP. If there is a need for a predefined approach to SFP bypass, the PP/ST author should consider incorporating a privilege policy.

FDP_IFF.1 Simple security attributes

User application notes

This component requires security attributes on information, and on subjects that cause that information to flow and subjects that act as recipients of that information. The attributes of the containers of the information should also be considered if it is desired that they should play a part in information flow control decisions or if they are covered by an access control policy. This component specifies the key rules that are enforced, and describes how security attributes are derived. For example, this component should be used when at least one of the information flow control SFPs in the TSP is based on labels as defined in the Bell and LaPadula security policy model [B&L], but these security attributes do not form a hierarchy.

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 authorise and deny an information flow based upon security attributes. This could be used to implement a privilege policy that covers exceptions to the basic policy defined in this component.

Operations

Assignment:

In FDP_IFF.1.1, the PP/ST author should specify the information flow control SFPs enforced by the TSF. The name of the information flow control SFP, and the scope of control for that policy are defined in components from FDP_IFC.

In FDP_IFF.1.1 the PP/ST author should specify the minimum number and type of security attributes that the function will use in the specification of the rules. For example, such attributes may be things such as subject identifier, subject sensitivity level, subject clearance level, information sensitivity level, etc. The minimum number of each type of security attribute should be sufficient to support the environmental needs.

In FDP_IFF.1.2 the PP/ST author should specify for each operation, the security attribute-based relationship that must hold between subject and information security attributes that the TSF will enforce.

In FDP_IFF.1.3 the PP/ST author should specify any additional information flow control SFP rules that the TSF is to enforce. If there are no additional rules then the PP/ST author should specify "none".

In FDP_IFF.1.4 the PP/ST author should specify any additional SFP capabilities that the TSF is to provide. If there are no additional capabilities then the PP/ST author should specify "none".

In FDP_IFF.1.5, the PP/ST author should specify the rules, based on security attributes, that explicitly authorise information flows. These rules are in addition to those specified in the preceding elements. They are included in FDP_IFF.1.5 as they are intended to contain exceptions to the rules in the preceding elements. An example of rules to explicitly authorise information flows is based on a privilege vector associated with a subject that always grants the subject the ability to cause an information flow for information that is covered by the SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify "none".

In FDP_IFF.1.6, the PP/ST author should specify the rules, based on security attributes, that explicitly deny information flows. These rules are in addition to those specified in the preceding elements. They are included in FDP_IFF.1.6 as they are intended to contain exceptions to the rules in the preceding elements. An example of rules to explicitly authorise information flows is based on a privilege vector associated with a subject that always denies the subject the ability to cause an information flow for information that is covered by the SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify "none".

FDP_IFF.2 Hierarchical security attributes

User application notes

This component requires that all information flow control SFPs in the TSP use hierarchical security attributes that form a lattice.

For example, it should be used when at least one of the information flow control SFPs in the TSP is based on labels as defined in the Bell and LaPadula security policy model [B&L] and form a hierarchy.

It is important to note that the hierarchical relationship requirements identified in FDP_IFF.2.5 need only apply to the information flow control security attributes for the information flow control SFPs that have been identified in FDP_IFF.2.1. This component is not meant to apply to other SFPs such as access control SFPs.

Like the preceding component, this component could also be used to implement a privilege policy that covers rules that allow for the explicit authorisation or denial of information flows.

If it is the case that multiple information flow control SFPs are to be specified, and that each of these SFPs will have their own security attributes that are not related to one another, then the PP/ST author should iterate this component once for each of those SFPs. Otherwise a conflict might arise with the sub-items of FDP_IFF.2.5 since the required relationships will not exist.

Operations

Assignment:

In FDP_IFF.2.1, the PP/ST author should specify the information flow control SFPs enforced by the TSF. The name of the information flow control SFP, and the scope of control for that policy are defined in components from FDP_IFC.

In FDP_IFF.2.1 the PP/ST author should specify the minimum number and type of security attributes that the function will use in the specification of the rules. For example, such attributes may be things such as subject identifier, subject sensitivity level, subject clearance level, information sensitivity level, etc. The minimum number of each type of security attribute should be sufficient to support the environmental needs.

In FDP_IFF.2.2 the PP/ST author should specify for each operation, the security attribute-based relationship that must hold between subject and information security

attributes that the TSF will enforce. **These relationships should be based upon the ordering relationships between the security attributes.**

In FDP_IFF.2.3 the PP/ST author should specify any additional information flow control SFP rules that the TSF is to enforce. If there are no additional rules then the PP/ ST author should specify “none”.

In FDP_IFF.2.4 the PP/ST author should specify any additional SFP capabilities that the TSF is to enforce. If there are no additional rules then the PP/ST author should specify “none”.

In FDP_IFF.2.5, the PP/ST author should specify the rules, based on security attributes, that explicitly authorise information flows. These rules are in addition to those specified in the preceding elements. They are included in FDP_IFF.2.5 as they are intended to contain exceptions to the rules in the preceding elements. An example of rules to explicitly authorise information flows is based on a privilege vector associated with a subject that always grants the subject the ability to cause an information flow for information that is covered by the SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify “none”.

In FDP_IFF.2.6, the PP/ST author should specify the rules, based on security attributes, that explicitly deny information flows. These rules are in addition to those specified in the preceding elements. They are included in FDP_IFF.2.6 as they are intended to contain exceptions to the rules in the preceding elements. An example of rules to explicitly authorise information flows is based on a privilege vector associated with a subject that always denies the subject the ability to cause an information flow for information that is covered by the SFP that has been specified. If such a capability is not desired, then the PP/ST author should specify “none”.

FDP_IFF.3 Limited illicit information flows

User application notes

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/ST author has the ability to specify whether the illicit information flows must be audited.

Operations

Assignment:

In FDP_IFF.3.1 the PP/ST author should specify the information flow control SFPs enforced by the TSF. The name of the information flow control SFP, and the scope of control for that policy are defined in components from FDP_IFC.

In FDP_IFF.3.1 the PP/ST author should specify the types of illicit information flows that are subject to a maximum capacity limitation.

In FDP_IFF.3.1 the PP/ST author should specify the maximum capacity permitted for any identified illicit information flows.

FDP_IFF.4 Partial elimination of illicit information flows

User application notes

This component should be used when all the SFPs that requires control of illicit information flows require elimination of some (but not necessarily all) illicit information flows.

Operations

Assignment:

In FDP_IFF.4.1 the PP/ST author should specify the information flow control SFPs enforced by the TSF. The name of the information flow control SFP, and the scope of control for that policy are defined in components from FDP_IFC.

In FDP_IFF.4.1 the PP/ST author should specify the types of illicit information flows which are subject to a maximum capacity limitation.

In FDP_IFF.4.1 the PP/ST author should specify the maximum capacity permitted for any identified illicit information flows.

In FDP_IFF.4.2 the PP/ST author should specify the types of illicit information flows to be eliminated. This list may not be empty as this component requires that some illicit information flows are to be eliminated.

FDP_IFF.5 No illicit information flows

User application notes

This component should be used when the SFPs that require control of illicit information flows require elimination of all illicit information flows. However, the PP/ST author should carefully consider the potential impact that eliminating all illicit information flows might have on the normal functional operation of the TOE. Many practical applications have shown that there is an indirect relationship between illicit information flows and normal functionality within a TOE and eliminating all illicit information flows may result in less than desired functionality.

Operations

Assignment:

In FDP_IFF.5.1 the PP/ST author should specify the information flow control SFP for which illicit information flows are to be eliminated. The name of the information flow control SFP, and the scope of control for that policy are defined in components from FDP_IFC.

FDP_IFF.6 Illicit Information Flow Monitoring

User 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 FAU_GEN Security audit data generation family.

Operations

Assignment:

In FDP_IFF.6.1 the PP/ST author should specify the information flow control SFPs enforced by the TSF. The name of the information flow control SFP, and the scope of control for that policy are defined in components from FDP_IFC.

In FDP_IFF.6.1 the PP/ST author should specify the types of illicit information flows that will be monitored for exceeding a maximum capacity.

In FDP_IFF.6.1 the PP/ST author should specify the maximum capacity above which illicit information flows will be monitored by the TSF.

F.7 Import from outside TSF control (FDP_ITC)

This family defines mechanisms for importing user data from outside the TSC into the TOE such that the user data security attributes can be preserved. Consistency of these security attributes are addressed by FPT_TDC Inter-TSF TSF data consistency.

FDP_ITC is concerned with limitations on import, user specification of security attributes, and association of security attributes with the user data.

User notes

This family, and the corresponding export family 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.

A variety of activities might be involved here:

a) importing user data from an unformatted medium (e.g. floppy disk, tape, scanner, video or audit signal), without including any security attributes, and physically marking the medium to indicate its contents;

b) importing user data, including security attributes, from a medium and verifying that the object security attributes are appropriate;

c) importing user data, including security attributes, from a medium using a cryptographic sealing technique to protect the association of user data and security attributes.

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, e.g. cryptographic checksum).

This family is concerned with importing user data and maintaining the association of security attributes as required by the SFP. Other families are concerned with other import aspects such as consistency, trusted channels, and integrity that are beyond the scope of this family. Furthermore, FDP_ITC is only concerned with the interface to the import medium. 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 organisational security policy. For example, if user data is received on a “confidential” channel, the security attributes of the objects will be set to “confidential”.

If there are multiple SFPs (access control and/or information flow control) then it may be appropriate to iterate these components once for each named SFP.

FDP_ITC.1 Import of User Data Without Security Attributes

User application notes

This component is used to specify the import of user data that does not have reliable (or any) security attributes associated with it. This function requires that the security attributes for the imported user data be initialised within the TSF. It could also be the case that the PP/ST author specifies the rules for import. It may be appropriate, in some environments, to require that these attributes be supplied via a Trusted Path or a Trusted Channel mechanism.

Operations

Assignment:

In FDP_ITC.1.1, the PP/ST author should specify the access control SFP and/or information flow control SFP that will be enforced when importing user data from outside of the TSC. The user data that this function imports is scoped by the assignment of these SFPs.

In FDP_ITC.1.3, the PP/ST author should specify any additional importation control rules or “none” if there are no additional importation control rules. These rules will be enforced by the TSF in addition to the access control SFPs and/or information flow control SFPs selected in FDP_ITC.1.1.

FDP_ITC.2 Import of User Data with Security Attributes

User application notes

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 FPT_TDC to ensure the consistency of the data. It could also be the case that the PP/ST author specifies the rules for import.

Operations

Assignment:

In FDP_ITC.2.1, the PP/ST author should specify the access control SFP and/or information flow control SFP that will be enforced when importing user data from outside of the TSC. The user data that this function imports is scoped by the assignment of these SFPs

In FDP_ITC.2.5, the PP/ST author should specify any additional importation control rules or "none" if there are no additional importation control rules. These rules will be enforced by the TSF in addition to the access control SFPs and/or information flow control SFPs selected in FDP_ITC.2.1.

F.8 Internal TOE transfer (FDP_ITT)

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 FDP_UCT and FDP_UIT family, which provide protection for user data when it is transferred between distinct TSFs across an external channel, and FDP_ETC and FDP_ITC, which address transfer of data to or from outside the TSF's Control.

User notes

The requirements in this family allow a PP/ST author to specify the desired security for user data while in transit within the TOE. This security could be protection against disclosure, modification, or loss of availability.

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.

FDP_ITT.1 Basic internal transfer protection

Operations

Assignment:

In FDP_ITT.1.1, the PP/ST author should specify the access control SFP(s) and/ or information flow control SFP(s) covering the information being transferred.

Selection:

In FDP_ITT.1.1 the PP/ST author should specify the types of transmission errors that the TSF should prevent occuring for user data while in transport. The options are disclosure, modification, loss of use.

FDP_ITT.2 Transmission separation by attribute

User application notes

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.

Operations

Assignment:

In FDP_ITT.2.1, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) covering the information being transferred.

Selection:

In FDP_ITT.2.1 the PP/ST author should specify the types of transmission errors that the TSF should prevent occuring for user data while in transport. The options are disclosure, modification, loss of use.

Assignment:

In FDP_ITT.2.2, the PP/ST author should specify the security attributes, the values of which the TSF will use to determine when to separate data that is being trasmitted between physically-separated parts of the TOE. An example is that user data associated with the identity of one owner is transmitted separately from the user data associated with the identify of a different owner. In this case, the value of the identity of the owner of the data is what is used to determine when to separate the data for transmission.

FDP_ITT.3 Integrity monitoring

User application notes

This component is used in combination with either FDP_ITT.1 or FDP_ITT.2. It ensures that the TSF checks received user data (and their attributes) for integrity. FDP_ITT.1 or FDP_ITT.2 will provide the data in a manner such that it is protected from modification (so that FDP_ITT.3 can detect any modifications).

The PP/ST author has to specify the types of errors that must be detected. The PP/ST author should consider: modification of data, substitution of data, unrecoverable ordering change of data, replay of data, incomplete data, in addition to other integrity errors.

The PP/ST author must specify the actions that the TSF should take on detection of a failure. For example: ignore the user data, request the data again, inform the authorised administrator, reroute traffic for other lines.

Operations

Assignment:

In FDP_ITT.3.1, the PP/ST author should specify the access control SFP(s) and/ or information flow control SFP(s) covering the information being transferred and monitored for integrity errors.

In FDP_ITT.3.1, the PP/ST author should specify the type of possible integrity errors to be monitored during transmission of the user data.

In FDP_ITT.3.2, the PP/ST author should specify the action to be taken by the TSF when an integrity error is encountered. An example might be that the TSF should request the resubmission of the user data. The SFP(s) specified in FDP_ITT.3.1 will be enforced as the actions are taken by the TSF.

FDP_ITT.4 Attribute-based integrity monitoring

This component is used in combination with FDP_ITT.2. It ensures that the TSF checks received user data, that has been transmitted by separate channels (based on values of specified security attributes), for integrity. It allows the PP/ST author to specify actions to be taken upon detection of an integrity error.

For example, this component could be used to provide different integrity error detection and action for information at different integrity levels.

The PP/ST author has to specify the types of errors that must be detected. The PP/ST author should consider: modification of data, substitution of data, unrecoverable ordering change of data, replay of data, incomplete data, in addition to other integrity errors.

The PP/ST author should specify the attributes (and associated transmission channels) that necessitate integrity error monitoring

The PP/ST author must specify the actions that the TSF should take on detection of a failure. For example: ignore the user data, request the data again, inform the authorised administrator, reroute traffic for other lines.

Operations

Assignment:

In FDP_ITT.4.1, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) covering the information being transferred and monitored for integrity errors.

In FDP_ITT.4.1, the PP/ST author should specify the type of possible integrity errors to be monitored during transmission of the user data.

In FDP_ITT.4.1, the PP/ST author should specify a list of security attributes that require separate transmission channels. This list is used to determine which user data to monitor for integrity errors., based on its security attributes and its

transmission channel. This element is directly related to FDP_ITT.2 Transmission separation by attribute.

In FDP_ITT.4.2, the PP/ST author should specify the action to be taken by the TSF when an integrity error is encountered. An example might be that the TSF should request the resubmission of the user data. The SFP(s) specified in FDP_ITT.3.1 will be enforced as the actions are taken by the TSF.

F.9 Residual information protection (FDP_RIP)

This family addresses the need to ensure that deleted information is no longer accessible, and that newly-created objects do not contain information from previously used objects within the TOE. This family does not address objects stored off-line.

User notes

This family requires protection for information that has been logically deleted or released (not available to the user but still within the system and may be recoverable). In particular, this includes information that is contained in an object, as part of the TSF reusable resources, where destruction of the object does not necessarily equate to destruction of the resource or any contents of the resource.

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, FDP_RIP could apply to such events and resources.

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 FDP_RIP even though it is still part of object “B”.

It is important to note that 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, FDP_RIP can be instantiated to require that no residual information exists upon deallocation; however, the TSF cannot extend this enforcement to that same file that exists on the off-line back-up. Therefore that same file is still available. If this is a concern, then the PP/ST author should make sure that the proper environmental objectives are in place to support administrative guidance to address off-line objects.

FDP_RIP and FDP_ROL can conflict when 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 FDP_RIP selection of “deallocation” should not be used with FDP_ROL since there would be no information to roll back. The other selection, “unavailability upon allocation”, may be used with 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 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/ST author.

FDP_RIP.1 Subset residual information protection

User 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.

Operations

Selection:

In FDP_RIP.1.1, the PP/ST author should specify the event, allocation of the resource to or deallocation of the resource from, that invokes the residual information protection function.

Assignment:

In FDP_RIP.1.1, the PP/ST author should specify the list of objects subject to residual information protection.

FDP_RIP.2 Full residual information protection

User application notes

This component requires that for **all objects** in the TOE, the TSF will ensure that there is no available residual information contained in a resource allocated to those objects or deallocated from those objects.

Operations

Selection:

In FDP_RIP.2.1, the PP/ST author should specify the event, allocation of the resource to or deallocation of the resource from, that invokes the residual information protection function.

F.10 Rollback (FDP_ROL)

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.

FDP_RIP and FDP_ROL conflict when 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 FDP_RIP cannot be combined with FDP_ROL as there would be no information to roll back. FDP_RIP can be used only with 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 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. For example a text editor typically only allows you roll back up to a certain number of commands. Another example would be backups. If backup tapes are rotated, after a tape is reused, the information can no longer be retrieved. This also poses a bound on the rollback requirement.

FDP_ROL.1 Basic rollback

User application notes

This component allows a user or subject to undo a set of operations on a predefined set of objects. The undo is only possible within certain limits, for example up to a number of characters or up to a time limit.

Operations

Assignment:

In FDP_ROL.1.1, the PP/ST author should specify the access control SFP(s) and/ or information flow control SFP(s) that will be enforced when performing rollback operations. This is necessary to make sure that roll back is not used to circumvent the specified SFPs.

In FDP_ROL.1.1 the PP/ST author should specify the list of operations that can be rolled back.

In FDP_ROL.1.1 the PP/ST author should specify the list of objects that are subjected to the rollback policy.

In FDP_ROL.1.2 the PP/ST author should specify the boundary limit to which rollback operations may be performed. The boundary may be specified as a predefined period of time, for example, operations may be undone which were

performed within the past two minutes. Other possible boundaries may be defined as the maximum number of operations allowable or the size of a buffer.

FDP_ROL.2 Advanced rollback

User application notes

This component enforces that the TSF provide the capability to rollback all operations; however, the user can choose to rollback only a part of them.

Operations

Assignment:

In FDP_ROL.2.1, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced when performing rollback operations. This is necessary to make sure that roll back is not used to circumvent the specified SFPs.

In FDP_ROL.2.1 the PP/ST author should specify the list of objects that are subjected to the rollback policy.

In FDP_ROL.2.2 the PP/ST author should specify the boundary limit to which rollback operations may be performed. The boundary may be specified as a predefined period of time, for example, operations may be undone which were performed within the past two minutes. Other possible boundaries may be defined as the maximum number of operations allowable or the size of a buffer.

F.11 Stored data integrity (FDP_SDI)

This family provides requirements that address protection of user data while it is stored within the TSC.

User notes

Hardware glitches or errors may affect data stored in memory. This family provides requirements to detect these unintentional errors. The integrity of user data while stored on storage devices within the TSC are also addressed by this family.

To prevent a subject from modifying the data, the FDP_IFF or FDP_ACF families are required (rather than this family).

This family differs from FDP_ITT Internal TOE transfer that protects the user data from integrity errors while being transferred within the TOE.

FDP_SDI.1 Stored data integrity monitoring

User application notes

This component monitors data stored on media for integrity errors. The PP/ST author can specify different kinds of user data attributes that will be used as the basis for monitoring.

Operations

Assignment:

In FDP_SDI.1.1 the PP/ST author should specify the integrity errors that the TSF will detect.

In FDP_SDI.1.1 the PP/ST author should specify the user data attributes that will be used as the basis for the monitoring.

FDP_SDI.2 Stored data integrity monitoring and action

User application notes

This component monitors data stored on media for integrity errors. The PP/ST author can specify which action should be taken in case an integrity error is detected.

Operations

Assignment:

In FDP_SDI.2.1 the PP/ST author should specify the integrity errors that the TSF will detect.

In FDP_SDI.2.1 the PP/ST author should specify the user data attributes that will be used as the basis for the monitoring.

In FDP_SDI.2.2 the PP/ST author should specify the actions to be taken in case an integrity error is detected.

F.12 Inter-TSF user data confidentiality transfer protection (FDP_UCT)

This family defines the requirements for ensuring the confidentiality of user data when it is transferred using an external channel between the TOE and another Trusted IT Product. Confidentiality is enforced by preventing unauthorised disclosure of user data in transit between the two end points. The end points may be a TSF or a user.

User notes

This family provides a requirement for the protection of user data during transit. In contrast, FTP_ITC handles TSF data.

FDP_UCT.1 Basic data exchange confidentiality

User application notes

The TSF has the ability to protect from disclosure some user data which is exchanged.

Operations

Assignment:

In FDP_UCT.1.1, the PP/ST author should specify the access control SFP(s) and/ or information flow control SFP(s) that will be enforced when exchanging user data. The specified policies will be enforced to make decisions about who can exchange data and which data can be exchanged.

Selection:

In FDP_UCT.1.1, the PP/ST author should specify whether this element applies to a mechanism that transmits or receives user data.

F.13 Inter-TSF user data integrity transfer protection (FDP_UIT)

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.

User notes

This family defines the requirements for providing integrity for user data in transit; while FPT_ITI handles TSF data.

FDP_UIT and FDP_UCT are duals of each other, as FDP_UCT addresses user data confidentiality. Therefore, the same mechanism that implements FDP_UIT could possibly be used to implement other families such as FDP_UCT and FDP_ITC.

FDP_UIT.1 Data exchange integrity

User application notes

The TSF has a basic ability to send or receive user data in a manner such that modification of the user data can be detected. There is no requirement for a TSF mechanism to attempt to recover from the modification.

Operations

Assignment:

In FDP_UIT.1.1, the PP/ST author should specify the access control SFP(s) and/ or information flow control SFP(s) that will be enforced on the transmitted data or on the received data. The specified policies will be enforced to make decisions about who can transmit or who can receive data, and which data can be transmitted or received.

Selection:

In FDP_UIT.1.1, the PP/ST author should specify whether this element applies to a TSF that is transmitting or receiving objects.

In FDP_UIT.1.1 the PP/ST author should specify whether the data should be protected from modification, deletion, insertion or replay.

In FDP_UIT.1.2 the PP/ST author should specify whether the errors of the type: modification, deletion, insertion or replay are detected.

FDP_UIT.2 Source data exchange recovery

User application notes

This component provides the ability to recover from a set of identified transmission errors, if required, with the help of the other trusted IT product. As the other trusted IT product is outside the TSC, the TSF cannot control its behaviour. However, it can provide functions that have the ability to cooperate with the other trusted IT product for the purposes of recovery. For example, the TSF could include functions that depend upon the source trusted IT product to re-send the data in the event that an error is detected. This component deals with the ability of the TSF to handle such an error recovery.

Operations

Assignment:

In FDP_UIT.2.1, the PP/ST author should specify the access control SFP(s) and/ or information flow control SFP(s) that will be enforced when recovering user data. The specified policies will be enforced to make decisions about which data can be recovered and how it can be recovered.

In FDP_UIT.2.1, the PP/ST author should specify the list of integrity errors from which the TSF, with the help of the source Trusted IT Product, is be able to recover the original user data.

FDP_UIT.3 Destination data exchange recovery

User application notes

This component provides the ability to recover from a set of identified transmission errors. It accomplishes this task without help from the source trusted IT product. For example, if certain errors are detected, the transmission protocol must be robust enough to allow the TSF to recover from the error based on checksums and other information available within that protocol.

Operations

Assignment:

In FDP_UIT.3.1, the PP/ST author should specify the access control SFP(s) and/or information flow control SFP(s) that will be enforced when recovering user data. The specified policies will be enforced to make decisions about which data can be recovered and how it can be recovered.

In FDP_UIT.3.1, the PP/ST author should specify the list of integrity errors from which the **receiving** TSF, **alone**, is able to recover the original user data.

Annex G

Identification and authentication (FIA)

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 (e.g. identity, groups, roles, security or integrity levels).

The unambiguous identification of authorised users and the correct association of security attributes with users and subjects is critical to the enforcement of the security policies.

The FIA_UID family addresses determining the identity of a user.

The FIA_UAU family addresses verifying the identity of a user.

The FIA_AFL family addresses defining limits on repeated unsuccessful authentication attempts.

The FIA_ATD family address the definition of user attributes that are used in the enforcement of the TSP.

The FIA_USB family addresses the correct association of security attributes for each authorised user.

The FIA_SOS family addresses the generation and verification of secrets that satisfy a defined metric.

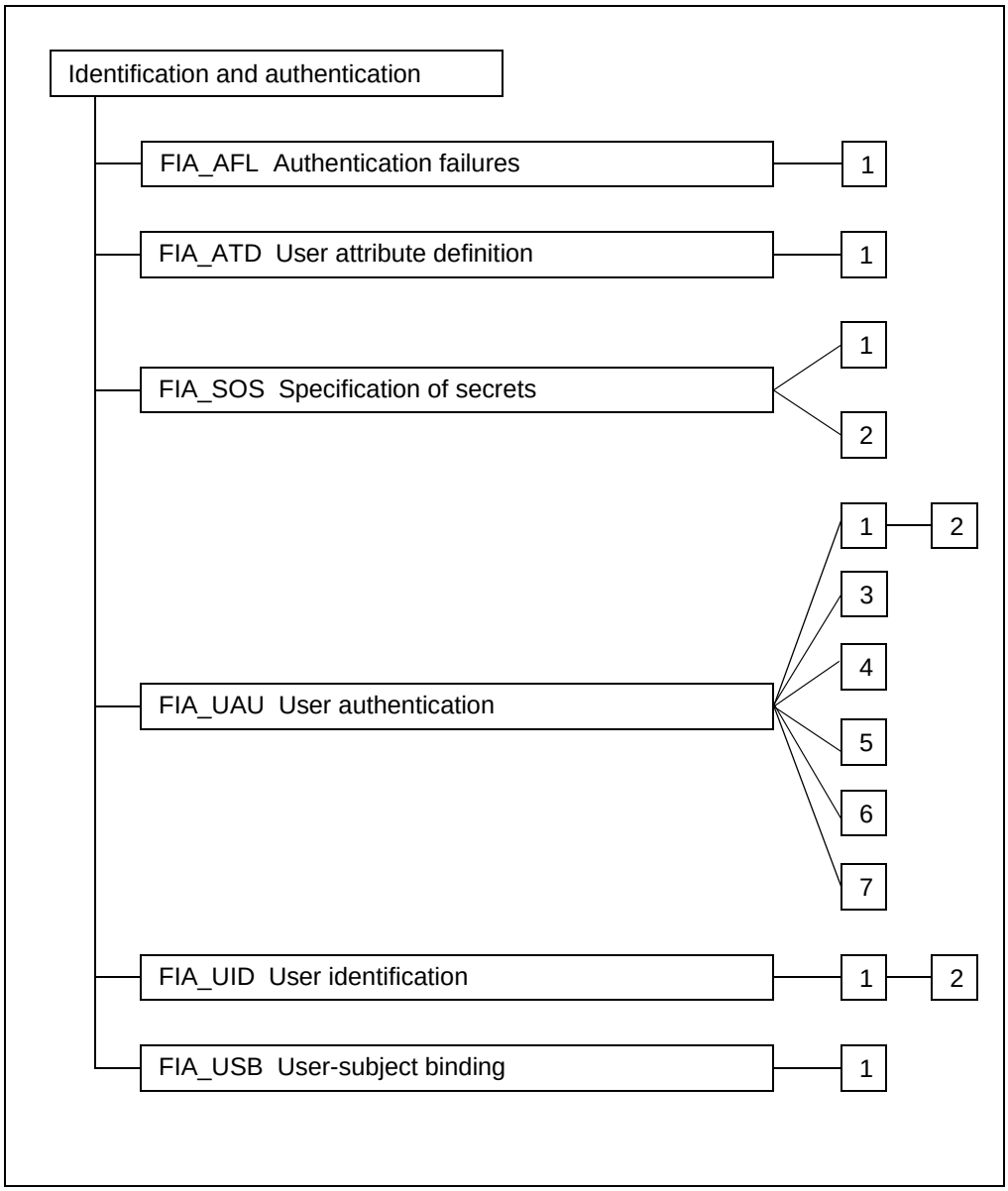

Figure G.1 - Identification and authentication class decomposition

G.1 Authentication failures (FIA_AFL)

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.

FIA_AFL.1 Authentication failure handling

User application notes

The PP/ST author may define the number of unsuccessful authentication attempts or may choose to let the TOE developer or the authorised user to define this number. The unsuccessful authentication attempts need not be consecutive, but rather related to an authentication event. Such an authentication event could be the count from the last successful session establishment at a given terminal.

The PP/ST author could specify a list of actions that the TSF shall take in the case of authentication failure. An authorised administrator could also be allowed to manage the events, if deemed opportune by the PP/ST author. These actions could be, among other things, terminal deactivation, user account deactivation, or administrator alarm. The conditions under which the situation will be restored to normal must be specified on the action.

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/ST author, including rules for re-enabling the user session establishment process, or sending an alarm to the administrator. Examples of these actions are: until a specified time has lapsed, until the authorised administrator re-enables the terminal/account, a time related to failed previous attempts (every time the attempt fails, the disabling time is doubled).

Operations

Assignment:

In FIA_AFL.1.1, if the PP/ST author should specify the default number of unsuccessful authentication attempts that, when met or surpassed, will trigger the events. The PP/ST author may specify that the number is: “an authorised administrator configurable number”.

In FIA_AFL.1.1, the PP/ST author should specify the authentication events. Examples of these authentication events are: the unsuccessful authentication attempts since the last successful authentication for the indicated user identity,

the unsuccessful authentication attempts since the last successful authentication for the current terminal, the number of unsuccessful authentication attempts in the last 10 minutes. At least one authentication event must be specified.

In FIA_AFL.1.2, the PP/ST author should specify the actions to be taken in case the threshold is met or surpassed. These actions could be disabling of an account for 5 minutes, disabling the terminal for an increasing amount of time (2 to the power of the number of unsuccessful attempts in seconds), or disabling of the account until unlocked by the administrator and simultaneously informing the administrator. The actions should specify the measures and if applicable the duration of the measure (or the conditions under which the measure will be ended).

G.2 User attribute definition (FIA_ATD)

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

User notes

There are dependencies on the individual security policy definitions. These individual definitions should contain the listing of attributes that are necessary for policy enforcement.

FIA_ATD.1 User attribute definition

User application notes

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.

Operations

Assignment:

In FIA_ATD.1.1, the PP/ST author should specify the security attributes that are associated to an individual user. An example of such a list is {'clearance', 'group identifier', 'rights'}.

G.3 Specification of secrets (FIA_SOS)

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 (e.g. selected by the user and introduced in the system). In such cases, the FIA_SOS.1 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 component can be used to require the TOE to ensure that the secrets that will adhere to some specified metrics.

User notes

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 should be used instead of this family.

FIA_SOS.1 Verification of secrets

User 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.

Operations

Assignment:

In FIA_SOS.1.1, the PP/ST author should provide a defined quality metric. The quality metric specification can be as simple as a description of the quality checks to be performed, or as formal as a reference to a government published standard that defines the quality metrics that secrets must meet. Examples of quality metrics could include a description of the alphanumeric structure of acceptable secrets and/or the space size that acceptable secrets must meet.

FIA_SOS.2 TSF generation of secrets

This component allows the TSF to generate secrets for specific functions such as authentication by means of passwords.

User application notes

When a pseudo-random number generator is used in a secret generation algorithm, it should accept as input random data that would provide output that has a high degree of unpredictability. This random data (seed) can be derived from a number of available parameters such as a system clock, system registers, date, time, etc. The parameters should be selected to ensure that the number of

unique seeds that can be generated from these inputs should be at least equal to the minimum number of secrets that must be generated.

Operations

Assignment:

In FIA_SOS.2.1, the PP/ST author should provide a defined quality metric. The quality metric specification can be as simple as a description of the quality checks to be performed or as formal as a reference to a government published standard that defines the quality metrics that secrets must meet. Examples of quality metrics could include a description of the alphanumeric structure of acceptable secrets and/or the space size that acceptable secrets must meet.

In FIA_SOS.2.2, the PP/ST author should provide a list of TSF functions for which the TSF generated secrets must be used. An example of such a function could include a password based authentication mechanism.

G.4 User authentication (FIA_UAU)

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

FIA_UAU.1 Timing of authentication

User application notes

This component requires that the PP/ST author define the TSF-mediated actions that can be performed by the TSF on behalf of the user before the claimed identity of the user is authenticated. The TSF-mediated actions should have no security concerns with users incorrectly identifying themselves prior to being authenticated. For all other TSF-mediated actions not in the list, the user must be authenticated before the action can be performed by the TSF on behalf of the user.

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 and FIA_UID.2 with the appropriate assignments.

Operations

Assignment:

In FIA_UAU.1.1, the PP/ST author should specify a list of TSF-mediated actions that can be performed by the TSF on behalf of a user before the claimed identity of the user is authenticated. This list cannot be empty. If no actions are appropriate, component FIA_UAU.2 should be used instead. An example of such an action might include the request for help on the login procedure.

FIA_UAU.2 User authentication before any action

User application notes

This component requires that users are identified before any TSF-mediated action can take place on behalf of that user.

FIA_UAU.3 Unforgeable authentication

User 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 (e.g. biometrics). It is impossible for a TSF to detect or prevent the sharing of passwords outside the control of the TSF.

Operations

Selection:

In FIA_UAU.3.1, the PP/ST author should specify whether the TSF will detect, prevent, or detect and prevent forging of authentication data

In FIA_UAU.3.2, the PP/ST author should specify whether the TSF will detect, prevent, or detect and prevent copying of authentication data

FIA_UAU.4 Single-use authentication mechanisms

User application notes

This component addresses requirements for authentication mechanisms based on single-use authentication data. Single-use authentication data can be something the user has or knows, but not something the user is. Examples of single-use authentication data include single-use passwords, encrypted time-stamps, and/or random numbers from a secret lookup table.

The PP/ST author can specify to which authentication mechanism(s) this requirement applies.

Operations

Assignment:

In FIA_UAU.4.1, the PP/ST author should specify the list of authentication mechanisms to which this requirement applies. This assignment can be 'all authentication mechanisms'. An example of this assignment could be "the authentication mechanism employed to authenticate people on the external network".

FIA_UAU.5 Multiple authentication mechanisms

User application notes

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 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 be on the system, a 'none' authentication mechanism can be incorporated. The use of such access should be clearly explained in the rules of FIA_UAU.5.2.

Operations

Assignment:

In FIA_UAU.5.1, the PP/ST author should define the available authentication mechanisms. An example of such a list could be: "none, password mechanism, biometric (retinal scan), S/key mechanism".

In FIA_UAU.5.2, the PP/ST author should specify the rules that describe how the authentication mechanisms provide authentication and when each is to be used. This means that for each situation the set of mechanisms that might be used for authenticating the user must be described. An example of a list of such rules is: "if the user has special privileges a password mechanism and a biometric mechanism both shall be used, with success only if both succeed; for all other users a password mechanism shall be used."

The PP/ST author might give the boundaries within which the authorised administrator may specify specific rules. An example of a rule is: "the user shall always be authenticated by means of a token; the administrator might specify additional authentication mechanisms that also must be used." The PP/ST author also might choose not to specify any boundaries but leave the authentication mechanisms and their rules completely up to the authorised administrator.

FIA_UAU.6 Re-authenticating

User application notes

This component addresses potential needs to re-authenticate users at defined points in time. These may include user requests for the TSF to perform security relevant actions, as well as requests from non-TSF entities for re-authentication (e.g. a server application requesting that the TSF re-authenticate the client it is serving).

Operations

Assignment:

In FIA_UAU.6.1, the PP/ST author should specify the list of conditions requiring re-authentication. This list could include a specified user inactivity period that has elapsed, the user requesting a change in active security attributes, or the user requesting the TSF to perform some security critical function.

The PP/ST author might give the boundaries within which the reauthentication should occur and leave the specifics to the authorised administrator. An example of such a rule is: "the user shall always be re-authenticated at least once a day; the administrator might specify that the re-authentication should happen more often but not more often than once every 10 minutes."

FIA_UAU.7 Protected authentication feedback

User application notes

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' (e.g. star) for each password character provided, and not the original character.

Operations

Assignment:

In FIA_UAU.7.1, the PP/ST author should specify the feedback related to the authentication process that will be provided to the user. An example of a feedback assignment is "the number of characters typed", another type of feedback is "the authentication mechanism that failed the authentication".

G.5 User identification (FIA_UID)

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

FIA_UID.1 Timing of identification

User application notes

This component poses requirements for the user to be identified. The PP/ST author can indicate specific actions that can be performed before the identification takes place.

If FIA_UID.1 is used, the TSF-mediated actions mentioned in FIA_UID.1 should also appear in this FIA_UAU.1.

Operations

Assignment:

In FIA_UID.1.1, the PP/ST author should specify a list of TSF-mediated actions that can be performed by the TSF on behalf of a user before the user has to identify itself. If no actions are appropriate, component FIA_UID.2 should be used instead. An example of such an action might include the request for help on the login procedure.

FIA_UID.2 User identification before any action

User application notes

In this component users will be identified. A user is not allowed by the TSF to perform any action before being identified.

G.6 User-subject binding (FIA_USB)

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

FIA_USB.1 User-subject binding

User application notes

The phrase "acting on behalf of" has proven to be a contentious issue in previous criteria. It is intended that a subject is acting on behalf of the user who caused the subject to come into being or to be activated to perform a certain task. Therefore, when a subject is created, that subject is acting on behalf of the user who initiated the creation. In case anonymity is used, the subject is still acting on behalf of a user, but the identity of the user is unknown. A special category are the subjects that serve multiple users (e.g. a server process). In such cases the user that created this subject is assumed to be the 'owner'.

Annex H

Security management (FMT)

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 that form a distributed system, the timing issues with respect to propagation of security attributes, TSF data, and function modification become very complex, especially if the information is required to be replicated across the parts of the TOE. This should be considered when selecting components such as FMT_REV.1 Revocation, or FMT_SAE.1 Time-limited authorisation, where the behaviour might be impaired. In such situations, use of components from FPT_TRC is advisable.

.

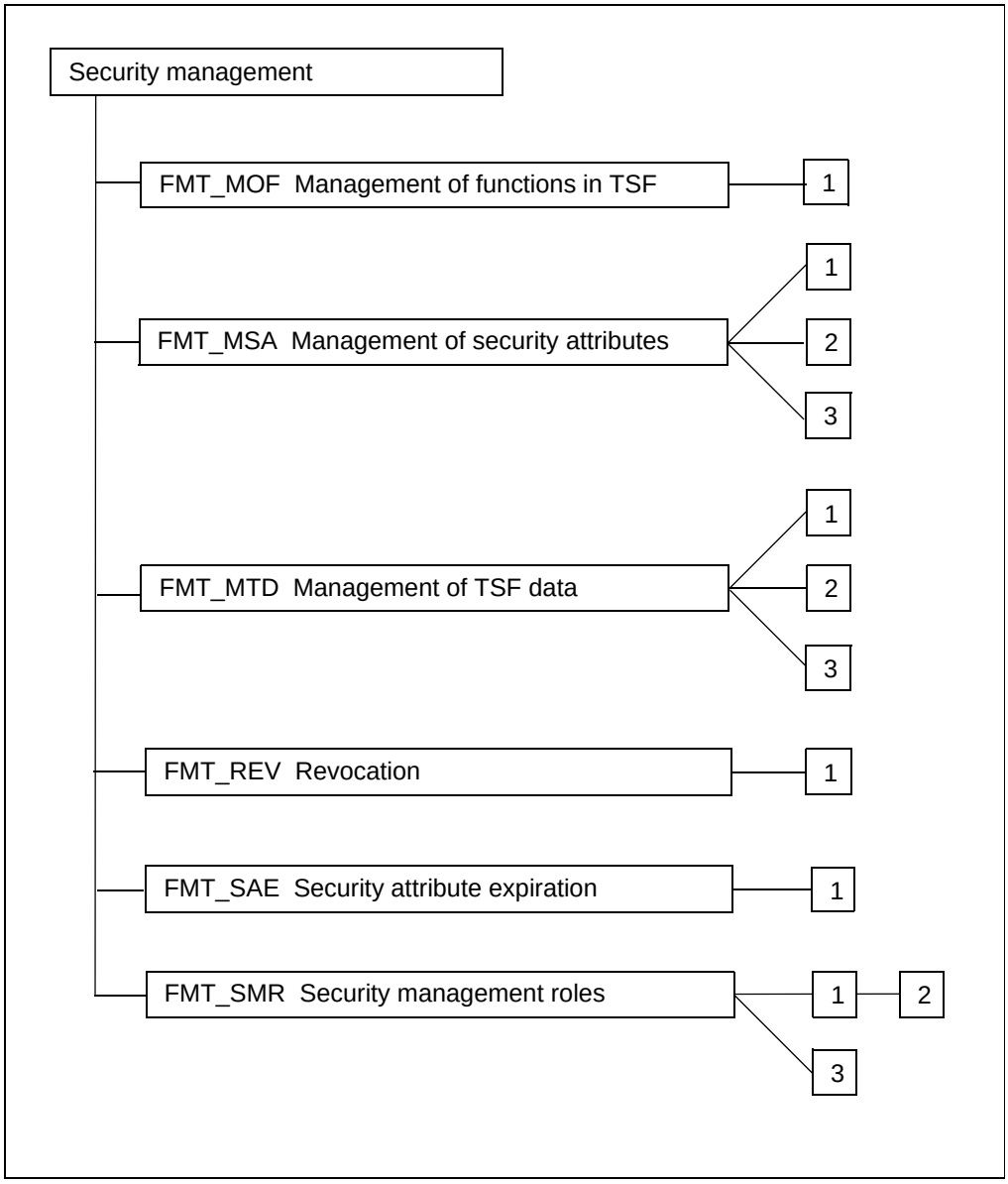

Figure H.1 - Security management class decomposition

H.1 Management of functions in TSF (FMT_MOF)

The TSF management functions enable authorised users to set up and control the secure operation of the TOE. These administrative functions typically fall into a number of different categories:

a) Management functions that relate to access control, accountability and authentication controls enforced by the TOE. For example, definition and update of user security characteristics (e.g. unique identifiers associated with user names, user accounts, system entry parameters) or definition and update of auditing system controls (e.g. selection of audit events, management of audit trails, audit trail analysis, and audit report generation), definition and update of per-user policy attributes (such as user clearance), definition of known system access control labels, and control and management of user groups.

b) Management functions that relate to controls over availability. For example, definition and update of availability parameters or resource quotas.

c) Management functions that relate to general installation and configuration. For example, TOE configuration, manual recovery, installation of TOE security fixes (if any), repair and reinstallation of hardware.

d) Management functions that relate to routine control and maintenance of TOE resources. For example, enabling and disabling peripheral devices, mounting of removable storage media, backup and recovery of user and system objects.

Note that these functions need to be present in a TOE based on the families included in the PP or ST. It is the responsibility of the PP/ST author to ensure that adequate functions will be provided to manage the system in a secure fashion.

The TSF might contain functions that can be controlled by an administrator. For example, the auditing functions could be switched off, the time synchronisation could be switchable, and/or the authentication mechanism could be modifiable.

FMT_MOF.1 Management of security functions behaviour

This component allows identified roles to manage the security functions of the TSF. This might entail obtaining the current status of a security function, disabling or enabling the security function, or modifying the behaviour of the security function. An example of modifying the behaviour of the security functions is changing of authentication mechanisms.

Operations

Selection:

In FMT_MOF.1.1 the PP/ST author should select whether the role can determine the behaviour of, disable, enable, and/or modify the behaviour of the security functions.

Assignment:

In FMT_MOF.1.1 the PP/ST author should specify the functions that can be modified by the identified roles. Examples include auditing and time determination.

In FMT_MOF.1.1 the PP/ST author should specify the roles that are allowed to modify the functions in the TSF. The possible roles are specified in FMT_SMR.1.

H.2 Management of security attributes (FMT_MSA)

This family defines the requirements on the management of security attributes.

Users, subjects and objects have associated security attributes that will affect the behaviour of the TSF. Examples of such security attributes are the groups to which a user belongs, the roles he/she might assume, the priority of a process (subject), and the rights belonging to a role or a user. These security attributes might need to be managed by the user, a subject or a specific authorised user (a user with explicitly given rights for this management).

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.

FMT_MSA.2 can be used to ensure that any accepted combination of security attributes is within a secure state. The definition of what “secure” means is left to the TOE guidance and the TSP model. If the developer provided a clear definition of the secure values and the reason why they should be considered secure, the dependency from FMT_MSA.2 to ADV_SPM.1 can be argued away.

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 can be used to specify that these default values can be managed.

FMT_MSA.1 Management of security attributes

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.

The default value of a parameter is the value the parameter takes when it is instantiated without specifically assigned values. An initial value is provided during the instantiation (creation) of a parameter, and overrides the default value.

Operations

Assignment:

In FMT_MSA.1.1, the PP/ST author should list the access control SFP or the information flow control SFP for which the security attributes are applicable.

Selection:

In FMT_MSA.1.1 the PP/ST author should specify the operations that can be applied to the identified security attributes. The PP/ST author can specify that the role can modify the default value (change_default), query, modify the security attribute, delete the security attributes entirely or define their own operation.

Assignment:

In FMT_MSA.1.1, if selected, the PP/ST author should specify which other operations the role could perform. An example of such an operation could be 'create'.

In FMT_MSA.1.1 the PP/ST author should specify the security attributes that can be operated on by the identified roles. It is possible for the PP/ST author to specify that the default value such as default access-rights can be managed. Examples of these security attributes are user-clearance, priority of service level, access control list, default access rights.

In FMT_MSA.1.1 the PP/ST author should specify the roles that are allowed to operate on the security attributes. The possible roles are specified in FMT_SMR.1.

FMT_MSA.2 Secure security attributes

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 (specifically ADV_SPM.1 Informal TOE security policy model) and the resulting information in the guidance. An example could be that if a user account is created, it should have a non-trivial password.

FMT_MSA.3 Static attribute initialisation

User application notes

This component requires that the TSF provide default values for relevant object security attributes, which can be overridden by an initial value. It may still be possible for a new object to have different security attributes at creation, if a mechanism exists to specify the permissions at time of creation.

Operations

Assignment:

In FMT_MSA.3.1,the PP/ST author should list the access control SFP or the information flow control SFP for which the security attributes are applicable.

Selection:

In FMT_MSA.3.1, the PP/ST author should select whether the default property of the access control attribute will be restrictive, permissive, or another property. In case of another property, the PP/ST author should refine this to a specific property.

Assignment:

In FMT_MSA.3.2 the PP/ST author should specify the roles that are allowed to modify the values of the security attributes. The possible roles are specified in FMT_SMR.1.

H.3 Management of TSF data (FMT_MTD)

This component imposes requirements on the management of TSF data. Examples of TSF data are the current time and the audit trail. So, for example, this family allows the specification of whom can read, delete or create the audit trail.

FMT_MTD.1 Management of TSF data

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.

The default value of a parameter is the values the parameter takes when it is instantiated without specifically assigned values. An initial value is provided during the instantiation (creation) of a parameter and overrides the default value.

Operations

Selection:

In FMT_MTD.1.1 the PP/ST author should specify the operations that can be applied to the identified TSF data. The PP/ST author can specify that the role can modify the default value (change_default), clear, query or modify the TSF data, or delete the TSF data entirely. If so desired the PP/ST author could specify any type of operation. To clarify "clear TSF data" means that the content of the TSF data is removed, but that the entity itself remains in the system.

Assignment:

In FMT_MTD.1.1, if selected, the PP/ST author should specify which other operations the role could perform. An example could be 'create'.

In FMT_MTD.1.1 the PP/ST author should specify the TSF data that can be operated on by the identified roles. It is possible for the PP/ST author to specify that the default value can be managed.

In FMT_MTD.1.1 the PP/ST author should specify the roles that are allowed to operate on the TSF data. The possible roles are specified in FMT_SMR.1.

FMT_MTD.2 Management of limits on TSF data

This component specifies limits on TSF data, and actions to be taken if these limits are exceeded. This component, for example, will allow limits on the size of the audit trail to be defined, and specification of the actions to be taken when these limits are exceeded.

Operations

Assignment:

In FMT_MTD.2.1 the PP/ST author should specify the TSF data that can have limits, and the value of those limits. An example of such TSF data is the number of users logged-in.

In FMT_MTD.2.1 the PP/ST author should specify the roles that are allowed to modify the limits on the TSF data and the actions to be taken. The possible roles are specified in FMT_SMR.1.

In FMT_MTD.2.2 the PP/ST author should specify the actions to be taken if the specified limit on the specified TSF data is exceeded. An example of such TSF action is that the authorised user is informed and an audit record is generated.

FMT_MTD.3 Secure TSF data

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 (specifically ADV_SPM.1 Informal TOE security policy model) and the resulting information in the guidance. If the developer provided a clear definition of the secure values and the reason why they should be considered secure, the dependency from FMT_MSA.2 to ADV_SPM.1 can be argued away.

H.4 Revocation (FMT_REV)

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

FMT_REV.1 Revocation

This component specifies requirements on the revocation of rights. It requires the specification of the revocation rules. Examples are:

a) Revocation will take place on the next login of the user;

b) Revocation will take place on the next attempt to open the file;

c) Revocation will take place within a fixed time. This might mean that all open connections are re-evaluated every x minutes.

Operations

Selection:

In FMT_REV.1.1, the PP/ST author should specify whether the ability to revoke security attributes from users, subjects, objects, or any other resources shall be provided by the TSF. If the last option is chosen, then the PP/ST author should use the refinement operation to define the resources.

Assignment:

In FMT_REV.1.1 the PP/ST author should specify the roles that are allowed to modify the functions in the TSF. The possible roles are specified in FMT_SMR.1.

In FMT_REV.1.2, the PP/ST author should specify the revocation rules. Examples of these rules could include: “prior to the next operation on the associated resource”, or “for all new subject creations”.

H.5 Security attribute expiration (FMT_SAE)

This family addresses the capability to enforce time limits for the validity of security attributes. This family can be applied to specify expiration requirements for access control attributes, identification and authentication attributes, certificates (key certificates such as ANSI X509 for example), audit attributes, etc.

FMT_SAE.1 Time-limited authorisation

Operations

Assignment:

For FMT_SAE.1.1, the PP/ST author should provide the list of security attributes for which expiration is to be supported. An example of such an attribute might be a user's security clearance.

In FMT_SAE.1.1 the PP/ST author should specify the roles that are allowed to modify the security attributes in the TSF. The possible roles are specified in FMT_SMR.1.

For FMT_SAE.1.2, the PP/ST author should provide a list of actions to be taken for each security attribute when it expires. An example might be that the user's security clearance, when it expires, is set to the lowest allowable clearance on the TOE. If immediate revocation is desired by the PP/ST, the action "immediate revocation" should be specified.

H.6 Security management roles (FMT_SMR)

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 authorised to use a particular security-relevant administrative function.

Some management actions can be performed by users, others only by designated people within the organisation. This family allows the definition of different roles, such as owner, auditor, administrator, daily-management.

The roles as used in this family are security related roles. Each role can encompass an extensive set of capabilities (e.g. root in UNIX), or can be a single right (e.g. right to read a single object such as the helpfile). This family defines the roles. The capabilities of the role are defined in FMT_MOF, FMT_MSA and FMT_MTD.

Some type of roles might be mutually exclusive. For example the daily-management might be able to define and activate users, but might not be able to remove users (which is reserved for the administrator (role)). This class will allow policies such as two-person control to be specified.

FMT_SMR.1 Security roles

This component specifies the different roles that the TSF should recognise. Often the system distinguishes between the owner of an entity, an administrator and other users.

Operations

Assignment:

In FMT_SMR.1.1 the PP/ST author should specify the roles that are recognised by the system. These are the roles that users could occupy with respect to security. Examples are: owner, auditor and administrator.

FMT_SMR.2 Restrictions on security roles

This component specifies the different roles that the TSF should recognise, and conditions on how those roles could be managed. Often the system distinguishes between the owner of an entity, an administrator and other users.

The conditions on those roles specify the interrelationship between the different roles, as well as restrictions on when the role can be assumed by a user.

Operations

Assignment:

In FMT_SMR.2.1 the PP/ST author should specify the roles that are recognised by the system. These are the roles that users could occupy with respect to security. Examples are: owner, auditor, administrator.

In FMT_SMR.2.3 the PP/ST author should specify the conditions that govern role assignment. Examples of these conditions are: “an account cannot have both the auditor and administrator role” or “a user with the assistant role must also have the owner role”.

FMT_SMR.3 Assuming roles

This component specifies that an explicit request must be given to assume the specific role.

Operations

Assignment:

In FMT_SMR.3.1 the PP/ST author should specify the roles that require an explicit request to be assumed. Examples are: auditor and administrator.

Annex I

Privacy (FPR)

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 authorised users are covered by the required security functions. For example, a PP/ST author might consider it appropriate not to require protection of the privacy of users against a suitably authorised user.

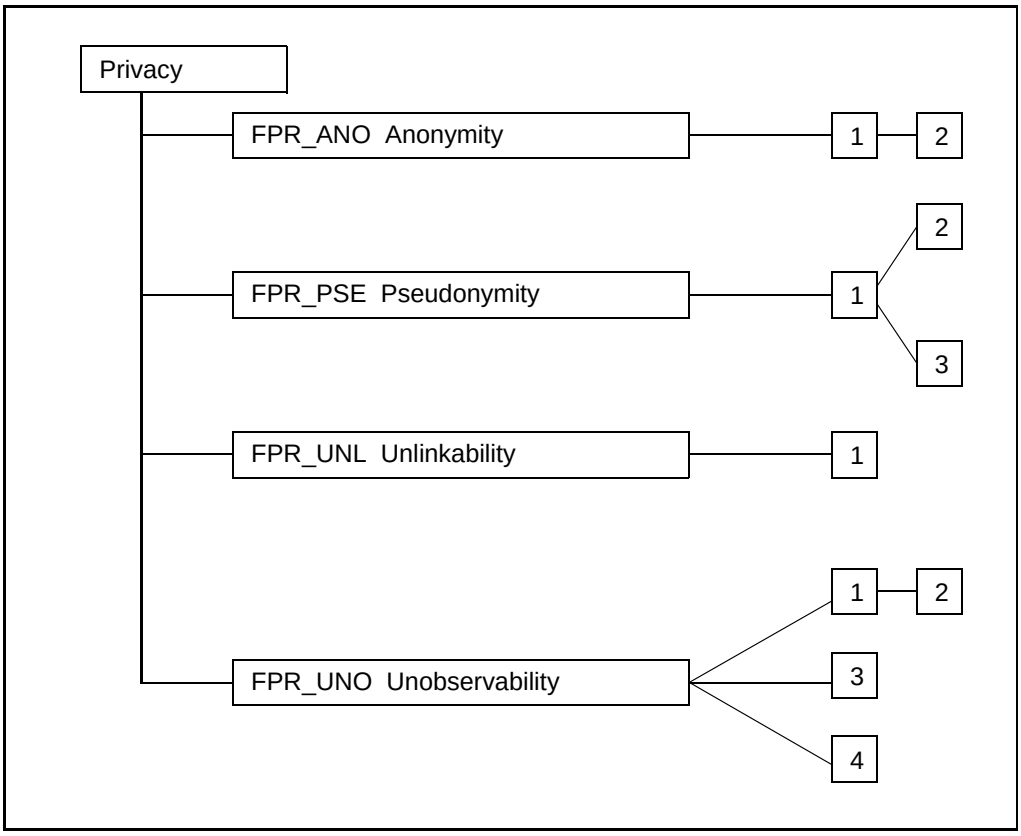

Figure I.1 - Privacy class decomposition

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 or FAU. For example, if authorised users are not allowed to see the user identity (e.g. Anonymity or Pseudonymity), it will obviously not be possible to hold individual users accountable for any security relevant actions they perform that are covered by the privacy requirements. However, it may still be possible to include audit requirements in a PP/ST, where

the fact that a particular security relevant event has occurred is more important than knowing who was responsible for it.

Additional information is provided in the application notes for class FAU, 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 (e.g. if there is a requirement for auditing). In addition, some types of privacy threats are best countered by a combination of components from several families.

All families assume that a user does not explicitly perform an action that discloses the user’s own identity. For example, the TSF is not expected to screen the user name in electronic messages or databases.

All families in this class have components that can be scoped through operations. These operations allow the PP/ST author to state the cooperating users/subjects to which the TSF must be resistant. An example of an instantiation of anonymity could be: “The TSF shall ensure that the users and/ or subjects are unable to determine the user identity bound to the teleconsulting application”.

It is noted that the TSF should not only provide this protection against individual users, but also against users cooperating to obtain the information. The strength of the protection provided by this class should be described as strength of function as specified in part 1 Annex B and part 1 Annex C.

I.1 Anonymity (FPR_ANO)

Anonymity ensures that a subject may use a resource or service without disclosing its user identity.

User notes

The intention of this family is to specify that a user or subject might take action without releasing its user identity to others such as users, subjects, or objects. The family provides the PP/ST author with a means to identify the set of users that cannot see the identity of someone performing certain actions.

Therefore if a subject, using anonymity, performs an action, another subject will not be able to determine either the identity or even a reference to the identity of the user employing the subject. The focus of the anonymity is on the protection of the users identity, not on the protection of the subject identity; hence, the identity of the subject is not protected from disclosure.

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 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 PP/ST author might want to use a Strength of Function to indicate how much rigour should be applied.

The component levelling distinguishes between the users and an authorised user. An authorised user is often excluded from the component, and therefore allowed to retrieve a user’s identity. However, there is no specific requirement that an authorised user must be able to have the capability to determine the user’s identity. For ultimate privacy the components would be used to say that no user or authorised user can see the identity of anyone performing any action.

Although some systems will provide anonymity for all services that are provided, other systems provide anonymity for certain subjects/operations. To provide this flexibility, an operation is included where the scope of the requirement is defined. If the PP/ST author wants to address all subjects/operations, the words “all subjects and all operations” could be provided.

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.

Examples of potential hostile users or subjects are providers, system operators, communication partners and users, who smuggle malicious parts (e.g. Trojan Horses) into systems. All of these users can investigate usage patterns (e.g. which users used which services) and misuse this information.

FPR_ANO.1 Anonymity

User application notes

This component ensures that the identity of a user is protected from disclosure. There may be instances, however, that a given authorised user can determine who performed certain actions. This component gives the flexibility to capture either a limited or total privacy policy.

Operations

Assignment:

In FPR_ANO.1.1 the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_ANO.1.1 the PP/ST author should identify the list of subjects and/or operations and/or objects where the real user name of the subject should be protected, for example, "the voting application".

FPR_ANO.2 Anonymity without soliciting information

User application notes

This component is used to ensure that the TSF is not allowed to know the identity of the user.

Operations

Assignment:

In FPR_ANO.2.1 the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_ANO.2.1 the PP/ST author should identify the list of subjects and/or operations and/or objects where the real user name of the subject should be protected, for example, "the voting application".

In FPR_ANO.2.2 the PP/ST author should identify the list of services which are subject to the anonymity requirement, for example, "the accessing of job descriptions".

For FPR_ANO.2.2 the PP/ST author should identify the list of subjects from which the real user name of the subject should be protected when the specified services are provided.

I.2 Pseudonymity (FPR_PSE)

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.

User notes

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 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 and FPR_PSE.3.

A way to use the reference is by being able to obtain the original user identifier. For example, in a digital cash environment it would be advantageous to be able to trace the user's identity when a check has been issued multiple times (i.e. fraud). In general, the user's identity needs to be retrieved under specific conditions. The PP/ST author might want to incorporate FPR_PSE.2 Reversible pseudonymity to describe those services.

Another usage of the reference is as an alias for a user. For example, a user who does not wish to be identified, can provide an account to which the resource utilisation should be charged. In such cases, the reference to the user identity is an alias for the user, where other users or subjects can use the alias for performing their functions without ever obtaining the user's identity (for example, statistical operations on use of the system). In this case, the PP/ST author might wish to incorporate FPR_PSE.3 Alias pseudonymity to specify the rules to which the reference must conform.

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 realised that the more stringent components potentially cannot be combined with other requirements, such as identification and authentication or audit. The interpretation of "determine the identity" should be taken in the broadest sense of the word. The information is not provided by the TSF during the operation, nor can the entity determine the subject or the owner of the subject that invoked the operation, nor will the TSF record information, available to the users or subjects, which might release the user identity in the future.

The intent is that the TSF not reveal any information that would compromise the identity of the user, e.g. the identity of subjects acting on the user's behalf. The information that is considered to

be sensitive depends on the effort an attacker is capable of spending. Therefore, the FPR_PSE Pseudonymity family is subject to Strength of Function requirements.

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.

Examples of potential hostile users are providers, system operators, communication partners and users, who smuggle malicious parts (e.g. Trojan Horses) into systems. All of these attackers can investigate which users used which services and misuse this information. Additionally to Anonymity services, Pseudonymity Services contains methods for authorisation without identification, especially for anonymous payment ("Digital Cash"). This helps providers to obtain their payment in a secure way while maintaining customer anonymity.

FPR_PSE.1 Pseudonymity

User application notes

This component provides the user protection against disclosure of identity to other users. The user will remain accountable for its actions.

Operations

Assignment:

In FPR_PSE.1.1 the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_PSE.1.1 the PP/ST author should identify the list of subjects and/or operations and/or objects where the real user name of the subject should be protected, for example, 'the accessing of job offers'. Note that 'objects' includes any other attributes that might enable another user or subject to derive the actual identity of the user.

In FPR_PSE.1.2 the PP/ST author should identify the (one or more) number of aliases the TSF is able to provide.

In FPR_PSE.1.2 the PP/ST author should identify the list of subjects to whom the TSF is able to provide an alias.

Selection:

In FPR_PSE.1.3 the PP/ST author should specify whether the user alias is generated by the TSF, or supplied by the user.

Assignment:

In FPR_PSE.1.3 the PP/ST author should identify the metric to which the TSF-generated or user-generated alias should conform.

FPR_PSE.2 Reversible pseudonymity

User application notes

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 the TSF shall provide an alias instead of the user identity. When the specified conditions are satisfied, the user identity to which the alias belong can be determined. An example of such a condition in an electronic cash environment is: “The TSF shall provide the notary a capability to determine the user identity based on the provided alias only under the conditions that a check has been issued twice.”.

Operations

Assignment:

In FPR_PSE.2.1 the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/ or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_PSE.2.1 the PP/ST author should identify the list of subjects and/or operations and/or objects where the real user name of the subject should be protected, for example, ‘the accessing of job offers’. Note that ‘objects’ includes any other attributes that might enable another user or subject to derive the actual identity of the user.

In FPR_PSE.2.2 the PP/ST author should identify the (one or more) number of aliases the TSF, is able to provide.

In FPR_PSE.2.2 the PP/ST author should identify the list of subjects to whom the TSF is able to provide an alias.

Selection:

In FPR_PSE.2.3 the PP/ST author should specify whether the user alias is generated by the TSF or supplied by the user.

Assignment:

In FPR_PSE.2.3 the PP/ST author should identify the metric to which the TSF-generated or user-generated alias should conform.

Selection:

In FPR_PSE.2.4 the PP/ST author should select whether the authorised user and/ or trusted subjects can determine the real user name.

Assignment:

In FPR_PSE.2.4 the PP/ST author should identify the list of trusted subjects that can obtain the real user name under a specified condition, for example, a notary or special authorised user.

In FPR_PSE.2.4 the PP/ST author should identify the list of conditions under which the trusted subjects and authorised user can determine the real user name based on the provided reference. These conditions can be conditions such as time of day, or they can be administrative such as on a court order.

FPR_PSE.3 Alias pseudonymity

User 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.

Operations

Assignment:

In FPR_PSE.3.1 the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/ or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_PSE.3.1 the PP/ST author should identify the list of subjects and/or operations and/or objects where the real user name of the subject should be protected, for example, ‘the accessing of job offers’. Note that ‘objects’ includes any other attributes which might enable another user or subject to derive the actual identity of the user.

In FPR_PSE.3.2 the PP/ST author should identify the (one or more) number of aliases the TSF is able to provide.

In FPR_PSE.3.2 the PP/ST author should identify the list of subjects to whom the TSF is able to provide an alias.

Selection:

In FPR_PSE.3.3 the PP/ST author should specify whether the user alias is generated by the TSF, or supplied by the user.

Assignment:

In FPR_PSE.3.3 the PP/ST author should identify the metric to which the TSF-generated or user-generated alias should conform.

In FPR_PSE.3.4 the PP/ST author should identify the list of conditions that indicate when the used reference for the real user name shall be identical and when it shall be different, for example, "when the user logs on to the same host" it will use a unique alias.

I.3 Unlinkability (FPR_UNL)

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.

User notes

The requirements for unlinkability are intended to protect the user identity against the use of profiling of the operations. For example, when a telephone smart card is employed with a unique number, the telephone company can determine the behaviour of the user of this telephone card. When a telephone profile of the users is known, the card can be linked to a specific user. Hiding the relationship between different invocations of a service or access of a resource will prevent this kind of information gathering.

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. For example, the user associated with the operation, or the terminal which initiated the action, or the time the action was executed. The PP/ST author can specify what kind of relationships are present that must be countered.

Possible applications include the ability to make multiple use of a pseudonym without creating a usage pattern that might disclose the user's identity.

Examples for potential hostile subjects and users are providers, system operators, communication partners and users, who smuggle malicious parts, (e.g. Trojan Horses) into systems, they do not operate but want to get information about. All of these attackers can investigate (e.g. which users used which services) and misuse this information. Unlinkability protects users from linkages, which could be drawn between several actions of a customer. An example is a series of phone calls made by an anonymous customer to different partners, where the combination of the partner's identities might disclose the identity of the customer.

FPR_UNL.1 Unlinkability

User application notes

This component ensures that users cannot link different operations in the system and thereby obtain information.

Operations

Assignment:

In FPR_UNL.1.1 the PP/ST author should specify the set of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect

to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

In FPR_UNL.1.1 the PP/ST author should identify the list of operations which should be subjected to the unlinkability requirement, for example, "sending email".

Selection:

In FPR_UNL.1.1 the PP/ST author should select the relationships that should be obscured. The selection allows either the user identity or an assignment of relations to be specified.

Assignment:

In FPR_UNL.1.1 the PP/ST author should identify the list of relations which should be protected against, for example, "originate from the same terminal".

I.4 Unobservability (FPR_UNO)

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.

User notes

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. Examples of techniques to provide unobservability are:

a) Allocation of information impacting unobservability: Unobservability relevant information (e.g. information that describes that an operation occurred) can be allocated in several locations within the TOE. The information might be allocated to a single randomly chosen part of the TOE such that an attacker does not know which part of the TOE should be attacked. An alternative system might distribute the information such that no single part of the TOE has sufficient information that, if circumvented, the privacy of the user would be compromised. This technique is explicitly addressed in FPR_UNO.2.

b) Broadcast: When information is broadcast (e.g. ethernet, radio), users cannot determine who actually received and used that information. This technique is especially useful when information should reach receivers which have to fear a stigma for being interested in that information (e.g. sensitive medical information).

c) Cryptographic protection and message padding: People observing a message stream might obtain information from the fact that a message is transferred and from attributes on that message. By traffic padding, message padding and encrypting the message stream, the transmission of a message and its attributes can be protected.

Sometimes, users should not see the use of a resource, but an authorised user must be allowed to see the use of the resource in order to perform his duties. In such cases, the FPR_UNO.4 could be used, which provides the capability for one or more authorised 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. In the case of logical separation FPT_SEP may be relevant.

Unobservability of communications may be an important factor in many areas, such as the enforcement of constitutional rights, organisational policies, or in defence related applications.

FPR_UNO.1 Unobservability

User application notes

This component requires that the use of a function or resource cannot be observed by unauthorised users. In addition to this component, a PP/ST author might want to incorporate Covert Channel Analysis.

Operations

Assignment:

In FPR_UNO.1.1 the PP/ST author should specify the list of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

For FPR_UNO.1.1 the PP/ST author should identify the list of operations that are subjected to the unobservability requirement. Other users/subjects will then not be able to observe the operations on a covered object in the specified list (e.g. reading and writing to the object).

For FPR_UNO.1.1 the PP/ST author should identify the list of objects which are covered by the unobservability requirement. An example could be a specific mail server or ftp site.

In FPR_UNO.1.1 the PP/ST author should specify the set of protected users and/ or subjects whose unobservability information will be protected. An example could be: "users accessing the system through the internet".

FPR_UNO.2 Allocation of information impacting unobservability

User application notes

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.

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).

In addition to this component, a PP/ST author might want to incorporate Covert Channel Analysis.

Operations

Assignment:

In FPR_UNO.2.1 the PP/ST author should specify the list of users and/or subjects against which the TSF must provide protection. For example, even if the PP/ST author

specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject, but must protect with respect to cooperating users and/ or subjects. A set of users, for example, could be a group of users which can operate under the same role or can all use the same process(es).

For FPR_UNO.2.1 the PP/ST author should identify the list of operations that are subjected to the unobservability requirement. Other users/subjects will then not be able to observe the operations on a covered object in the specified list (e.g. reading and writing to the object).

For FPR_UNO.2.1 the PP/ST author should identify the list of objects which are covered by the unobservability requirement. An example could be a specific mail server or ftp site.

In FPR_UNO.2.1 the PP/ST author should specify the set of protected users and/or subjects whose unobservability information will be protected. An example could be: “users accessing the system through the internet”.

For FPR_UNO.2.2 the PP/ST author should identify which privacy related information should be distributed in a controlled manner. Examples of this information could be: IP address of subject, IP address of object, time, used encryption keys.

For FPR_UNO.2.2 the PP/ST author should specify the conditions to which the dissemination of the information should adhere. These conditions should be maintained throughout the lifetime of the privacy related information of each instance. Examples of these conditions could be: “the information shall only be present at a single separated part of the TOE and shall not be communicated outside this part of the TOE.”, “the information shall only reside in a single separated part of the TOE, but shall be moved to another part of the TOE periodically”, “the information shall be distributed between the different parts of the TOE such that compromise of any 5 separated parts of the TOE will not compromise the security policy”.

FPR_UNO.3 Unobservability without soliciting information

User application notes

This component is used to require that the TSF does not try to obtain information that might compromise unobservability when provided specific services. Therefore the TSF will not solicit (i.e. try to obtain from other entities) any information that might be used to compromise unobservability.

Operations

Assignment:

In FPR_UNO.3.1 the PP/ST author should identify the list of services which are subject to the unobservability requirement, for example, “the accessing of job descriptions”.

For FPR_UNO.3.1 the PP/ST author should identify the list of subjects from which privacy related information should be protected when the specified services are provided.

In FPR_UNO.3.1 the PP/ST author should specify the privacy related information that will be protected from the specified subjects. Examples include the identity of the subject that used a service and the quantity of a service that has been used such as memory resource utilisation.

FPR_UNO.4 Authorised user observability

User application notes

This component is used to require that there will be one or more authorised users with the rights to view the resource utilisation. Without this component, this review is allowed, but not mandated.

Operations

Assignment:

In FPR_UNO.4.1 the PP/ST author should specify the set of authorised users for which the TSF must provide the capability to observe the resource utilisation. A set of authorised users, for example, could be a group of authorised users which can operate under the same role or can all use the same process(es).

In FPR_UNO.4.1 the PP/ST author should specify the set of resources and/or services that the authorised user must be able to observe.

Annex J

Protection of the TOE Security Functions (FPT)

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

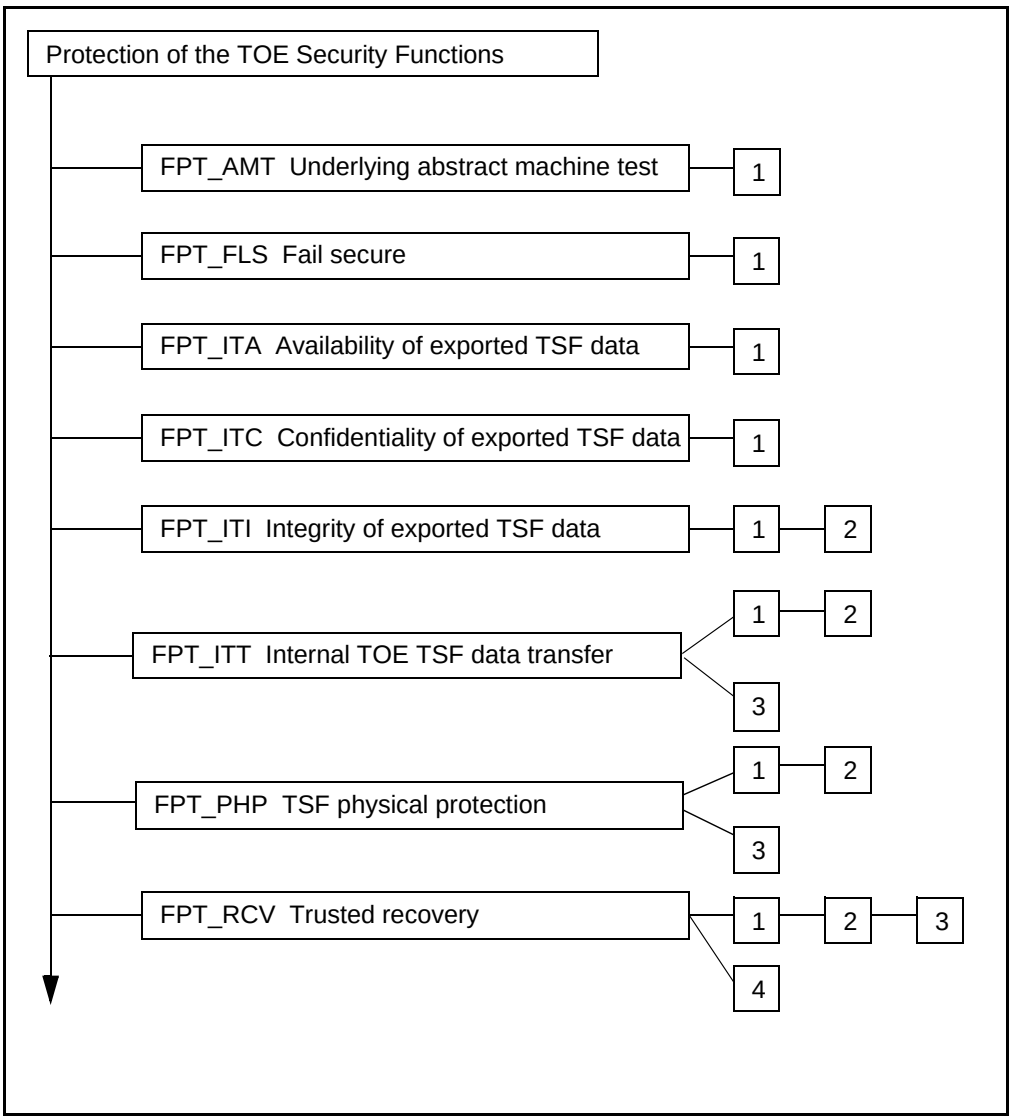

Figure J.1 - Protection of the TOE Security Functions class decomposition

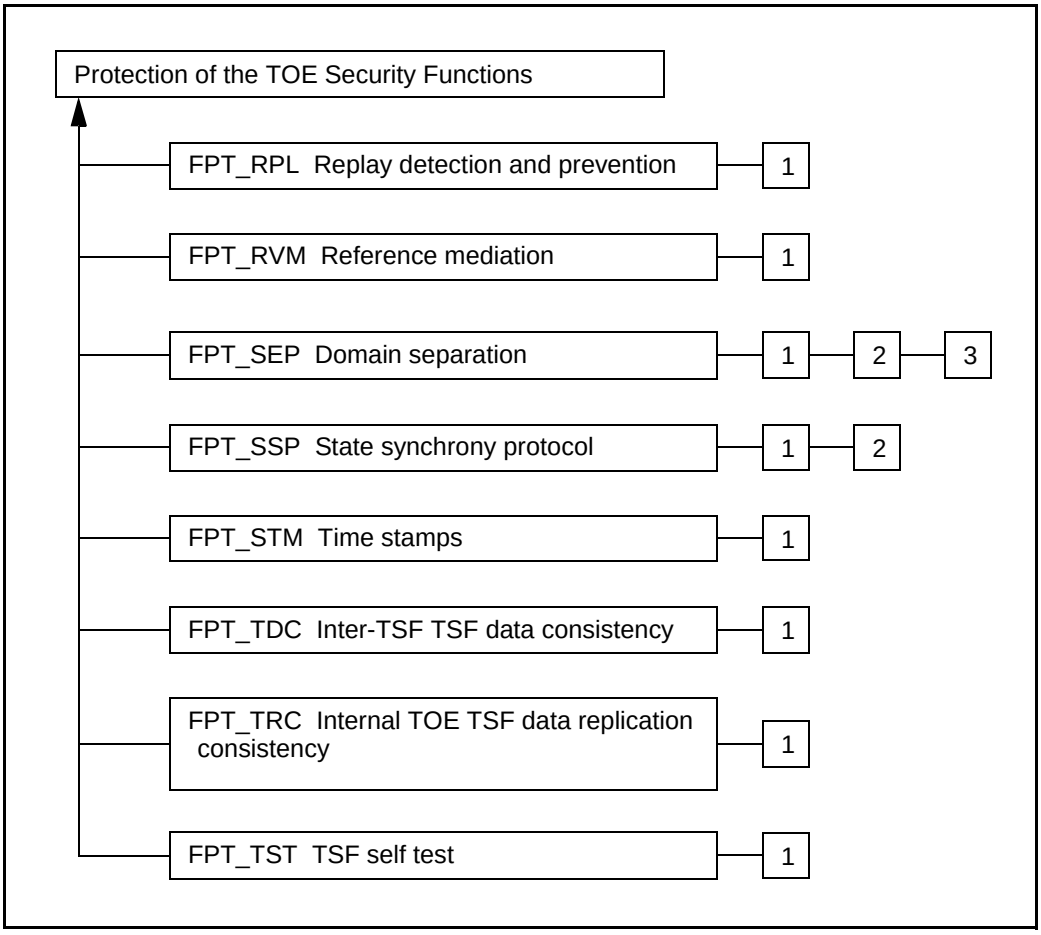

Figure J.2 - Protection of the TOE Security Functions class decomposition (Cont.)

From the point of view of this class, there are three significant portions that make up the TSF:

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

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

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

All of the families in the FPT class can be related to these areas, and fall into the following groupings:

a) FPT_PHP (TSF physical protection), which provides an authorised user with the ability to detect external attacks on the parts of the TOE that comprise the TSF.

b) FPT_AMT (Underlying abstract machine test) and FPT_TST (TSF self test), which provide an authorised user with the ability to verify the correct operation of the underlying abstract machine and the TSF as well as the integrity of the TSF data and executable code.

c) FPT_SEP (Domain separation) and FPT_RVM (Reference mediation), which protect the TSF during execution and ensure that the TSF cannot be bypassed. When appropriate components from these families are combined with the appropriate components from ADV_INT (TSF internals), the TOE can be said to have what has been traditionally called a “Reference Monitor.”

d) FPT_RCV (Trusted recovery), FPT_FLS (Fail secure), and FPT_TRC (Internal TOE TSF data replication consistency), which address the behaviour of the TSF when failure occurs and immediately after.

e) FPT_ITA (Availability of exported TSF data), FPT_ITC (Confidentiality of exported TSF data), FPT_ITI (Integrity of exported TSF data), which address the protection and availability of TSF data between the TSF and a remote trusted IT product.

f) FPT_ITT (Internal TOE TSF data transfer), which addresses protection of TSF data when it is transmitted between physically-separated parts of the TOE.

g) FPT_RPL (Replay detection), which addresses the replay of various types of information and/or operations.

h) FPT_SSP (State synchrony protocol), which addresses the synchronisation of states, based upon TSF data, between different parts of a distributed TSF.

i) FPT_STM (Time stamps), which addresses reliable timing.

j) FPT_TDC (Inter-TSF TSF data consistency), which addresses the consistency of TSF data shared between the TSF and a remote trusted IT product.

J.1 Underlying abstract machine test (FPT_AMT)

This family defines the requirements for the TSF's testing of security assumptions made about the underlying abstract machine upon which the TSF relies. This "abstract" machine could be a hardware/firmware platform, or it could be some known and assessed hardware/software combination acting as a virtual machine. Examples of such testing could be testing hardware page protection, sending sample packets across a network to ensure receipt, and verifying the behaviour of the virtual machine interface. These tests can be carried out either in some maintenance state, at start-up, on-line, or continuously. The actions to be taken by the TOE as the result of testing are defined in FPT_RCV.

User notes

The term "underlying abstract machine" typically refers to the hardware components upon which the TSF has been implemented. However, the phrase can also be used to refer to an underlying, previously evaluated hardware and software combination behaving as a virtual machine upon which the TSF relies.

The tests of the abstract machine may take various forms:

a) **Power-On Tests**. These are tests that ensure the correct operation of the underlying platform. For hardware and firmware, this might include tests of elements such as memory boards, data paths, buses, control logic, processor registers, communication ports, console interfaces, speakers, and peripherals. For software elements (virtual machine), this would include verification of correct initialisation and behaviour.

b) **Loadable Tests**. These are tests that might be loaded and executed by an authorised user or be activated by specific conditions. This might include processor component stress tests (logic units, calculation units, etc.) and control memory.

Evaluator Notes

The tests of the underlying abstract machine should be sufficient to test all of the characteristics of the underlying abstract machine upon which the TSF relies.

FPT_AMT.1 Abstract machine testing

User application notes

This component provides support for the periodic testing of the security assumptions of the underlying abstract machine upon which the TSF's operation depends, by requiring the ability to periodically invoke testing functions.

The PP/ST author may refine the requirement to state whether the function should be available in off-line, on-line or maintenance mode.

Evaluator application notes

It is acceptable for the functions for periodic testing to be available only in an off-line or maintenance mode. Controls should be in place to limit access, during maintenance, to authorised users.

Operations

Selection:

In FPT_AMT.1.1 the PP/ST author should specify when the TSF will execute the abstract machine testing, during initial start-up, periodically during normal operation, at the request of an authorised user, or under other conditions. In the case of the latter option, the PP/ST author should refine what those conditions are. The PP/ST author, through this selection, has the ability to indicate the frequency with which the self tests will be run. If the tests are run often, then the end users should have more confidence that the TOE is operating correctly then if the tests are run less frequently. However, this need for confidence that the TOE is operating correctly must be balanced with the potential impact on the availability of the TOE, as often times, self tests may delay the normal operation of a TOE.

J.2 Fail secure (FPT_FLS)

The requirements of this family ensure that the TOE will not violate its TSP in the event of certain types of failures in the TSF.

FPT_FLS.1 Failure with preservation of secure state

User application notes

The term “secure state” refers to a state in which the TSF data are consistent and the TSF continues correct enforcement of the TSP. The “secure state” is defined in the TSP model. If the developer provided a clear definition of the secure state and the reason why it should be considered secure, the dependency from FPT_FLS.1 to ADV_SPM.1 can be argued away.

Although it is desirable to audit situations in which failure with preservation of secure state occurs, it is not possible in all situations. The PP/ST author should specify those situations in which audit is desired and feasible.

Failures in the TSF may include “hard” failures, which indicate an equipment malfunction and which may require maintenance, service or repair of the TSF. Failures in the TSF may also include recoverable “soft” failures, which may only require initialisation or resetting of the TSF.

Operations

Assignment:

For FPT_FLS.1.1, the PP/ST author should list the types of failures in the TSF for which the TSF should “fail secure,” that is, should preserve a secure state and continue to correctly enforce the TSP.

J.3 Availability of exported TSF data (FPT_ITA)

This family defines the rules for the prevention of loss of availability of TSF data moving between the TSF and a remote trusted IT product. This data could be TSF critical data such as passwords, keys, audit data, or TSF executable code.

User application notes

This family is used in a distributed system context where the TSF is providing TSF data to a remote trusted IT product. The TSF can only take the measures at its site and cannot be held responsible for the TSF at the other trusted IT product.

If there are different availability metrics for different types of TSF data, then this component should be iterated for each unique pairing of metrics and types of TSF data.

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

Operations

Assignment:

For FPT_ITA.1.1, the PP/ST author should specify the types of TSF data that are subject to the availability metric.

For FPT_ITA.1.1, the PP/ST should specify the availability metric for the applicable TSF data.

For FPT_ITA.1.1, the PP/ST author should specify the conditions under which availability must be ensured. For example: there must be a connection between the TOE and the remote trusted IT product

J.4 Confidentiality of exported TSF data (FPT_ITC)

This family defines the rules for the protection from unauthorised disclosure of TSF data moving between the TSF and a remote trusted IT product. Examples of this data are TSF critical data such as passwords, keys, audit data, or TSF executable code.

User application notes

This family is used in a distributed system context where the TSF is providing TSF data to a remote trusted IT product. The TSF can only take the measures at its site and cannot be held responsible for the behaviour of the other trusted IT product.

FPT_ITC.1 Inter-TSF confidentiality during transmission

Evaluator application notes

Confidentiality of TSF Data during transmission is necessary to protect such information from disclosure. Some possible implementations that could provide confidentiality include the use of cryptographic algorithms as well as spread spectrum techniques.

J.5 Integrity of exported TSF data (FPT_ITI)

This family defines the rules for the protection, from unauthorised modification, of TSF data during transmission between the TSF and a remote trusted IT product. Examples of this data are TSF critical data such as passwords, keys, audit data, or TSF executable code.

User notes

This family is used in a distributed system context where the TSF is exchanging TSF data with a remote trusted IT product. Note that a requirement that addresses modification, detection, or recovery at the remote trusted IT product cannot be specified, as the mechanisms that a remote trusted IT product will use to protect its data cannot be determined in advance. For this reason, these requirements are expressed in terms of the "TSF providing a capability" which the remote trusted IT product can use.

FPT_ITI.1 Inter-TSF detection of modification

User application notes

This component should be used in situations where it is sufficient to detect when data have been modified. An example of such a situation is one in which the remote trusted IT product can request the TOE's TSF to retransmit data when modification has been detected, or respond to such types of request.

The desired strength of modification detection is based upon a specified modification metric that is a function of the algorithm used, which may range from a weak checksum and parity mechanisms that may fail to detect multiple bit changes, to more complicated cryptographic checksum approaches.

Operations

Assignment:

For FPT_ITI.1.1, the PP/ST should specify the modification metric that the detection mechanism must satisfy. This modification metric shall specify the desired strength of the modification detection.

For FPT_ITI.1.2, the PP/ST should specify the actions to be taken if a modification of TSF data has been detected. An example of an action is: "ignore the TSF data, and request the originating trusted product to send the TSF data again".

FPT_ITI.2 Inter-TSF detection and correction of modification

User application notes

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 (e.g. only 1 in a 1000 random messages will be accepted), or to mechanisms that are well known in the public literature (e.g. the strength must be conformant to the strength offered by Secure Hash Algorithm).

The approach taken to correct modification might be done through some form of error correcting checksum.

Evaluator Notes

Some possible means of satisfying this requirement involves the use of cryptographic functions or some form of checksum.

Operations

Assignment:

For FPT_ITI.2.1, the PP/ST should specify the modification metric that the detection mechanism must satisfy. This modification metric shall specify the desired strength of the modification detection.

For FPT_ITT.2.2, the PP/ST should specify the actions to be taken if a modification of TSF data has been detected. An example of an action is: “ignore the TSF data, and request the originating trusted product to send the TSF data again”.

For FPT_ITI.2.3, the PP/ST author should define the types of modification from which the TSF should be capable of recovering.

J.6 Internal TOE TSF data transfer (FPT_ITT)

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

User notes

The determination of the degree of separation (i.e., physical or logical) that would make application of this family useful depends on the intended environment of use. In a hostile environment, there may be risks arising from transfers between parts of the TOE separated by only a system bus or an inter-process communications channel. In more benign environments, the transfers may be across more traditional network media.

Evaluator Notes

One practical mechanism available to a TSF to provide this protection is cryptographically-based.

FPT_ITT.1 Basic internal TSF data transfer protection

Operations

Selection:

In FPT_ITT.1.1, the PP/ST author should specify the desired type of protection to be provided from the choices: disclosure, modification.

FPT_ITT.2 TSF data transfer separation

User application notes

One of the ways to achieve separation of TSF data based on SFP-relevant attributes is through the use of separate logical or physical channels.

Operations

Selection:

In FPT_ITT.1.1, the PP/ST author should specify the desired type of protection to be provided from the choices: disclosure, modification.

FPT_ITT.3 TSF data integrity monitoring

Operations

Selection:

In FPT_ITT.3.1, the PP/ST author should specify the desired type of modification that the TSF shall be able to detect. The PP/ST author should select from:

modification of data, substitution of data, re-ordering of data, deletion of data, or any other integrity errors.

Assignment:

In FPT_ITT.3.1, if the PP/ST author chooses the latter selection noted in the preceding paragraph, then the author should also specify what those other integrity errors are that the TSF should be capable of detecting.

In FPT_ITT.3.2, the PP/ST author should specify the action to be taken when an integrity error is identified.

J.7 TSF physical protection (FPT_PHP)

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

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.

Examples of physical tampering scenarios include mechanical attack, radiation, changing the temperature.

User notes

It is acceptable for the functions that are available to an authorised user for detecting physical tampering to be available only in an off-line or maintenance mode. Controls should be in place to limit access during such modes to authorised users. As the TSF may not be “operational” during those modes, it may not be able to provide normal enforcement for authorised user access. The physical implementation of a TOE might consist of several structures: for example an outer shielding, cards, and chips. This set of “elements” as a whole must protect (protect, notify and resist) the TSF from physical tampering. This does not mean that all devices must provide these features, but the complete physical construct as a whole should.

Although there is only minimal auditing associating with these components, this is solely because there is the potential that the detection and alarm mechanisms may be implemented completely in hardware, below the level of interaction with an audit subsystem (for example, a hardware-based detection system based on breaking a circuit and lighting a light emitting diode (LED) if the circuit is broken when a button is pressed by the authorised user). Nevertheless, a PP/ST author may determine that for a particular anticipated threat environment, there is a need to audit physical tampering. If this is the case, the PP/ST author should include appropriate requirements in the list of audit events. Note that inclusion of these requirements may have implications on the hardware design and its interface to the software.

FPT_PHP.1 Passive detection of physical attack

User application notes

FPT_PHP.1 should be used when threats from unauthorised physical tampering with parts of the TOE are not countered by procedural methods. It addresses the threat of undetected physical tampering with the TSF. Typically, an authorised user would be given the function to verify whether tampering took place. As written, this component simply provides a TSF capability to detect tampering. The dependency on FMT_MOF.1 is required to specify who can make use of that capability, and how they can make use of that capability. If this function is realised by non-IT mechanisms (e.g. physical inspection) it could be justified that the dependency on FMT_MOF.1 is not satisfied.

FPT_PHP.2 Notification of physical attack

User application notes

FPT_PHP.2 should be used when threats from unauthorised physical tampering with parts of the TOE are not countered by procedural methods, and it is required that designated individuals be notified of physical tampering. It addresses the threat that physical tampering with TSF elements, although detected, may not be noticed.

Operations

Assignment:

For FPT_PHP.2.3, the PP/ST author should provide a list of TSF devices/ elements for which active detection of physical tampering is required.

For FPT_PHP.2.3, the PP/ST author should designate a user or role that is to be notified when tampering is detected. The type of user or role may vary depending on the particular security administration component (from the FMT_MOF.1 family) included in the PP/ST.

FPT_PHP.3 Resistance to physical attack

For some forms of tampering, it is necessary that the TSF not only detects the tampering, but actually resists it or delays the attacker.

User application notes

This component should be used when TSF devices and TSF elements are expected to operate in an environment where a physical tampering (e.g. observation, analysis, or modification) of the internals of a TSF device or TSF element itself is a threat.

Operations

Assignment:

For FPT_PHP.3.1, the PP/ST author should specify tampering scenarios to a list of TSF devices/elements for which the TSF should resist physical tampering. This list may be applied to a defined subset of the TSF physical devices and elements based on considerations such as technology limitations and relative physical exposure of the device. Such subsetting should be clearly defined and justified. Furthermore, the TSF should automatically respond to physical tampering. The automatic response should be such that the policy of the device is preserved; for example, with a confidentiality policy, it would be acceptable to physically disable the device so that the protected information may not be retrieved.

For FPT_PHP.3.1, the PP/ST author should specify the list of TSF devices/ elements for which the TSF should resist physical tampering in the scenarios that have been identified.

J.8 Trusted recovery (FPT_RCV)

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. Failures that must be generally anticipated include the following:

a) Unmaskable action failures that always result in a system crash (e.g. persistent inconsistency of critical system tables, uncontrolled transfers within the TSF code caused by transient failures of hardware or firmware, power failures, processor failures, communication failures).

b) Media failures causing part or all of the media representing the TSF objects to become inaccessible or corrupt (e.g. parity errors, disk head crash, persistent read/write failure caused by misaligned disk heads, worn-out magnetic coating, dust on the disk surface).

c) Discontinuity of operation caused by erroneous administrative action or lack of timely administrative action (e.g. unexpected shutdowns by turning off power, ignoring the exhaustion of critical resources, inadequate installed configuration).

Note that recovery may be from either a complete or partial failure scenario. Although a complete failure might occur in a monolithic operating system, it is less likely to occur in a distributed environment. In such environments, subsystems may fail, but other portions remain operational. Further, critical components may be redundant (disk mirroring, alternative routes), and checkpoints may be available. Thus, recovery is expressed in terms of recovery to a secure state.

This family identifies a maintenance mode. In this maintenance mode normal operation might be impossible or severely restricted, as otherwise insecure situations might occur. Typically, only authorised users should be allowed access to this mode but the real details of who can access this mode is a function of Class FMT Security management. If FMT does not put any controls on who can access this mode, then it may be acceptable to allow any user to restore the system if the TOE enters such a state. However, in practice, this is probably not desirable as the user restoring the system has an opportunity to configure the TOE in such a way as to violate the TSP.

Mechanisms designed to detect exceptional conditions during operation fall under FPT_TST (TSF self test), FPT_FLS (Fail secure), and other areas that address the concept of “Software Safety.”

User notes

Throughout this family, the phrase “secure state” is used. This refers to some state in which the TOE has consistent TSF data and a TSF that can correctly enforce the policy. This state may be the initial “boot” of a clean system, or it might be some checkpointed state. The “secure state” is defined in the TSP model. If the developer provided a clear definition of the secure state and the reason why it should be considered secure, the dependency from FPT_FLS.1 to ADV_SPM.1 can be argued away.

FPT_RCV.1 Manual recovery

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.

User application notes

This component is intended for use in TOEs that do not require unattended recovery to a secure state. The requirements of this component reduce the threat of protection compromise resulting from an attended TOE returning to an insecure state after recovery from a failure or other discontinuity.

Evaluator application notes

It is acceptable for the functions that are available to an authorised user for trusted recovery to be available only in a maintenance mode. Controls should be in place to limit access during maintenance to authorised users.

FPT_RCV.2 Automated recovery

Automated recovery is considered to be more useful than manual recovery, as it allows the machine to operate in an unattended fashion.

User application notes

The component FPT_RCV.2 extends the feature coverage of FPT_RCV.1 by requiring that there be at least one automated method of recovery from failure or service discontinuity. It addresses the threat of protection compromise resulting from an unattended TOE returning to an insecure state after recovery from a failure or other discontinuity.

Evaluator application notes

It is acceptable for the functions that are available to an authorised user for trusted recovery to be available only in a maintenance mode. Controls should be in place to limit access during maintenance to authorised users.

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.

Operations

Assignment:

For FPT_RCV.2.2, the PP/ST author should specify the list of failures or other discontinuities for which automated recovery must be possible.

FPT_RCV.3 Automated recovery without undue loss

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.

User application notes

The component FPT_RCV.3 extends the feature coverage of FPT_RCV.2 by requiring that there not be undue loss of TSF data or objects within the TSC. At FPT_RCV.2, 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.

This component addresses the threat of protection compromise resulting from an unattended TOE returning to an insecure state after recovery from a failure or other discontinuity with a large loss of TSF data or objects within the TSC.

Evaluator application notes

It is acceptable for the functions that are available to an authorised user for trusted recovery to be available only in a maintenance mode. Controls should be in place to limit access during maintenance to authorised users.

It is assumed that the evaluators will verify the robustness of the automated recovery mechanisms.

Operations

Assignment:

For FPT_RCV.3.2, the PP/ST author should specify the list of failures or other discontinuities for which automated recovery must be possible.

For FPT_RCV.3.3, the PP/ST author should provide a quantification for the amount of loss of TSF data or objects that is acceptable.

FPT_RCV.4 Function recovery

Function recovery requires that if there should be some failure in the TSF, that certain SFs in the TSF should either complete successfully or recover to a secure state.

Operations

Assignment:

In FPT_RCV.4.1, the PP/ST author should specify a list the SFs and failure scenarios. In the event that any of the identified failure scenarios happen, the SFs that have been specified must either complete successfully or recover to a consistent and secure state.

J.9 Replay detection (FPT_RPL)

This family addresses detection of replay for various types of entities and subsequent actions to correct.

FPT_RPL.1 Replay detection

User application notes

The entities included here are, for example, messages, service requests, service responses, or sessions.

Operations

Assignment:

In FPT_RPL.1.1, the PP/ST author should provide a list of identified entities for which detection of replay should be possible. Examples of such entities might include: messages, service requests, service responses, and user sessions.

In FPT_RPL.1.2, the PP/ST author should specify the list of actions to be taken by the TSF when replay is detected. The potential set of actions that can be taken includes: ignoring the replayed entity, requesting confirmation of the entity from the identified source, and terminating the subject from which the re-played entity originated.

J.10 Reference mediation (FPT_RVM)

The components of this family address the “always invoked” aspect of a traditional reference monitor. The goal of these components is to ensure, with respect to the TSC, that all actions requiring policy enforcement invoked by subjects untrusted with respect to any or all of that SFP to objects controlled by that SFP are validated by the TSF against the SFP. If the portion of the TSF that enforces the SFP also meets the requirements of appropriate components from FPT_SEP (Domain separation) and ADV_INT (TSF internals), than that portion of the TSF provides a “reference monitor” for that SFP.

The Reference Monitor is that portion of the TSF responsible for the enforcement of the TSP; it has the following three characteristics:

a) Untrusted subjects cannot interfere with its operation; i.e. it is tamperproof. This is addressed by the components in the FPT_SEP family.

b) Untrusted subjects cannot bypass its checks; i.e. it is always invoked. This is addressed by the components in the FPT_RVM family.

c) It is simple enough to be analysed and its behaviour understood (i.e. its design is conceptually simple.) This is addressed by the components in the ADV_INT family.

This component states that, “the TSF shall ensure that TSP enforcement functions are invoked and succeed before each and every function within the TSC is allowed to proceed.” In any system (distributed or otherwise) there are a finite number of functions responsible for enforcing the TSP. There is nothing in this requirement that mandates or prescribes that a single function is invoked to handle security. Rather, it allows multiple functions to fill the role of reference monitor, and the collection of them responsible for enforcing the TSP are simply called, collectively, the reference monitor. However, this must be balanced by the goal of keeping the “reference monitor” simple.

A TSF that implements a SFP provides effective protection against unauthorised functions if and only if all enforceable actions (e.g. accesses to objects) requested by subjects untrusted with respect to any or all of that SFP are validated by the TSF before succeeding, If the enforceable action is incorrectly enforced or bypassed, the overall enforcement of the SFP has been compromised. “Untrusted” subjects could then bypass the SFP in a variety of unauthorised ways (e.g. circumvent access checks for some subjects or objects, bypass checks for objects whose protection was assumed by applications, retain access rights beyond their intended lifetime, bypass auditing of audited actions, or bypass authentication). Note that the term “untrusted subjects” refers to subjects untrusted with respect to any or all of the specific SFPs being enforced; a subject may be trusted with respect to one SFP and untrusted with respect to a different SFP.

FPT_RVM.1 Non-bypassability of the TSP

User application notes

In order to obtain the equivalent of a reference monitor, this component must be used with either FPT_SEP.2 (SFP domain separation) or FPT_SEP.3 (Complete reference monitor), and ADV_INT.3 (Minimisation of complexity). Further, if complete reference mediation is required, the components from Class FDP User data protection must cover all objects.

J.11 Domain separation (FPT_SEP)

The components of this family ensure that at least one security domain is available for the TSF's own execution, and that the TSF is protected from external interference and tampering (e.g. by modification of TSF code or data structures) by untrusted subjects. Satisfying the requirements of this family makes the TSF self-protecting, meaning that an untrusted subject cannot modify or damage the TSF.

This family requires the following:

a) The resources of the TSF's security domain ("protected domain") and those of subjects and unconstrained entities external to the domain are separated such that the entities external to the protected domain cannot observe or modify data structures or code internal to the protected domain.

b) The transfer of subjects between domains are controlled such that arbitrary entry to, or return from, the protected domain is not possible.

c) The user or application parameters passed to the protected domain by addresses are validated with respect to the protected domain's address space, and those passed by value are validated with respect to the values expected by the protected domain.

d) The security domains of subjects are distinct except for controlled sharing via the TSF.

User notes

This family is needed whenever confidence is required that the TSF has not been subverted.

In order to obtain the equivalent of a reference monitor, the components FPT_SEP.2 (SFP domain separation) or FPT_SEP.3 (Complete reference monitor) from this family must be used in conjunction with FPT_RVM.1 (Non-bypassability of the TSP), and ADV_INT.3 (Minimisation of complexity). Further, if complete reference mediation is required, the components from Class FDP User data protection must cover all objects.

FPT_SEP.1 TSF domain separation

Without a separate protected domain for the TSF, there can be no assurance that the TSF has not been subjected to any tampering attacks by untrusted subjects. Such attacks may involve modification of the TSF code and/or TSF data structures.

FPT_SEP.2 SFP domain separation

The most important function provided by a TSF is the enforcement of its SFPs. In order to simplify the design and increase the likelihood that those significant SFPs exhibit the characteristics of a reference monitor (RM), in particular, being tamperproof, they must be in a domain distinct from the remainder of the TSF.

Evaluator application notes

It is possible that a reference monitor in a layered design may provide functions beyond those of the SFPs. This arises out of the practical nature of layered software design. The goal should be to minimise the non-SFP related functions.

Note that it is acceptable for the reference monitors for all included SFPs to be in a single distinct reference monitor domain, as well as having multiple reference monitor domains (each enforcing one or more SFPs). If multiple reference monitor domains for SFPs are present, it is acceptable for them to be either peers or in a hierarchical relationship.

For FPT_SEP.2.1, the phrase “unisolated portion of the TSF” refers to that portion of the TSF consisting of those functions in the TSF not covered by FPT_SEP.2.3.

Operations

Assignment:

For FPT_SEP.2.3, the PP/ST author should specify the access control and/or information flow control SFPs in the TSP that should have a separate domain.

FPT_SEP.3 Complete reference monitor

The most important function provided by a TSF is the enforcement of its SFPs. This component builds upon the intentions of the previous component by requiring that *all* access control and/or information flow control FSPs be enforced in a domain distinct from the remainder of the TSF. This further simplifies the design and increases the likelihood that the characteristics of a reference monitor (RM), in particular, being tamperproof, are found in the TSF.

Evaluator application notes

It is possible that a reference monitor in a layered design may provide functions beyond those of the SFPs. This arises out of the practical nature of layered software design. The goal should be to minimise the non-SFP related functions.

Note that it is acceptable for the reference monitors for all included SFPs to be in a single distinct reference monitor domain, as well as having multiple reference monitor domains (each enforcing one or more SFPs). If multiple reference monitor domains for SFPs are present, it is acceptable for them to be either peers or in a hierarchical relationship.

J.12 State synchrony protocol (FPT_SSP)

Distributed systems may give rise to greater complexity than monolithic systems through the potential for differences in state between parts of the system, and through delays in communication. In most cases, synchronisation of state between distributed functions involves an exchange protocol, not a simple action. When malice exists in the distributed environment of these protocols, more complex defensive protocols are required.

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

User notes

Some states may never be synchronised, or the transaction cost may be too high for practical use; encryption key revocation is an example, where knowing the state after the revocation action is initiated can never be known. Either the action was taken and acknowledgment cannot be sent, or the message was ignored by hostile communication partners and the revocation never occurred. Indeterminacy is unique to distributed systems. Indeterminacy and state synchrony are related, and the same solution may apply. It is futile to design for indeterminate states; the PP/ST author should express other requirements in such cases (e.g. raise an alarm, audit the event).

FPT_SSP.1 Simple trusted acknowledgement

User application notes

In this component, the TSF must supply an acknowledgement to another part of the TSF when requested. This acknowledgement should indicate that one part of a distributed TOE successfully received an unmodified transmission from a different part of the distributed TOE.

FPT_SSP.2 Mutual trusted acknowledgement

User application notes

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.

For example, the local TSF transmits some data to a remote part of the TSF. The remote part of the TSF acknowledges the successful receipt of the data and requests that the sending TSF confirm that it receives the acknowledgement. This mechanism provides additional confidence that both parts of the TSF involved in the data transmission know that the transmission completed successfully.

J.13 Time stamps (FPT_STM)

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

User notes

It is the responsibility of the PP/ST author to clarify the meaning of the phrase “reliable time stamp”, and to indicate where the responsibility lies in determining the acceptance of trust.

FPT_STM.1 Reliable time stamps

User application notes

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)

In a distributed or composite system environment, a TOE may need to exchange TSF data (e.g. the SFP-attributes associated with data, audit information, identification information) with another trusted IT Product. This family defines the requirements for sharing and consistent interpretation of these attributes between the TSF of the TOE and that of a different trusted IT Product. 0

User notes

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 FPT_ITI family. These components specify requirements for the TSF to be able to detect or detect and correct modifications to TSF data in transit.

FPT_TDC.1 Inter-TSF basic TSF data consistency

User application notes

The TSF is responsible for maintaining the consistency of TSF data used by or associated with the specified function and that are common between two or more trusted systems. For example, the TSF data of two different systems may have different conventions internally. For the TSF data to be used properly (e.g. to afford the user data the same protection as within the TOE) by the receiving trusted IT product, the TOE and the other trusted IT product must use a pre-established protocol to exchange TSF data.

Operations

Assignment:

In FPT_TDC.1.1, the PP/ST author should define the list of TSF data types, for which the TSF shall provide the capability to consistently interpret, when shared between the TSF and another trusted IT product. 0

In FPT_TDC.1.2, the PP/ST should assign the list of interpretation rules to be applied by the TSF. 0

J.15 Internal TOE TSF data replication consistency (FPT_TRC)

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.

User notes

The method of ensuring consistency is not specified in this component. It could be attained through a form of transaction logging (where appropriate transactions are “rolled back” to a site upon reconnection); it could be updating the replicated data through a synchronisation protocol. If a particular protocol is necessary for a PP/ST, it can be specified through refinement.

It may be impossible to synchronise some states, or the cost of such synchronisation may be too high. Examples of this situation are communication channel and encryption key revocations. Indeterminate states may also occur; if a specific behaviour is desired, it should be specified via refinement.

FPT_TRC.1 Internal TSF consistency

Operations

Assignment:

In FPT_TRC.1.2, the PP/ST author should specify the list of SFs dependent on TSF data replication consistency.

J.16 TSF self test (FPT_TST)

The family defines the requirements for the self-testing of the TSF with respect to some expected correct operation. Examples are interfaces to enforcement functions, and sample arithmetical operations on critical parts of the TOE. These tests can be carried out at start-up, periodically, at the request of an authorised user, or when other conditions are met. The actions to be taken by the TOE as the result of self testing are defined in other families.

The requirements of this family are also needed to detect the corruption of TSF executable code (i.e. TSF software) and TSF data by various failures that do not necessarily stop the TOE's operation (which would be handled by other families). These checks must be performed because these failures may not necessarily be prevented. Such failures can occur either because of unforeseen failure modes or associated oversights in the design of hardware, firmware, or software, or because of malicious corruption of the TSF due to inadequate logical and/or physical protection.

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.

User notes

The term “correct operation of the TSF” refers primarily to the operation of the TSF software and the integrity of the TSF data. The abstract machine upon which the TSF software is implemented is tested via dependency on FPT_AMT.

FPT_TST.1 TSF testing

User 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.

Evaluator application notes

It is acceptable for the functions that are available to the authorised user for periodic testing to be available only in an off-line or maintenance mode. Controls should be in place to limit access during these modes to authorised users.

Operations

Selection:

In FPT_TST.1 the PP/ST author should specify when the TSF will execute the TSF test; during initial start-up, periodically during normal operation, at the request of an authorised user, at other conditions. In the case of the latter option, the PP/ST author should also assign what those conditions are via the following assignment.

Assignment:

In FPT_TST.1.1 the PP/ST author should, if selected, specify the conditions under which the self test should take place.

Annex K

Resource utilisation (FRU)

This class provides three families that support the availability of required resources such as processing capability and/or storage capacity. The family Fault Tolerance provides protection against unavailability of capabilities caused by failure of the TOE. The family Priority of Service ensures that the resources will be allocated to the more important or time-critical tasks, and cannot be monopolised by lower priority tasks. The family Resource Allocation provides limits on the use of available resources, therefore preventing users from monopolising the resources.

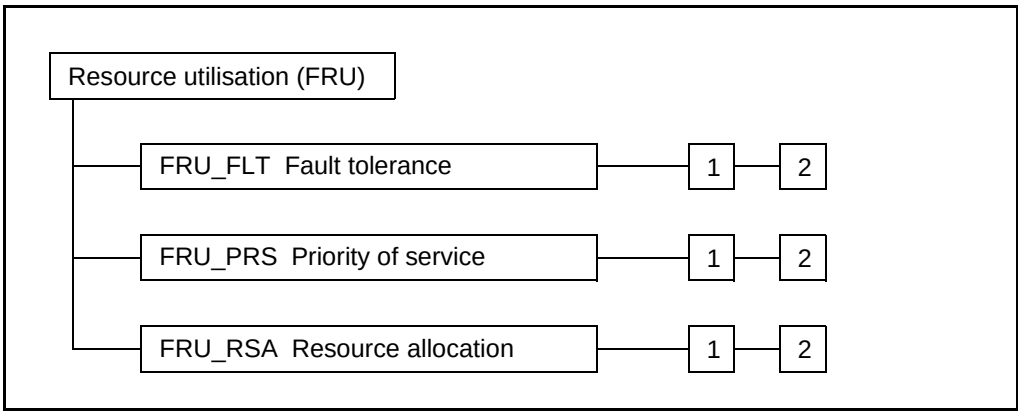

Figure K.1 - Resource utilisation class decomposition

K.1 Fault tolerance (FRU_FLT)

This family provides requirements for the availability of capabilities even in the case of failures. Examples of such failures are power failure, hardware failure, or software error. In case of these errors, if so specified, the TOE will maintain the specified capabilities. The PP/ST author could specify, for example, that a TOE used in a nuclear plant will continue the operation of the shut-down procedure in the case of power-failure or communication-failure.

User notes

Because the TOE can only continue its correct operation if the TSP is enforced, there is a requirement that the system must remain in a secure state after a failure. This capability is provided by FPT_FLS.1.

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 monopolising).

FRU_FLT.1 Degraded fault tolerance

User 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. Examples of general failures are flooding of the computer room, short term power interruption, breakdown of a CPU or host, software failure, or buffer overflow.

Operations

Assignment:

In FRU_FLT.1.1 the PP/ST author should specify the list of TOE capabilities the TOE will maintain during and after a specified failure.

In FRU_FLT.1.1 the PP/ST author should specify the list of type of failures against which the TOE has to be explicitly protected. If a failure in this list occurs, the TOE will be able to continue its operation.

FRU_FLT.2 Limited fault tolerance

User application notes

This component is intended to specify against what type of failures the TOE must be resistant. Since it would be difficult to describe all specific failures, categories of failures may be specified. Examples of general failures are flooding of the computer room, short term power interruption, breakdown of a CPU or host, software failure, or overflow of buffer.

Operations

Assignment:

In FRU_FLT.2.1 the PP/ST author should specify the list of type of failures against which the TOE has to be explicitly protected. If a failure in this list occurs, the TOE will be able to continue its operation.

K.2 Priority of service (FRU_PRS)

The requirements of this family allow the TSF to control the use of resources within the TSC by users and subjects such that high priority activities within the TSC will always be accomplished without interference or delay due to low priority activities. In other words, time critical tasks will not be delayed by tasks that are less time critical.

This family could be applicable to several types of resources, for 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.

User notes

The audit requirement states that all reasons for rejection should be audited. It is left to the developer to argue that an operation is not rejected but delayed.

FRU_PRS.1 Limited priority of service

User application notes

This component defines priorities for a subject, and the resources for which this priority will be used. If a subject attempts to take action on a resource controlled by the Priority of Service requirements, the access and/or time of access will be dependent on the subject's priority, the priority of the currently acting subject, and the priority of the subjects still in the queue.

Operations

Assignment:

For FRU_PRS.1.2, the PP/ST author should specify the list of controlled resources for which the TSF enforces priority of service (e.g. resources such as processes, disk space, memory, bandwidth).

FRU_PRS.2 Full priority of service

User application notes

This component defines priorities for a subject. All shareable resources in the TSC will be subjected to the Priority of Service mechanism. If a subject attempts to take action on a shareable TSC resource, the access and/or time of access will be dependent on the subject's priority, the priority of the currently acting subject, and the priority of the subjects still in the queue.

K.3 Resource allocation (FRU_RSA)

The requirements of this family allow the TSF to control the use of resources within the TSC by users and subjects such that unauthorised denial of service will not take place by means of monopolisation of resources by other users or subjects.

User notes

Resource allocation rules allow the creation of quotas or other means of defining limits on the amount of resource space or time that may be allocated on behalf of a specific user or subjects. These rules may, for example:

- Provide for object quotas that constrain the number and/or size of objects a specific user may allocate.
- Control the allocation/deallocation of preassigned resource units where these units are under the control of the TSF.

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 (e.g. a single user should not allocate all the available space) and subjects. Since resource allocation often goes beyond the lifespan of a subject (i.e. files often exist longer than the applications that generated them), and multiple instantiations of subjects by the same user should not negatively affect other users too much, the components allow that the allocation limits are related to the users. In some situations the resources are allocated by a subject (e.g. main memory or CPU cycles). In those instances the components allow that the resource allocation be on the level of subjects.

This family imposes requirements on resource allocation, not on the use of the resource itself. The audit requirements therefore, as stated, also apply to the allocation of the resource, not to the use of the resource.

FRU_RSA.1 Maximum quotas

User application notes

This component provides requirements for quota mechanisms that apply to only a specified set of the shareable resources in the TOE. The requirements allow the quotas to be associated with a user, possibly assigned to groups of users or subjects as applicable to the TOE.

Operations

Assignment:

In FRU_RSA.1.1, the PP/ST author should specify the list of controlled resources for which maximum resource allocation limits are required (e.g. processes, disk space, memory, bandwidth). If all resources in the TSC need to be included, the words “all TSC resources” can be specified.

Selection:

In FRU_RSA.1.1, the PP/ST author should select whether the maximum quotas apply to individual users, to a defined group of users, or subjects or any combination of these.

In FRU_RSA.1.1, the PP/ST author should select whether the maximum quotas are applicable to any given time (simultaneously), or over a specific time interval.

FRU_RSA.2 Minimum and maximum quotas

User application notes

This component provides requirements for quota mechanisms that apply to a specified set of the shareable resources in the TOE. The requirements allow the quotas to be associated with a user, or possibly assigned to groups of users as applicable to the TOE.

Operations

Assignment:

In FRU_RSA.2.1, the PP/ST author should specify the controlled resources for which maximum and **minimum** resource allocation limits are required (e.g. processes, disk space, memory, bandwidth). If all resources in the TSC need to be included, the words “all TSC resources” can be specified.

Selection:

In FRU_RSA.2.1, the PP/ST author should select whether the maximum quotas apply to individual users, to a defined group of users, or subjects or any combination of these.

In FRU_RSA.2.1, the PP/ST author should select whether the maximum quotas are applicable to any given time (simultaneously), or over a specific time interval.

Assignment:

In FRU_RSA.2.2, the PP/ST author should specify the controlled resources for which a minimum allocation limit needs to be set (e.g. processes, disk space, memory, bandwidth). If all resources in the TSC need to be included the words “all TSC resources” can be specified.

Selection:

In FRU_RSA.2.2, the PP/ST author should select whether the minimum quotas apply to individual users, to a defined group of users, or subjects or any combination of these.

In FRU_RSA.2.2, the PP/ST author should select whether the minimum quotas are applicable to any given time (simultaneously), or over a specific time interval.

Annex L

TOE access (FTA)

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.

Figure L.1 shows the decomposition of this class into its constituent components.

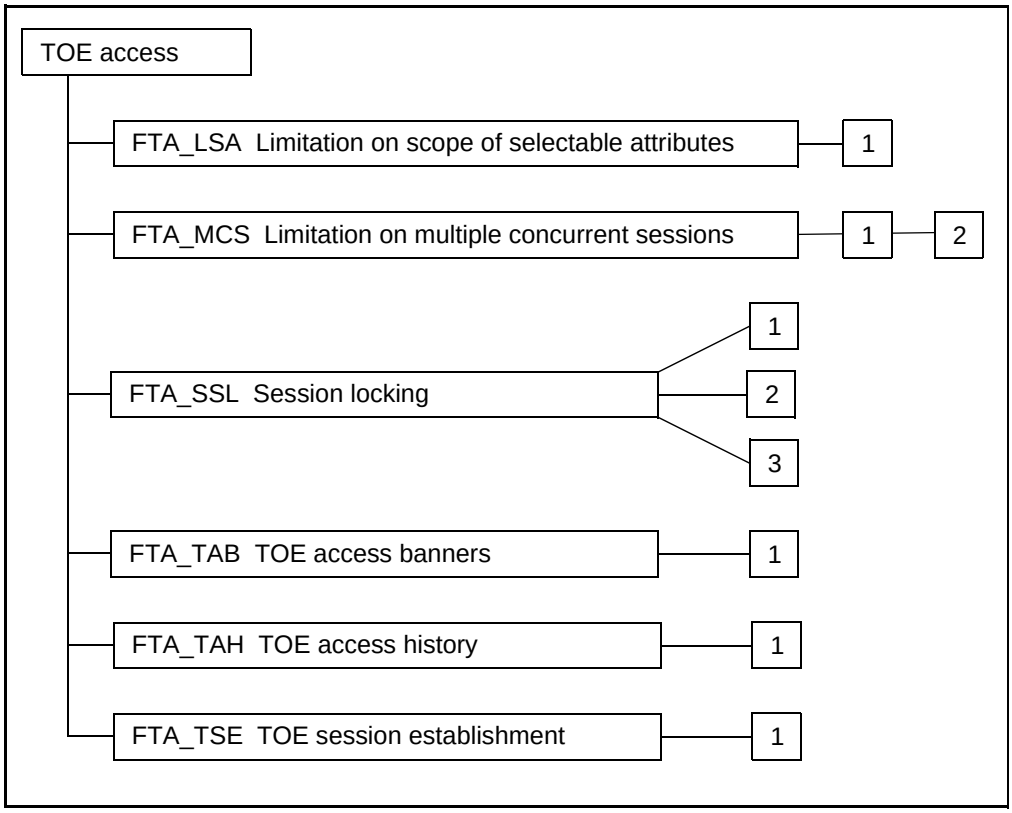

Figure L.1 - TOE access class decomposition

L.1 Limitation on scope of selectable attributes (FTA_LSA)

This family defines requirements that will limit the session security attributes a user may select, and the subjects to which a user may be bound, based on: the method of access; the location or port of access; and/or the time (e.g. time-of-day, day-of-week).

User notes

This family provides the capability for a PP/ST author to specify requirements for the TSF to place limits on the domain of an authorised user's security attributes based on an environmental condition. For example, a user may be allowed to establish a "secret session" during normal business hours but outside those hours the same user may be constrained to only establishing "unclassified sessions". The identification of relevant constraints on the domain of selectable attributes can be achieved through the use of the selection operation. These constraints can be applied on an attribute-by-attribute basis. When there exists a need to specify constraints on multiple attributes this component will have to be replicated for each attribute. Examples of attributes that could be used to limit the session security attributes are:

a) The method of access can be used to specify in which type of environment the user will be operating (e.g. file transfer protocol, terminal, vtam).

b) The location of access can be used to constrain the domain of a user's selectable attributes based on a user's location or port of access. This capability is of particular use in environments where dial-up facilities or network facilities are available.

c) The time of access can be used to constrain the domain of a user's selectable attributes. For example, ranges may be based upon time-of-day, day-of-week, or calendar dates. This constraint provides some operational protection against user actions that could occur at a time where proper monitoring or where proper procedural measures may not be in place.

FTA_LSA.1Limitation on scope of selectable attributes

Operations

Assignment:

In FTA_LSA.1.1 the PP/ST author should specify the set of session security attributes that are to be constrained. Examples of these session security attributes are user clearance level, integrity level and roles.

In FTA_LSA.1.1 the PP/ST author should specify the set of attributes that can be use to determine the scope of the session security attributes. Examples of such attributes are user identity, originating location, time of access, and method of access.

L.2 Limitation on multiple concurrent sessions (FTA_MCS)

This family defines how many sessions a user may have at the same time (concurrent sessions). This number of concurrent sessions can either be set for a group of users or for each individual user.

FTA_MCS.1 Basic limitation on multiple concurrent sessions

User application notes

This component allows the system to limit the number of sessions in order to effectively use the resources of the TOE.

Operations

Assignment:

In FTA_MCS.1.2 the PP/ST author should specify the default number of maximum concurrent sessions to be used.

FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions

User application notes

This component provides additional capabilities over those of FTA_MCS.1, by allowing further constraints to be placed on the number of concurrent sessions that users are able to invoke. These constraints are in terms of a user's security attributes, such as a user's identity, or membership of a role.

Operations

Assignment:

For FTA_MCS.2.1 the PP/ST author should specify the rules that determine the maximum number of concurrent sessions. An example of a rule is "maximum number of concurrent sessions is one if the user has a classification level of 'secret' and five otherwise".

In FTA_MCS.2.2 the PP/ST author should specify the default number of maximum concurrent sessions to be used.

L.3 Session locking (FTA_SSL)

This family defines requirements for the TSF to provide the capability for locking and unlocking of interactive sessions (e.g. keyboard locking).

When a user is directly interacting with subjects in the TOE (interactive session), the user's terminal is vulnerable if left unattended. This family provides requirements for the TSF to disable (lock) the terminal or terminate the session after a specified period of inactivity, and for the user to initiate the disabling (locking) of the terminal. To reactivate the terminal, an event specified by the PP/ST author, such as the user re-authentication must occur.

A user is considered inactive, if he/she has not provided any stimulus to the TOE for a period of time.

A PP/ST author should consider whether FTP_TRP.1 Trusted path should be included. In that case, the function 'session locking' should be included in the operation in FTP_TRP.1.

FTA_SSL.1 TSF-initiated session locking

User 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/ST author to specify what events will unlock the session. These events may be related to the terminal (e.g. fixed set of keystrokes to unlock the session), the user (e.g. reauthentication), or time.

Operations

Assignment:

In FTA_SSL.1.1 the PP/ST author should specify the interval of user inactivity that will trigger the locking of an interactive session. If so desired the PP/ST author could, through the assignment, specify that the time interval is left to the authorised administrator or the user. The management functions in the FMT class can specify the capability to modify this time interval, making it the default value.

In FTA_SSL.1.2 the PP/ST author should specify the event(s) that should occur before the session is unlocked. Examples of such an event are: "user re-authentication" or "user enters unlock key-sequence".

FTA_SSL.2 User-initiated locking

User application notes

FTA_SSL.2 User-initiated locking, provides the capability for an authorised user to lock and unlock his/her own terminal. This would provide authorised users with the ability to effectively block further use of their active sessions without having to terminate the active session.

If devices are overwritten, the replacement contents need not be static (i.e. 'screen savers' are permitted).

Operations

Assignment:

In FTA_SSL.2.2 the PP/ST author should specify the event(s) that should occur before the session is unlocked. Examples of such an event are: "user re-authentication", or "user enters unlock key-sequence".

FTA_SSL.3 TSF-initiated termination

User application notes

FTA_SSL.3 TSF-initiated termination, requires that the TSF terminate an interactive user session after a period of inactivity.

The PP/ST author should be aware that a session may continue after the user terminated his/her activity, for example, background processing. This requirement would terminate this background subject after a period of inactivity of the user without regard to the status of the subject.

Operations

Assignment:

In FTA_SSL.3.1 the PP/ST author should specify the interval of user inactivity that will trigger the termination of an interactive session. If so desired, the PP/ST author could, through the assignment, specify that the interval is left to the authorised administrator or the user. The management functions in the FMT class can specify the capability to modify this time interval, making it the default value.

L.4 TOE access banners (FTA_TAB)

Prior to identification and authentication, TOE access requirements provide the ability for the TOE to display an advisory warning message to potential users pertaining to appropriate use of the TOE.

FTA_TAB.1 Default TOE access banners

This component requires that there is an advisory warning regarding the unauthorised use of the TOE. A PP/ST author could refine the requirement to include a default banner.

L.5 TOE access history (FTA_TAH)

This family defines requirements for the TSF to display to users, upon successful session establishment to the TOE, a history of unsuccessful attempts to access the account. This history may include the date, time, means of access, and port of the last successful access to the TOE, as well as the number of unsuccessful attempts to access the TOE since the last successful access by the identified user.

FTA_TAH.1 TOE access history

This family can provide authorised users with information that may indicate the possible misuse of their user account.

This component request that the user is presented with the information. The user should be able to review the information, but is not forced to do so. If a user so desires he might, for example, create scripts that ignore this information and start other processes.

Operations

Selection:

In FTA_TAH.1.1, the PP/ST author should select the security attributes of the last successful session establishment that will be shown at the user interface. The items are: date, time, method of access (such as ftp), and/or location (e.g. terminal 50).

In FTA_TAH.1.2, the PP/ST author should select the security attributes of the last unsuccessful session establishment that will be shown at the user interface. The items are: date, time, method of access (such as ftp), and/or location (e.g. terminal 50).

L.6 TOE session establishment (FTA_TSE)

This family defines requirements to deny an user permission to establish a session with the TOE based on attributes such as the location or port of access, the user's security attribute (e.g. identity, clearance level, integrity level, membership in a role), ranges of time (e.g. time-of-day, day-of-week, calendar dates) or combinations of parameters.

User notes

This family provides the capability for the PP/ST author to specify requirements for the TOE to place constraints on the ability of an authorised user to establish a session with the TOE. The identification of relevant constraints can be achieved through the use of the selection operation. Examples of attributes that could be used to specify the session establishment constraints are:

a) The location of access can be used to constrain the ability of a user to establish an active session with the TOE, based on the user's location or port of access. This capability is of particular use in environments where dial-up facilities or network facilities are available.

b) The user's security attributes can be used to place constraints on the ability of a user to establish an active session with the TOE. For example, these attributes would provide the capability to deny session establishment based on any of the following:

- a user's identity;
- a user's clearance level;
- a user's integrity level; and
- a user's membership in a role.

This capability is particularly relevant in situations where authorisation or login may take place at a different location from where TOE access checks are performed.

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.

FTA_TSE.1 TOE session establishment

Operations

Assignment:

In FTA_TSE.1.1 the PP/ST author should specify the attributes that can be used to restrict the session establishment. Example of possible attributes are user identity, originating location (e.g. no remote terminals), time of access (e.g. outside hours), or method of access (e.g. X-windows).

Annex M

Trusted path/channels (FTP)

Users often need to perform functions through direct interaction with the TSF. A trusted path provides confidence that a user is communicating directly with the TSF whenever it is invoked. A user's response via the trusted path guarantees that untrusted applications cannot intercept or modify the user's response. Similarly, trusted channels are one approach for secure communication between the TSF and remote IT products.

Part 2, Figure 1.2, page 4, illustrates the relationships between the various types of communication that may occur within a TOE or network of TOEs (i.e. Internal TOE transfers, Inter-TSF transfers, and Import/Export Outside of TSF Control) and the various forms of trusted paths and channels.

Absence of a trusted path may allow breaches of accountability or access control in environments where untrusted applications are used. These applications can intercept user-private information, such as passwords, and use it to impersonate other users. As a consequence, responsibility for any system actions cannot be reliably assigned to an accountable entity. Also, these applications could output erroneous information on an unsuspecting user's display, resulting in subsequent user actions that may be erroneous and may lead to a security breach.

Figure M.1 shows the decomposition of this class into its constituent components.

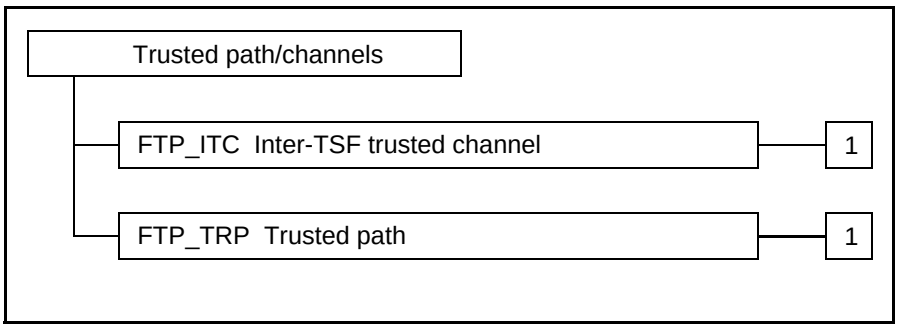

Figure M.1 - Trusted path/channels class decomposition

M.1 Inter-TSF trusted channel (FTP_ITC)

This family defines the rules for the creation of a trusted channel connection that goes between the TSF and another trusted IT product for the performance of security critical operations between the products. An example of such a security critical operation is the updating of the TSF authentication database by the transfer of data from a trusted product whose function is the collection of audit data.

FTP_ITC.1 Inter-TSF trusted channel

User application notes

This component should be used when a trusted communication channel between the TSF and another trusted IT product is required.

Operations

Selection:

In FTP_ITC.1.2, the PP/ST author must specify whether the local TSF, the remote trusted IT product, or both shall have the capability to initiate the trusted channel.

Assignment:

In FTP_ITC.1.3, the PP/ST author should specify the functions for which a trusted channel is required. Examples of these functions may include transfer of user, subject, and/or object security attributes and ensuring consistency of TSF data.

M.2 Trusted path (FTP_TRP)

This family defines the requirements to establish and maintain trusted communication to or from users and the TSF. A trusted path may be required for any security-relevant interaction. Trusted path exchanges may be initiated by a user during an interaction with the TSF, or the TSF may establish communication with the user via a trusted path.

FTP_TRP.1 Trusted path

User application notes

This component should be used when trusted communication between a user and the TSF is required, either for initial authentication purposes only or for additional specified user operations.

Operations

Selection:

In FTP_TRP.1.1, the PP/ST author should specify whether the trusted path must be extended to remote and/or local users.

In FTP_TRP.1.2, the PP/ST author should specify whether the TSF, local users, and/or remote users should be able to initiate the trusted path.

In FTP_TRP.1.3, the PP/ST author should specify whether the trusted path is to be used for initial user authentication and/or for other specified services.

Assignment:

In FTP_TRP.1.3, if selected, the PP/ST author should identify other services for which trusted path is required, if any.