D R A F T

Common Criteria for Information Technology Security Evaluation

Part 2 : Annexes

19 December 1997

Version 2.0 Draft

CCIB-97/082AR

D R A F T

Foreword

The CC Project Sponsoring Organisations are pleased to provide this **version 2.0 draft** of the *Common Criteria for Information Technology Security Evaluation*. This version is to be used by CC Project Sponsoring Organisations for their internal review. It will also be made available for information purposes to ISO/IEC, JTC 1, SC27/WG3 experts via the NIST website (see below). As previously agreed with WG3, the Common Criteria Implementation Board (CCIB) will continue to develop this document though early April, 1998. **Version 2.0 pre-final** will be released at that time, made available to WG 3 experts via the NIST website, and will also be provided in paper form at the WG3 meeting in Stockholm, Sweden.

CANADA:

Communications Security Establishment
Criteria Coordinator
R2B IT Security Standards and Initiatives
P.O. Box 9703, Terminal
Ottawa, Canada K1G 3Z4
Tel: +1.613.991.7409, Fax: +1.613.991.7411
E-mail: criteria@cse-cst.gc.ca
WWW: http://www.cse.dnd.ca/cse/english/cc.html
FTP: ftp://ftp.cse.dnd.ca/pub/criteria/CC1.0

FRANCE:

Service Central de la Sécurité des Systèmes d'Information (SCSSI)
Centre de Certification de la Sécurité des Technologies de l'Information
18, rue du docteur Zamenhof
F-92131 Issy les Moulineaux
France
Tel: +33.1.41463784, Fax: +33.1.41463701
E-mail: ssi20@calva.net

GERMANY:

D R A F T

German Information Security Agency (GISA)
Bundesamt für Sicherheit in der Informationstechnik
Abteilung V
Postfach 20 03 63
D-53133 Bonn
Germany
Tel: +49.228.9582.300, Fax: +49.228.9582.427
E-mail: cc@bsi.de
WWW: http://www.bsi.bund.de

NETHERLANDS:

Netherlands National Communications Security Agency
P.O. Box 20061
NL 2500 EB The Hague
The Netherlands
Tel: +31.70.3485637, Fax: +31.70.3486503
E-mail: criteria@nlncsa.minbuza.nl
WWW: http://www.tno.nl/instit/fel/refs/cc.html

UNITED KINGDOM:

Communications-Electronics Security Group
Compusec Evaluation Methodology
P.O. Box 144
Cheltenham GL52 5UE
United Kingdom
Tel: +44.1242.221.491 ext. 4134, Fax: +44.1242.235.233
E-mail: criteria@cesg.gov.uk
WWW: http://www.cesg.gov.uk/cchtml
FTP: ftp://ftp.itsec.gov.uk/pub/ccv1.0

UNITED STATES - NIST:

National Institute of Standards and Technology
Computer Security Division
820 Diamond, MS: NN426
Gaithersburg, Maryland 20899
U.S.A.
Tel: +1.301.975.2934, Fax: +1.301.948.0279
E-mail: criteria@nist.gov
WWW: http://csrc.nist.gov/cc

UNITED STATES - NSA:

D R A F T

National Security Agency
Attn: V2, Common Criteria Technical Advisor
Fort George G. Meade, Maryland 20755-6740
U.S.A.
Tel: +1.410.859.4458, Fax: +1.410.684.7512
E-mail: common_criteria@radium.ncsc.mil
WWW: http://www.radium.ncsc.mil/tpep/

D R A F T

Table of Contents

Annex A
Security functional requirements application notes 1

A.1 Overview 1
A.1.1 Class structure 1
A.1.2 Family structure 2
A.1.3 Component structure 3
A.2 Dependency Table 4

Annex B
Assurance classes, families, and components 11

Class FAU
Security Audit 13

FAU_ARP Security Audit Automatic Response 16
FAU_ARP.1 Security Alarms 16
FAU_GEN Security Audit Data Generation 17
FAU_GEN.1 Audit Data Generation 18
FAU_GEN.2 User Identity Generation 19
FAU_SAA Security Audit Analysis 20
FAU_SAA.1 Imminent Violation Analysis 20
FAU_SAA.2 Profile Based Anomaly Detection 20
FAU_SAA.3 Simple Attack Heuristics 22
FAU_SAA.4 Complex Attack Heuristics 23
FAU_SAR Security Audit Review 26
FAU_SAR.1 Audit Review 26
FAU_SAR.2 Restricted Audit Review 27
FAU_SAR.3 Selectable Audit Review 27
FAU_SEL Security Audit Event Selection 28
FAU_SEL.1 Selective Audit 28
FAU_STG Security Audit Event Storage 30
FAU_STG.1 Permanent Audit Trail Storage 30
FAU_STG.2 Guarantees of Audit Data Availability 30
FAU_STG.3 Action in Case of Possible Audit Data Loss 31
FAU_STG.4 Prevention of Audit Data Loss 31

Class FCO
Communication 33

FCO_NRO Non-Repudiation of Origin 34
FCO_NRO.1 Selective Proof of Origin 35
FCO_NRO.2 Enforced Proof of Origin 36
FCO_NRR Non-Repudiation of Receipt 37
FCO_NRR.1 Selective Proof of Receipt 38
FCO_NRR.2 Enforced Proof of Receipt 39

D R A F T

Class FCS

Cryptographic Support 41

FCS_CKM Cryptographic Key Management 43
FCS_CKM.1 Cryptographic Key Generation 43
FCS_CKM.2 Standards-Based Cryptographic Key Generation 44
FCS_CKM.3 Cryptographic Key Distribution 44
FCS_CKM.4 Standards-Based Cryptographic Key Distribution 44
FCS_CKM.5 Cryptographic Key Access 45
FCS_CKM.6 Standards-Based Cryptographic Key Access 45
FCS_CKM.7 Cryptographic Key Destruction 45
FCS_CKM.8 Standards-Based Cryptographic Key Destruction 46

FCS_COP Cryptographic Operation 47
FCS_COP.1 Cryptographic Operation 47
FCS_COP.2 Standards-Based Cryptographic Operation 48

Class FDP

User Data Protection 49

FDP_ACC Access Control Policy 54
FDP_ACC.1 Subset Access Control 55
FDP_ACC.2 Complete Access Control 55

FDP_ACF Access Control Functions 56
FDP_ACF.1 Security Attribute Based Access Control 56
FDP_ACF.2 Access Authorisation 57
FDP_ACF.3 Access Authorisation and Denial 58
FDP_ACF.4 Fixed Access Control 58

FDP_DAU Data Authentication 60
FDP_DAU.1 Basic Data Authentication 60
FDP_DAU.2 Data Authentication with Identity of Guarantor 60

FDP_ETC Export to Outside TSF Control 62
FDP_ETC.1 Export of User Data Without Security Attributes 62
FDP_ETC.2 Export of User Data With Security Attributes 62

FDP_IFC Information Flow Control Policy 64
FDP_IFC.1 Subset Information Flow Control 65
FDP_IFC.2 Complete Information Flow Control 65

FDP_IFF Information Flow Control Functions 66
FDP_IFF.1 Simple Security Attributes 66
FDP_IFF.2 Hierarchical Security Attributes 67
FDP_IFF.3 Limited Illicit Information Flows 68
FDP_IFF.4 Partial Elimination of Illicit Information Flows 69
FDP_IFF.5 No Illicit Information Flows 69
FDP_IFF.6 Illicit Information Flow Monitoring 69
FDP_IFF.7 Information Flow Authorisation 70
FDP_IFF.8 Information Flow Authorisation and Denial 70

FDP_ITC Import from Outside TSF Control 72
FDP_ITC.1 Import of User Data Without Security Attributes 73
FDP_ITC.2 Import of User Data with Security Attributes 73

FDP_ITT Internal TOE Transfer 75
FDP_ITT.1 Basic Internal Transfer Protection 75

D R A F T

	FDP_ITT.2 Transmission Separation by Attribute	75
	FDP_ITT.3 Integrity Monitoring	76
	FDP_ITT.4 Attribute-Based Integrity Monitoring	77
FDP_RIP	**Residual Information Protection**	**78**
	FDP_RIP.1 Subset Residual Information Protection	78
	FDP_RIP.2 Full Residual Information Protection	79
FDP_ROL	**Rollback**	**80**
	FDP_ROL.1 Basic Rollback	80
	FDP_ROL.2 Advanced Rollback	81
FDP_SDI	**Stored Data Integrity**	**82**
	FDP_SDI.1 Stored Data Integrity Monitoring	82
	FDP_SDI.2 Stored Data Integrity Monitoring and Action	82
FDP_UCT	**Inter-TSF User Data Confidentiality Transfer Protection**	**84**
	FDP_UCT.1 Basic Data Exchange Confidentiality	84
FDP_UIT	**Inter-TSF User Data Integrity Transfer Protection**	**85**
	FDP_UIT.1 Basic Data Exchange Integrity	85
	FDP_UIT.2 Source Data Exchange Recovery	85
	FDP_UIT.3 Destination Data Exchange Recovery	86
	Class FIA	
	Identification and Authentication	**87**
FIA_AFL	**Authentication Failures**	**90**
	FIA_AFL.1 Basic Authentication Failure Handling	90
FIA_ATD	**User Attribute Definition**	**92**
	FIA_ATD.1 User Attribute Definition	92
FIA_SOS	**Specification of Secrets**	**93**
	FIA_SOS.1 Verification of Secrets	93
	FIA_SOS.2 TSF Generation of Secrets	93
FIA_UAU	**User Authentication**	**95**
	FIA_UAU.1 Timing of authentication	95
	FIA_UAU.2 User authentication before any action	95
	FIA_UAU.3 Unforgeable Authentication	96
	FIA_UAU.4 Single-use Authentication Mechanisms	96
	FIA_UAU.5 Multiple Authentication Mechanisms	96
	FIA_UAU.6 Re-authenticating	97
	FIA_UAU.7 Protected authentication feedback	98
FIA_UID	**User Identification**	**99**
	FIA_UID.1 Timing of Identification	99
	FIA_UID.2 User Identification before any action	99
FIA_USB	**User-Subject Binding**	**100**
	FIA_USB.1 User-Subject Binding	100
	Class FMT	
	Security Management	**101**
FMT_MOF	**Management of functions in TSF**	**103**
	FMT_MOF.1 Management of security functions behaviour	103
FMT_MSA	**Management of Security Attributes**	**105**
	FMT_MSA.1 Management of security attributes	105

D R A F T

FMT_MSA.2 Safe security attributes . . . 106
FMT_MSA.3 Static Attribute Initialisation . . . 106
FMT_MTD Management of TSF data . . . 107
FMT_MTD.1 Management of TSF data . . . 107
FMT_MTD.2 Management of limits on TSF data . . . 107
FMT_MTD.3 Safe TSF data . . . 108
FMT_REV Revocation . . . 109
FMT_REV.1 Revocation . . . 109
FMT_SAE Security Attribute Expiration . . . 110
FMT_SAE.1 Time-Limited Authorisation . . . 110
FMT_SMR Security Management Roles . . . 111
FMT_SMR.1 Security roles . . . 111
FMT_SMR.2 Restrictions on security roles . . . 111
FMT_SMR.3 Assuming roles . . . 112

Class FPR

Privacy . . . 113

FPR_ANO Anonymity . . . 115
FPR_ANO.1 Anonymity . . . 115
FPR_ANO.2 TSF Anonymity . . . 116
FPR_PSE Pseudonymity . . . 118
FPR_PSE.1 Pseudonymity . . . 119
FPR_PSE.2 Reversible Pseudonymity . . . 120
FPR_PSE.3 Alias Pseudonymity . . . 122
FPR_UNL Unlinkability . . . 124
FPR_UNL.1 Unlinkability . . . 124
FPR_UNO Unobservability . . . 126
FPR_UNO.1 Unobservability . . . 126
FPR_UNO.2 Authorised Administrator Observability . . . 127

Class FPT

Protection of the TOE Security Functions . . . 129

FPT_AMT Underlying Abstract Machine Test . . . 133
FPT_AMT.1 Abstract Machine Testing . . . 133
FPT_FLS Fail Secure . . . 135
FPT_FLS.1 Failure with Preservation of Secure State . . . 135
FPT_ITA Inter-TSF Availability of TSF Data . . . 136
FPT_ITA.1 Inter-TSF Availability Within a Defined Availability Factor . 136
FPT_ITC Inter-TSF Confidentiality of TSF Data . . . 137
FPT_ITC.1 Inter-TSF Confidentiality During Transmission . . . 137
FPT_ITI Inter-TSF Integrity of TSF Data . . . 138
FPT_ITI.1 Inter-TSF Detection of Modification . . . 138
FPT_ITI.2 Inter-TSF Detection and Correction of Modification . . . 139
FPT_ITT Internal TOE TSF Data Transfer . . . 140
FPT_ITT.1 Basic Internal TSF Data Transfer Protection . . . 140
FPT_ITT.2 TSF Data Transfer Separation . . . 140
FPT_ITT.3 TSF Data Integrity Monitoring . . . 141
FPT_PHP TSF Physical Protection . . . 142

D R A F T

FPT_PHP.1 Passive Detection of Physical Attack 142
FPT_PHP.2 Notification of Physical Attack 143
FPT_PHP.3 Resistance to Physical Attack 143
FPT_RCV Trusted Recovery 145
FPT_RCV.1 Manual Recovery 146
FPT_RCV.2 Automated Recovery 146
FPT_RCV.3 Automated Recovery without Undue Loss 147
FPT_RCV.4 Function Recovery 148
FPT_RPL Replay Detection and Prevention 149
FPT_RPL.1 Replay Detection and Prevention 149
FPT_RVM Reference Mediation 150
FPT_RVM.1 Non-Bypassability of the TSP 151
FPT_SEP Domain Separation 152
FPT_SEP.1 TSF Domain Separation 152
FPT_SEP.2 SFP Domain Separation 153
FPT_SEP.3 Complete Reference Monitor 153
FPT_SSP State Synchrony Protocol 154
FPT_SSP.1 Simple Trusted Acknowledgement 154
FPT_SSP.2 Mutual Trusted Acknowledgement 154
FPT_STM Time Stamps 155
FPT_STM.1 Reliable Time Stamps 155
FPT_TDC Inter-TSF TSF Data Consistency 156
FPT_TDC.1 Inter-TSF Basic TSF Data Consistency 156
FPT_TRC Internal TOE TSF Data Replication Consistency 157
FPT_TRC.1 Internal TOE Data Consistency 157
FPT_TST TSF Self Test 158
FPT_TST.1 TSF Testing 158

Class FRU
Resource Utilisation 161
FRU_FLT Fault Tolerance 162
FRU_FLT.1 Degraded Fault Tolerance 162
FRU_FLT.2 Limited Fault Tolerance 163
FRU_PRS Priority of Service 164
FRU_PRS.1 Limited Priority of Service 164
FRU_PRS.2 Full Priority of Service 164
FRU_RSA Resource Allocation 166
FRU_RSA.1 Maximum Quotas 166
FRU_RSA.2 Minimum and Maximum Quotas 167

Class FTA
TOE Access 169
FTA_LSA Limitation on Scope of Selectable Attributes 170
FTA_LSA.1 Limitation on Scope of Selectable Attributes 170
FTA_MCS Limitation on Multiple Concurrent Sessions 172
FTA_MCS.1 Basic Limitation on Multiple Concurrent Sessions 172
FTA_MCS.2 Per User Attribute Limitation on Multiple Concurrent Sessions 172

D R A F T

FTA_SSL **Session Locking 173**
FTA_SSL.1 TSF-initiated Session Locking 173
FTA_SSL.2 User-initiated Locking 174
FTA_SSL.3 TSF-initiated Termination 174
FTA_TAB **TOE Access Banners 175**
FTA_TAB.1 Default TOE Access Banners 175
FTA_TAH **TOE Access History 176**
FTA_TAH.1 TOE Access History 176
FTA_TSE **TOE Session Establishment 177**
FTA_TSE.1 TOE Session Establishment 178

Class FTP
Trusted Path/Channels 179
FTP_ITC **Inter-TSF Trusted Channel 181**
FTP_ITC.1 Inter-TSF Trusted Channel 181
FTP_TRP **Trusted Path 182**
FTP_TRP.1 Trusted Path 182

Annex C
CC observation report (CCOR) 183
C.1 Introduction 183
C.2 Format of observation report 183
C.2.1 Tag definitions for observation report 184
C.2.2 Example observations: 187

D R A F T

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 B.1 - Audit requirements construction rules 13
Figure B.2 - Security Audit class decomposition 15
Figure B.3 - Communication class decomposition 33
Figure B.4 - Cryptographic Support class decomposition 41
Figure B.5 - Cryptographic Support construction rules 42
Figure B.6 - User Data Protection class decomposition 51
Figure B.7 - User Data Protection class decomposition (cont.) 52
Figure B.8 - Identification and Authentication class decomposition 88
Figure B.9 - Identification and Authentication requirements construction rules 89
Figure B.10 - Security Management class decomposition 101
Figure B.11 - Privacy class decomposition 113
Figure B.12 - Protection of the TOE Security Functions class decomposition 130
Figure B.13 - Protection of the TOE Security Functions class decomposition (Cont.) 131
Figure B.14 - Resource Utilisation class decomposition 161
Figure B.15 - TOE Access class decomposition 169
Figure B.16 - Trusted Paths and Trusted Channels 179
Figure B.17 - Trusted Path / Channels class decomposition 180

D R A F T

D R A F T

List of Tables

Table A.1 - Dependencies for functional components 5

D R A F T

D R A F T

Annex A

Security functional requirements application notes

1 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 are similar to the main body of Part 2. The class, family, and component structures in this annex differ from that found in the main body of Part 2 since this annex is concerned with only those sections which are informative.

A.1 Overview

2 This section defines the content and presentation of the notes related to functional requirements of the CC. It provides guidance on the organisation of the requirements for the supporting information provided for new components to be included in a security target and to be evaluated.

A.1.1 Class structure

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

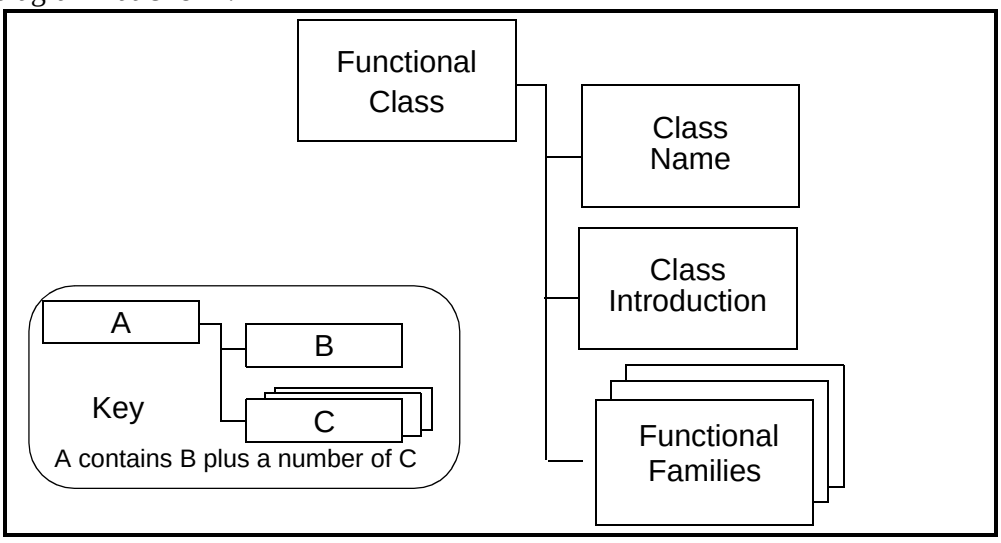

Figure A.1 - Functional class structure

D R A F T

A.1.1.1 Class name

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

A.1.1.2 Class introduction

5 The class introduction in this annex provides information about the construction rules to use families and components of the class to set up a consistent PP, ST or functional packages. 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

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

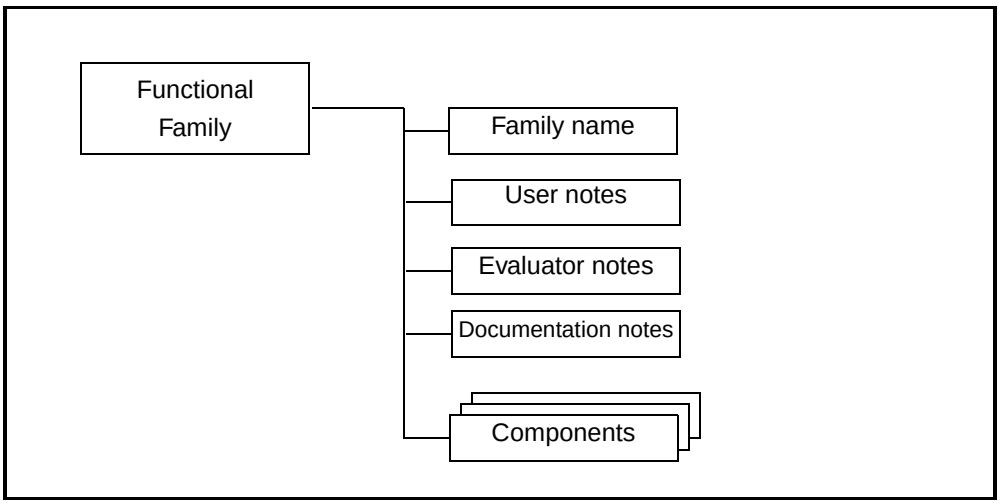

Figure A.2 - Functional family structure for application notes

A.1.2.1 Family name

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

A.1.2.2 User notes

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

D R A F T

A.1.2.3 Evaluator notes

9 The *evaluator notes* contain any information that is of interest to developers and evaluators of TOEs that claim compliance to 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 what needs to be documented to support the required functional behaviour, clarifications of meaning, and specification of the way to interpret specific requirements, as well as caveats and warnings of specific interest to evaluators.

A.1.2.4 Documentation notes

10 The *documentation notes* contain information that may be of interest to PP/ST authors when defining the set of expected information to be provided by the relevant documentation, part of the evaluation deliverables. The presentation is informative and is in the form of suggestions that are not considered as normative on the part of PP/ST authors.

11 These note sections are not mandatory and should appear only if appropriate.

A.1.3 Component structure

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

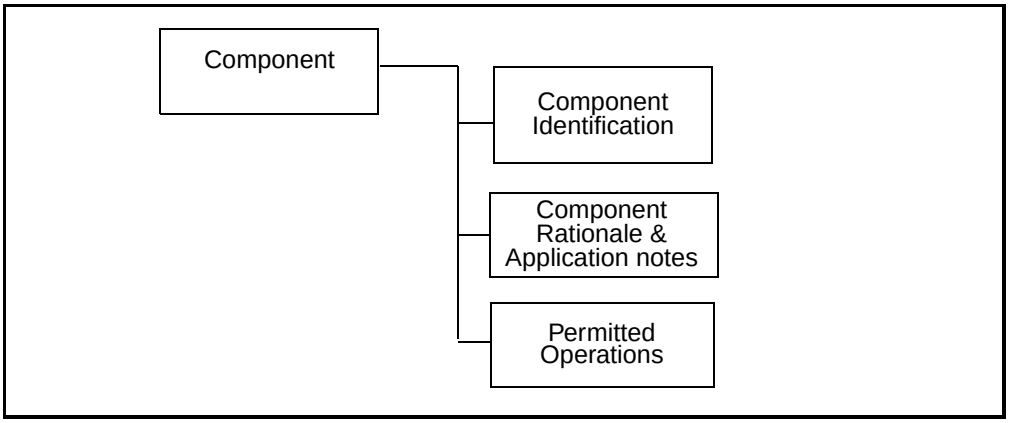

Figure A.3 - Functional component structure

A.1.3.1 Component identification

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

A.1.3.2 Component rationale and application notes

14 Any specific information related to the component should be provided in this section to enhance the description of the application notes defined in the family. As

with application notes the information presented is explanatory and intended to assist, it is not mandatory.

- The *rationale* contains the specifics of the rationale that refines 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, evaluator notes and/or documentation notes as described in section A.1.2 of this annex. This refinement can be used to explain dependencies (e.g. shared information, or shared operation).

15 This section is not mandatory and should appear only if appropriate.

A.1.3.3 Permitted operations

16 Components may be tailored through use of permitted operations before being incorporated into a PP, an ST or a functional package, based on the particular environment of use and security policies being addressed. The possible operations are defined in the CC Part 2 document and elaborated on in this annex. Not all operations are permitted on all functional components. The iteration and refinement can be applied to any component. For the selection and the assignment each component shall contain a description of the allowed operations, the circumstances under which the operation can be applied to the component, and the results of the application of this operation.

17 This section is not mandatory and should appear only if appropriate.

A.2 Dependency Table

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

D R A F T

	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.3	FCS_CKM.7	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_ETC.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	FDP_ATD.1	FDP_AUU.1	FIA_UID.1	FMT_MOF.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMR.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																										-				
FAU_STG.1					x																										-				
FAU_STG.2					x		x																								-				
FAU_STG.3					-			x																							-				
FAU_STG.4					x		x																								-				
FCO_NRO.1																						x													
FCO_NRO.2																						x													
FCO_NRR.1																						x													
FCO_NRR.2																						x													
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	-	x	-	-	-	o	-	-	-				-		-	x	-		-							
FCS_CKM.5	-								o	-	x	-	-	-	-	-	-	o				-		-	x	-		-							
FCS_CKM.6	-								o	-	x	-	-	-	-	-	-	o				-		-	x	-		-							

Table A.1 - Dependencies for functional components

D R A F T

	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.3	FCS_CKM.7	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_ETC.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	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	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.7	-								o	-	-	-	-	-	-	-	o					-		-	x	-		-								
FCS_CKM.8	-								o	-	-	-	-	-	-	-	o					-		-	x	-		-								
FCS_COP.1	-								o	-	x	-	-	-	-	-	o					-		-	x	-		-								
FCS_COP.2	-								o	-	x	-	-	-	-	-	o					-		-	x	-		-								
FDP_ACC.1	-												-	x		-	-					-		-		-		-								
FDP_ACC.2	-												-	x		-	-					-		-		-		-								
FDP_ACF.1	-												x	-		-	-					-		-		x		-								
FDP_ACF.2	-												x	-		-	-					-		-		x		-								
FDP_ACF.3	-												x	-		-	-					-		-		x		-								
FDP_ACF.4	-												x	-		-	-					-		-		-		-								
FDP_DAU.1																																				
FDP_DAU.2																																				
FDP_ETC.1	-												o	-		o	-					-		-		-		-								
FDP_ETC.2	-												o	-		o	-					-		-		-		-					x		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_IFF.7	-												-	-		x	-					-		-		x		-								
FDP_IFF.8	-												-	-		x	-					-		-		x		-								
FDP_ITC.1	-												o	-		o	-					-		-		x		-								
FDP_ITC.2	-												o	-		o	-					-		-		-		-					x		o	o

Table A.1 - Dependencies for functional components

D R A F T

	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.3	FCS_CKM.7	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_ETC.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	FDP_RIP.1	FIA_ATD.1	FIA_AUA.1	FIA_UID.1	FMT_MOF.1	FMT_MSA.1	FMT_MSA.2	FMT_MSA.3	FMT_MTD.1	FMT_SMR.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.1	-												o	-		o	-							-		-		-		-							o	o
FDP_ITT.2	-												o	-		o	-							-		-		-		-							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	
FDP_UIT.3	-												o	-		o	-							-		-		-		-							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																																						
FIA_UAU.7																							x	-														
FIA_UID.1																																						

Table A.1 - Dependencies for functional components

D R A F T

	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.3	FCS_CKM.7	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_ETC.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	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_REV.1	FMT_SAE.1	FMT_SMR.1	FMT_SMR.2	FMT_SMR.3	FPR_ANO.1	FPR_ANO.2	FPR_PSE.1	FPR_PSE.2	FPR_PSE.3	FPR_UNL.1	FPR_UNO.1	FPR_UNO.2	FPT_AMT.1	FPT_FLS.1	FPT_ITA.1	FPT_ITC.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		-	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																																															
FPR_UNO.2																																															
FPT_AMT.1																																															
FPT_FLS.1	x																																														
FPT_ITA.1																																															
FPT_ITC.1																																															

Table A.1 - Dependencies for functional components

D R A F T

	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.3	FCS_CKM.7	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_ETC.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	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	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_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	-						x		
FPT_RCV.2	x	x																				-						x	-						x		
FPT_RCV.3	x	x																				-						x	-						x		
FPT_RCV.4	x																																				
FPT_RPL.1																																					
FPT_RVM.1																																					
FPT_SEP.1																																					
FPT_SEP.2																																					
FPT_SEP.3																																					
FPT_SSP.1																															x						
FPT_SSP.2																															x						
FPT_STM.1																																					
FPT_TDC.1																																					
FPT_TRC.1																															x						
FPT_TST.1																													x								
FRU_FLT.1	-																													x							
FRU_FLT.2	-																													x							
FRU_PRS.1																																					

Table A.1 - Dependencies for functional components

D R A F T

	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.3	FCS_CKM.7	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FDP_ETC.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITT.1	FDP_ITT.2	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	FPT_AMT.1	FPT_FLS.1	FPT_ITT.1	FPT_STM.1	FPT_TDC.1	FPT_TST.1	FTP_ITC.1	FTP_TRP.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																																				
FTP_TRP.1																																				

Table A.1 - Dependencies for functional components

Annex B

Assurance classes, families, and components

19 This chapter provides the detailed requirements, presented in alphabetical order, of each of the assurance components, grouped by class and family.

Class FAU

Security Audit

20 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 the use of access rights by all users, 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. Audit data should be available in a useful format, that presents audit data in a human-readable format and/or delivers it to authorised users or processes acting on their behalf.

21 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).

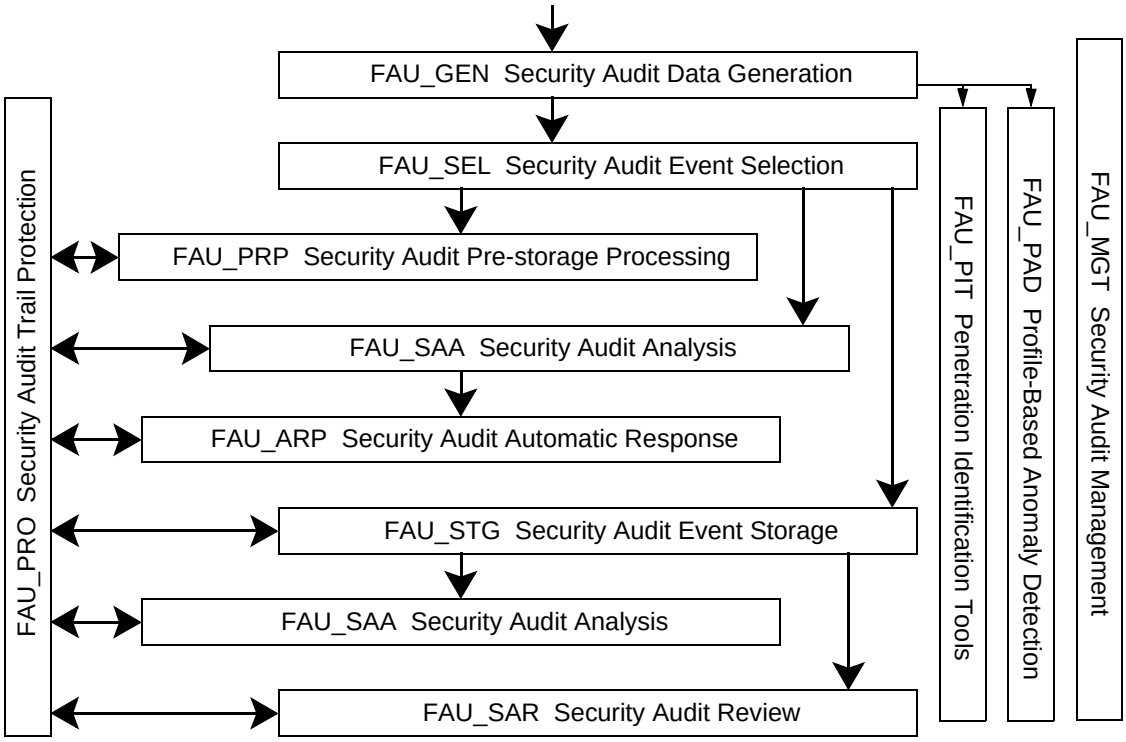

Figure B.1 - Audit requirements construction rules

D R A F T

Audit requirements in a distributed environment:

22 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 construed, 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.

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

24 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.

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

26 Figure B.2 shows the decomposition of this class into its constituent components.

Component Catalogue

D R A F T

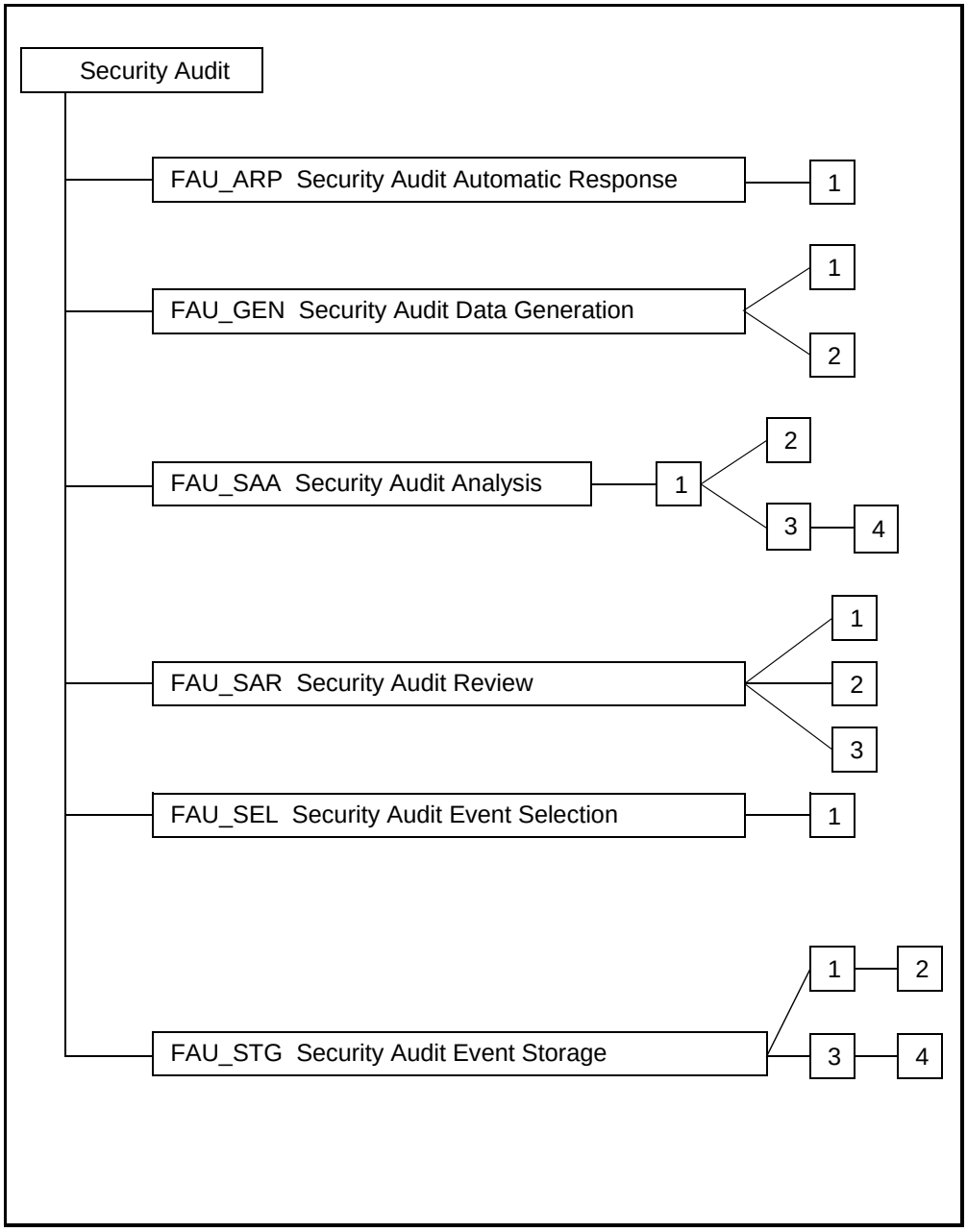

Figure B.2 - Security Audit class decomposition

D R A F T

FAU_ARP Security Audit Automatic Response

27 The Security Audit Automatic Response family describes requirements for the handling of audit events. The possibilities 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

28 An audit event appears to be an “imminent security violation” if so indicated by the FAU_SAA components.

FAU_ARP.1 Security Alarms

User Application Notes

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

Operations

Assignment:

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

D R A F T

FAU_GEN Security Audit Data Generation

31 The Security Audit Data Generation family includes requirements to specify the audit events that should be generated by the TSF for some activity.

32 This family is presented in a manner which 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.

33 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:

34 “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; and

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.”

35 If multiple auditable events are specified on the same level as audit, to satisfy the level of audit all those auditable events should be auditable.

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

37 The PP/ST author might also decide to include other auditable events above and beyond the auditable events indicated by the audit level.

D R A F T

Application Notes

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

39 The following are examples of the types of the events which 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 administrator (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

40 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.

41 FAU_GEN.1 is for use 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.

42 The information requested by this component to be recorded in each record is relevant for a general purpose operating system, but for some specific applications, a refinement of this information could be necessary to avoid requesting non available data.

Evaluator application notes

43 This component addresses the possible existence of audit functionality in the potential absence of individual user identities.

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

Operations

Selection:

45 **For FAU_GEN.1.1b, the PP/ST author should select the [*not specified, minimum, basic, detailed*] level of auditable events called out in the audit section of other functional components included in the PP/ST. If 'not specified' is selected the PP/ST author should fill in all desired auditable events in FAU_GEN.1.1c.**

Assignment:

46 **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 API.**

Selection:

47 **For FAU_GEN.1.2a, the PP/ST author should select the [*success, failure*] of auditable events to be audited. This selection should be consistent with the level of auditable events.**

Assignment:

48 **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 Generation

User Application Notes

49 This component addresses the requirement in the TSP 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.

50 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 actions to himself and not be identified by other persons (e.g. a site with job offers). In those cases the objectives for audit and privacy might contradict each other. Therefore if this requirement is selected, user pseudonimity might be selected that requires that the interpretation of the user identity is the pseudonym of the user. Requirements on determining the real identity of the user based on its pseudonym will need to be specified in the privacy class.

D R A F T

FAU_SAA Security Audit Analysis

51 This family defines requirements for automated means which 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.

52 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

53 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 Imminent Violation Analysis

User Application Notes

54 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:

55 **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:

56 **In FAU_SAA.1.2.b, the PP/ST author should assign [*any other rules*] which the TSF shall use in its analysis of the audit trail. Those rules could include specific requirements to express the need 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

57 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*.

58 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.

59 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.

60 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.

61 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 administrator, through addition, modification or deletion of events or rules.

62 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.

63 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.

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

D R A F T

65 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:

66 **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.**

Assignment:

67 **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 based on the type of anomalous activity observed.**

FAU_SAA.3 Simple Attack Heuristics

User Application Notes

68 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.

69 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.

70 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.

71 Administrative notification should be provided such that the authorised administrator understands the significance of the event and what possible responses might be appropriate.

72 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).

73 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:

74 **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.**

Assignment:

75 **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.**

76

FAU_SAA.4 Complex Attack Heuristics

User Application Notes

77 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.

78 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.

79 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.

80 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.

D R A F T

81 Administrative notification should be provided such that the authorised administrator understands the significance of the event and what possible responses might be appropriate.

82 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.

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

84 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:

85 **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.**

Assignment:

86 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.

Assignment:

87 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.

D R A F T

FAU_SAR Security Audit Review

88 The Security Audit Review family defines requirements related to review of the audit information.

89 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

90 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

91 This component is used to specify that users and or authorised administrators can read the audit records. These audit records will be provided in a manner appropriate to the user. The difference is between machine users and human users.

92 The information of the audit records that can be viewed can be specified.

Operations

Selection:

93 **In FAU_SAR.1.1 the PP/ST author must specify whether the requirement applies to the authorised administrator, and/or authorised users.**

Assignment:

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

D R A F T

FAU_SAR.2 Restricted Audit Review

User Application Notes

95 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

96 This component is used to specify that it should be possible to perform selection of the audit data to be reviewed. If based on a single criterion, this component could be used more than one time, to define different single criteria that could be used to perform the analysis. If based on multiple criteria, those criteria should be related together with logical (i.e. and / or) relations, and the tools should provide the ability to manipulate audit data (e.g. sort, filter).

Operations

Selection:

97 **For FAU_SAR.3.1 the PP/ST author should select whether the action [searching (through the set of audit records), sorting or ordering] are performed by the TSF.**

Assignment:

98 **For FAU_SAR.3.1, the PP/ST author should assign [*multiple criteria 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.**

D R A F T

FAU_SEL Security Audit Event Selection

99 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

100 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

101 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.

102 The existence of individual user identities is not assumed for this component. This would allow for TOEs such as routers that may not support the notion of users.

103 For a distributed environment, the Host identity could be used as a selection criteria for events to be audited.

104 Users not identified in the requirement are explicitly excluded from being able to perform the operations indicated.

Operations

Selection:

105 **For FAU_SEL.1.1a, the PP/ST author should select from [*Object identity, User identity, Subject identity, Host identity, Event Type*] the security attributes that audit selectivity is based upon.**

Assignment:

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

Selection:

107 **In FAU_SEL.1.2 the PP/ST author must specify whether the requirement applies to the authorised administrator, and/or authorised users.**

Selection:

108 **In FAU_SEL.1.2 the PP/ST author should specify whether the identified users or authorised administrators can only display, can only select auditable events or can do both.**

D R A F T

FAU_STG Security Audit Event Storage

109 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.

Application Notes

110 The permanence of the audit trail should be considered also in terms of duration of validity of the audit information.

FAU_STG.1 Permanent Audit Trail Storage

User Application Notes

111 In a distributed environment, as the location of the audit trail should be 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.

FAU_STG.2 Guarantees of Audit Data Availability

User Application Notes

112 This component allows the PP/ST author to specify to which metrics the audit trail should conform.

113 In a distributed environment, as the location of the audit trail should be 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:

114 **In FAU_STG.2.2, the PP/ST author should specify the condition [*audit storage exhaustion, failure, attack*] under which the TSF shall control audit data loss.**

Assignment:

115 **In FAU_STG.2.2, the PP/ST author should specify the metric that the TSF must ensure with respect to the audit trail. This metric could be based on time, and/or size. An example of the metric could be: “100,000” indicating that a 100,000 audit records can be stored.**

D R A F T

FAU_STG.3 Action in Case of Possible Audit Data Loss

User Application Notes

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

Operations

Assignment:

117 **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 administrator this value is the default value. The PP/ST author might choose to let the authorised administrator define this limit. In that case the assignment can be for example "an authorised administrator set limit".**

Assignment:

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

FAU_STG.4 Prevention of Audit Data Loss

User Application Notes

119 This component specifies what happens to 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 administrator can continue to generate auditable events (actions). The reason is that otherwise the authorised administrator 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. The authorised administrator is given the opportunity of selecting whether the TOE should continue to work or to lock the TOE if the audit trail is full.

Operations

Selection:

120 **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 of happening when the TSF can no longer store audit records.**

Assignment:

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

121

D R A F T

Class FCO

Communication

122 This class describes requirements specifically of interest for TOEs which are used for the transport of information. The currently identified families deal with non-repudiation.

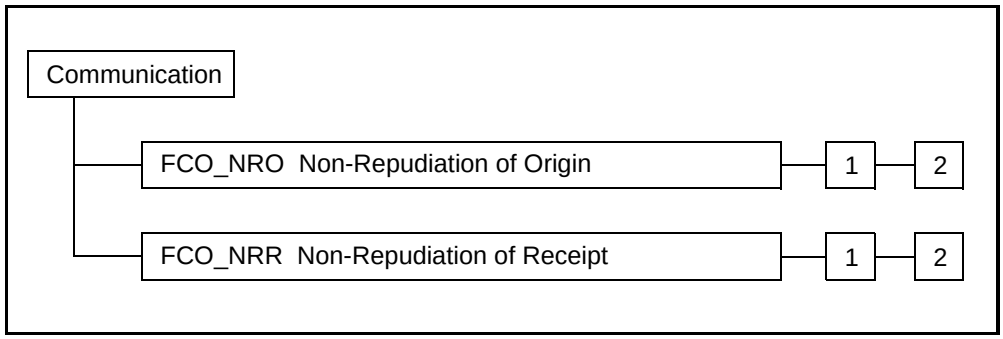

Figure B.3 - Communication class decomposition

123 Figure B.3 shows the decomposition of this class into its constituent components.

124 In this class the concept of “information” is being used. This information should be interpreted as the object being communicated. Therefore this information could contain an electronic mail message, a file, or a set of predefined attribute types.

125 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 R A F T

FCO_NRO Non-Repudiation of Origin

126 Non-repudiation of origin defines requirements to provide evidence to users/ subjects about, for example, 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.

User notes

127 The Non-repudiation of Origin requirements provide evidence to other subjects about the attributes of the originator of information. This evidence should not be forgeable.

128 If a part of the protected part of the information or of the associated attributes is altered in any way, validation of the evidence of origin may fail. Therefore a PP/ST author should consider including integrity requirements such as FDP_UIT.1 Data Exchange Integrity in the PP/ST.

129 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 like an arbiter.

130 The PP/ST author must specify the conditions which must be met to be able to verify the validity of the evidence. Such a condition could be a time interval, related to reserved memory, or the availability of third parties. These conditions therefore allow the tailoring of the non-repudiation to legal requirements such as being able to provide evidence for several years.

131 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.

132 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. The requirements can, in such instances, be customised to provide a timestamp indication (time of origin).

D R A F T

FCO_NRO.1 Selective Proof of Origin

Operations

Assignment:

133 **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:

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

Assignment:

135 **In FCO_NRO.1.1 the PP/ST author, dependent on the selection, should specify the *third parties* that can request evidence of origin.**

Assignment:

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

137 **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 the information.**

Selection:

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

Assignment:

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

140 **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.**

D R A F T

FCO_NRO.2 Enforced Proof of Origin

Operations

Assignment:

141 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.

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

143 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 the information.

Selection:

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

Assignment:

145 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.

146 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 R A F T

FCO_NRR Non-Repudiation of Receipt

147 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.

User notes

148 The Non-repudiation of Receipt requirements provide a requirement to provide evidence to other subjects about the attributes of the recipient of the information. This evidence should not be forgeable.

149 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.

150 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 like an arbiter.

151 The PP/ST author must specify the conditions which must be met to be able to verify the validity of the evidence. Such a condition could be a time interval, related to reserved memory, or the availability of third parties. These conditions therefore allow the tailoring of the non-repudiation to legal requirements such as being able to provide evidence for several years.

152 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.

153 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. The requirements can, in such instances, be customised to provide a timestamp indication (time of receipt).

D R A F T

FCO_NRR.1 Selective Proof of Receipt

Operations

Assignment:

154 **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:

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

Assignment:

156 **In FCO_NRR.1.1 the PP/ST author, dependent on the selection, should specify the *third parties* that can request evidence of receipt.**

Assignment:

157 **In FCO_NRR.1.2 the PP/ST author should specify the *list of attributes* which shall be linked to the information, for example recipient identity, time of receipt, and location of receipt.**

158 **In FCO_NRR.1.2 the PP/ST author should specify the *list of information fields* with the fields within the information over which the attributes provide evidence of receipt, such as the body of the information.**

Selection:

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

Assignment:

160 **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.**

161 **In FCO_NRR.1.3 the PP/ST author should specify 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 R A F T

FCO_NRR.2 Enforced Proof of Receipt

Operations

Assignment:

162 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.

163 In FCO_NRR.2.2 the PP/ST author should specify the *list of attributes* which shall be linked to the information, for example recipient identity, time of receipt, and location of receipt.

164 In FCO_NRR.2.2 the PP/ST author should specify the *list of information fields* with the fields within the information over which the attributes provide evidence of receipt, such as the body of the information.

Selection:

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

Assignment:

166 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.

167 In FCO_NRR.2.3 the PP/ST author should specify 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 R A F T

Class FCS

Cryptographic Support

168 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.

169 The FCS class is organised into 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.

170 Figure B.4 shows the decomposition of this class into its constituent components.

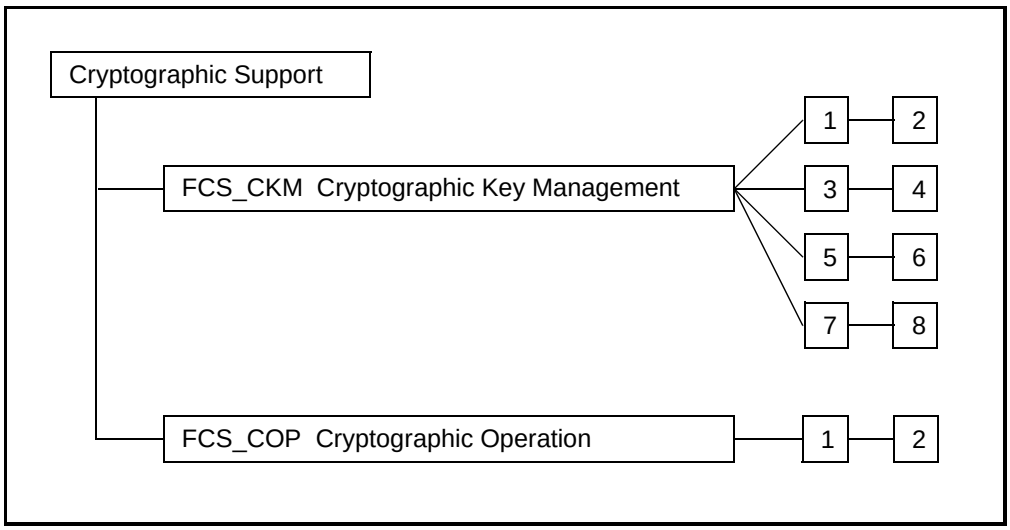

Figure B.4 - Cryptographic Support class decomposition

Construction Rules

171 The construction rules for cryptographic support requirements are shown diagramatically in Figure B.5. It should be noted that components from the FDP and FMT classes may also need to be used.

172 When building a PP, ST or package using components from the FCS class, these construction rules will provide guidance on where to look and what to select from the class.

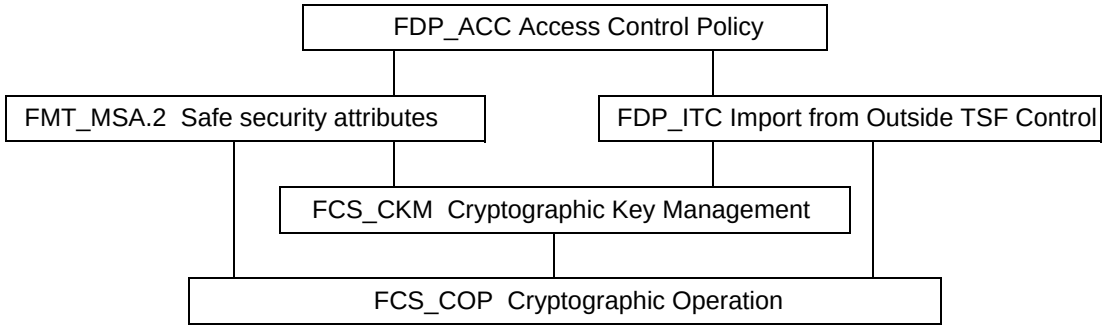

Figure B.5 - Cryptographic Support construction rules

173 Cryptographic keys should be stored in and protected by the TOE in accordance with an access control policy using a component from the FDP_ACC family.

174 A component from the FMT_MSA family should be used to define cryptographic key attributes used by the TOE.

175 For each cryptographic key generation method implemented by the TOE, if any, the PP/ST author should select either the FCS_CKM.1 or FCS_CKM.2 component.

176 If a cryptographic key is generated outside of the TOE, the FDP_ITC family should be used to specify how the cryptographic key is imported into the TOE.

177 For each cryptographic key distribution method implemented by the TOE, if any, the PP/ST author should select either the FCS_CKM.3 or FCS_CKM.4 component.

178 For each cryptographic key access method implemented by the TOE, if any, the PP/ ST author should select either the FCS_CKM.5 or FCS_CKM.6 component.

179 For each cryptographic key destruction method implemented by the TOE, if any, the PP/ST author should select either the FCS_CKM.7 or FCS_CKM.8 component.

180 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 either the FCS_COP.1 or FCS_COP.2 component.

D R A F T

FCS_CKM Cryptographic Key Management

User notes

181 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.

182 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).

183 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.

184 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).

FCS_CKM.1 Cryptographic Key Generation

User Application Notes

185 This component requires the cryptographic key sizes and method used to generate cryptographic keys to be specified. 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.

186 Typically random numbers are used to generate cryptographic keys. If this is the case, then this component 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.2 Standards-Based Cryptographic Key Generation

User Application Notes

187 This component requires the cryptographic key sizes and method used to generate cryptographic keys to be specified in accordance with an assigned standard. It should be used to specify the cryptographic key sizes and the standards-based 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.

188 Typically random numbers are used to generate cryptographic keys. If this is the case, then this component 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.

Operations

Assignment:

189 **In FCS_CKM.2.1, the PP/ST author should specify the assigned standard which documents the method used to generate cryptographic keys. The assigned standard may comprise one or more actual standards publications.**

FCS_CKM.3 Cryptographic Key Distribution

User Application Notes

190 This component requires the method used to distribute cryptographic keys to be specified.

FCS_CKM.4 Standards-Based Cryptographic Key Distribution

User Application Notes

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

Operations

Assignment:

192 **In FCS_CKM.4.1, the PP/ST author should specify the assigned standard which documents the method used to distribute cryptographic keys. The assigned standard may comprise one or more actual standards publications.**

D R A F T

FCS_CKM.5 Cryptographic Key Access

User Application Notes

193 This component requires the method used to access cryptographic keys to be specified.

Operations

Assignment:

194 **In FCS_CKM.5.1, 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.**

FCS_CKM.6 Standards-Based Cryptographic Key Access

User Application Notes

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

Operations

Assignment:

196 In FCS_CKM.6.1, 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.

197 **In FCS_CKM.6.1, the PP/ST author should specify the assigned standard which documents the method used to access cryptographic keys. The assigned standard may comprise one or more actual standards publications.**

FCS_CKM.7 Cryptographic Key Destruction

User Application Notes

198 This component requires the method used to destroy cryptographic keys to be specified.

D R A F T

FCS_CKM.8 Standards-Based Cryptographic Key Destruction

User Application Notes

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

Operations

Assignment:

200 **In FCS_CKM.8.1, the PP/ST author should specify the assigned standard which documents the method used to destroy cryptographic keys. The assigned standard may comprise one or more actual standards publications.**

D R A F T

FCS_COP Cryptographic Operation

User notes

201 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.

202 If FAU_GEN Security Audit Data Generation is included in the PP/ST then, in the context of the 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

203 This component requires the cryptographic algorithm and key size used to perform specified cryptographic operation(s) to be specified.

Operations

Assignment:

204 **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.**

D R A F T

FCS_COP.2 Standards-Based Cryptographic Operation

User Application Notes

205 This component requires the cryptographic algorithm and key size used to perform specified cryptographic operation(s) to be specified in accordance with an assigned standard.

Operations

Assignment:

206 In FCS_COP.2.1, 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.

207 **In FCS_COP.2.1, the PP/ST author should specify the assigned standard which documents how the identified cryptographic operation(s) are performed. The assigned standard may comprise one or more actual standards publications.**

D R A F T

Class FDP

User Data Protection

208 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.

209 The class does not contain explicit requirements for TCSEC Mandatory Access Controls or Discretionary Access Controls; however, such requirements may be constructed using components from this class.

210 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 policies in the PP/ST.

211 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”.

212 The access control policy is concerned with the operations on the object. Information flow policies are concerned with the content of the object. Therefore, information flow policies are considered more in terms of flow of the information rather than a specific operation on an object.

213 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 which are not captured in the definitions here.

214 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). However, this concept is not supported by existing practice, and research to date has not demonstrated practical general-purpose solutions, particularly in the context of a TOE supporting subjects that are not trusted to enforce that policy. Such concepts could, of course, be the subject of extensions to the FDP components.

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

216 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 (although FDP_ACC.1 does not mandate this) and that there are no conflicts arising from implementing the multiple SFPs.

217 Figures B.6 and B.7 show the decomposition of this class into its constituent components.

D R A F T

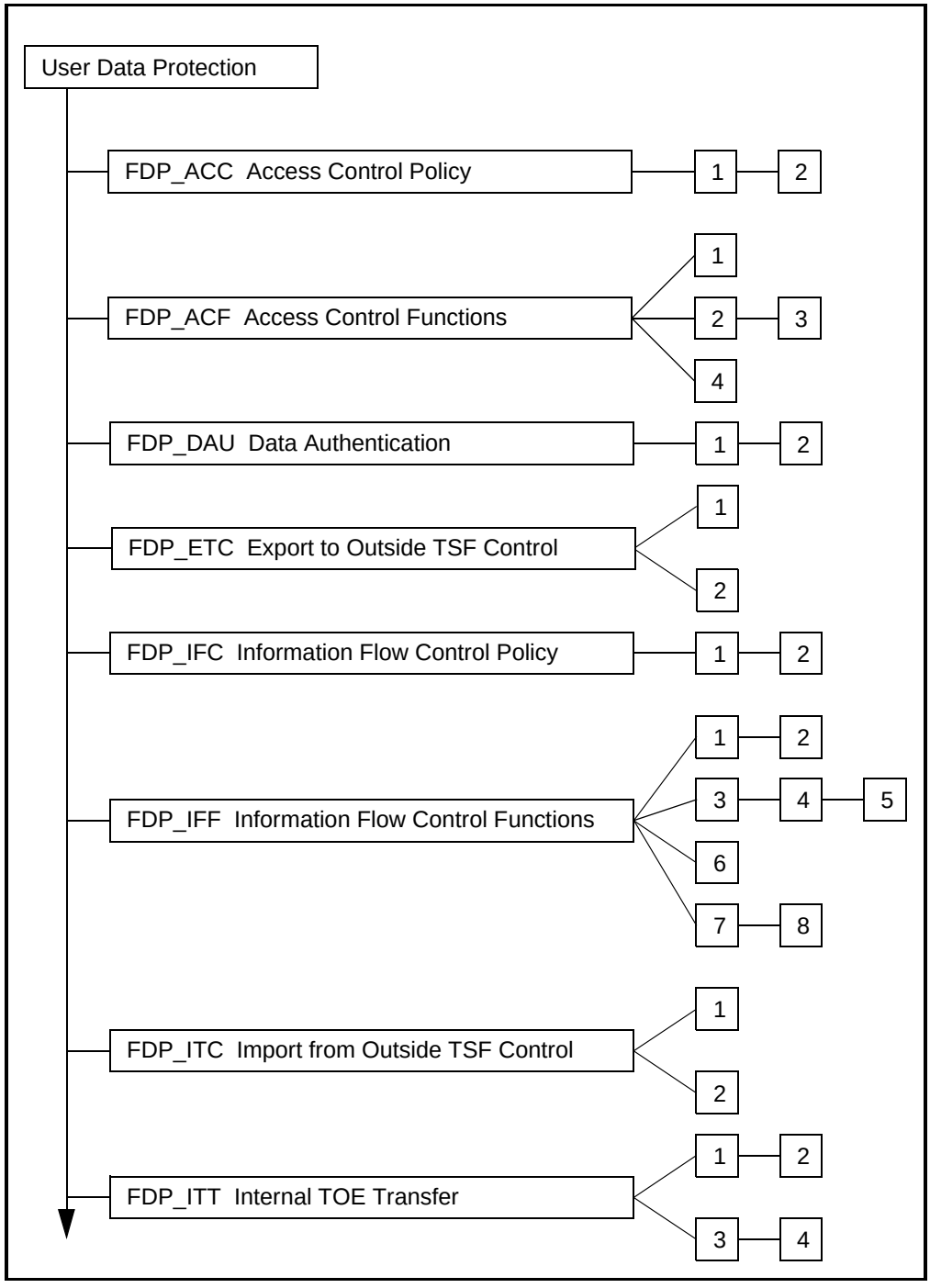

Figure B.6 - User Data Protection class decomposition

D R A F T

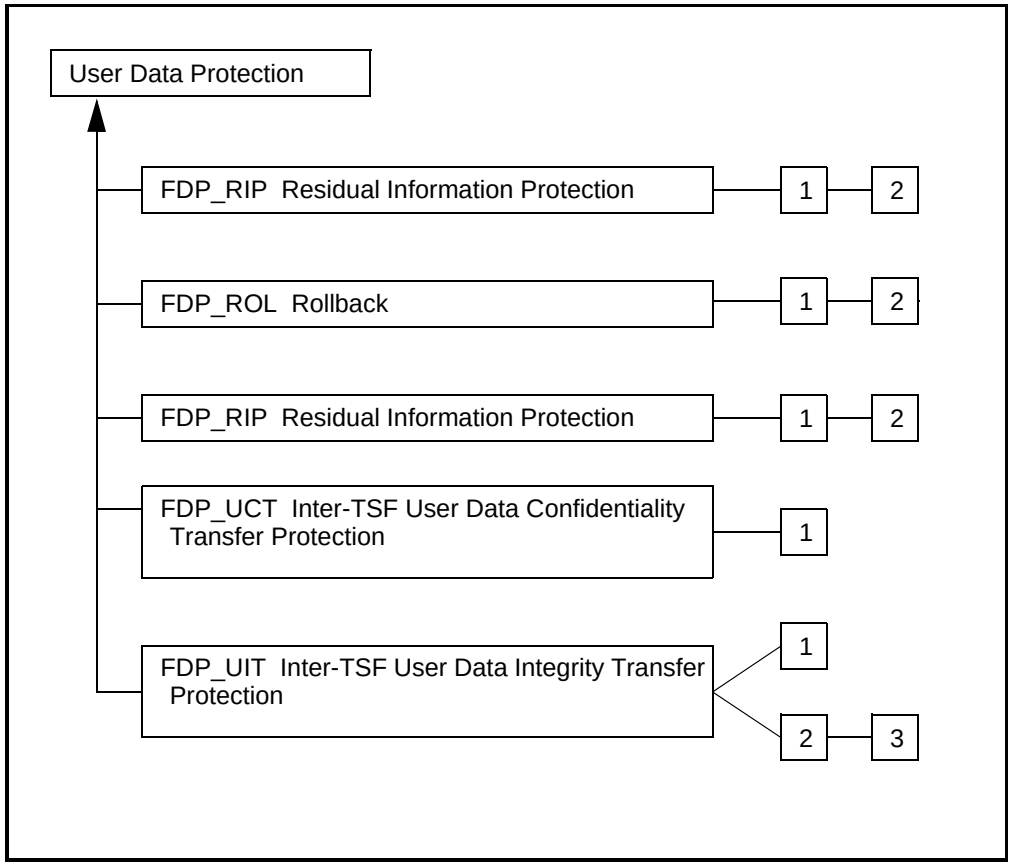

Figure B.7 - User Data Protection class decomposition (cont.)

Construction Rules

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

219 The requirements in the FDP class are defined in terms of a security function (abbreviated SF) which 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.

220 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.

221 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 these components will define the subjects, objects and operations covered by this security function.

222 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 respective policy families. Perform the operations to 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 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 the appropriate components from the Class FMT for initial values for new objects and subjects.

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

g) Identify any applicable object reuse requirements from the FDP_RIP family.

h) Identify any applicable import or export components from the FDP_ITC and FDP_ETC families.

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

j) Identify the 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.

D R A F T

FDP_ACC Access Control Policy

223 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

224 The components in this family are capable of defining the access control SFPs to be enforced by the traditional Discretionary Access Control mechanisms. It further defines the subjects, objects and operations that are covered by identified access control SFPs. The functionality that fulfills an access control SFP will be defined by other families such as FDP_ACF and FDP_RIP. The access control SFPs defined here in FDP_ACC are meant to be used throughout the remainder of the Part 2 functional components that have an operation that calls for an assignment or selection of an “access control SFP.”

225 The access control SFP covers a set of triplets subject, object, and operations. Therefore a subject can be covered by multiple SFPs but only with respect to a different operation or a different object. Of course the same applies to objects and operations.

226 This family would provide 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.

227 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 Security Management.

228 There are no audit requirements in FDP_ACC since 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.

229 This family can be applied multiple times in a PP/ST to different subsets of operations and objects. This will accommodate TOEs which 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 which the TOE 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.

D R A F T

FDP_ACC.1 Subset Access Control

User Application Notes

230 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.

231 This component simply 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:

232 **In FDP_ACC.1.1, the PP/ST author should specify a unique named [*access control SFP*] to be enforced by the TSF.**

233 **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

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

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

Operations

Assignment:

236 In FDP_ACC.2.1, the PP/ST author should specify a unique named [*access control SFP*] to be enforced by the TSF.

237 **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.**

D R A F T

FDP_ACF Access Control Functions

238 This family describes specific functions that can implement the rules for access control SFPs. This family is dependent on the definition of an access control SFP.

User notes

239 This family provides a PP/ST author the capability to describe the rules for access control. Furthermore, the PP/ST author can explicitly require that the access control attributes are fixed. 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.

240 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 DAC mechanisms, can be represented with the existing components, by careful drafting of the access control rules.

241 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; and
- Owner-controlled access control attributes

FDP_ACF.1 Security Attribute Based Access Control

User Application Notes

242 This component provides requirements for a mechanism that mediates access control based on a single security attribute 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. ACLs). This component allows the PP/ST author to specify the attribute that will be used for the access control mediation. Furthermore, this component allows access control rules, using this attribute, to be specified.

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

244 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.

245 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.

246 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.

D R A F T

247 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.

Operations

Assignment:

248 **In FDP_ACF.1.1, the PP/ST author should specify an [*access control SFP*] name which the TSF is to enforce.**

249 **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. The security attributes may be things like 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.**

250 **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 and can require general access control functions (e.g. typical permission bits) or granular access control functions (e.g. ACLs).**

FDP_ACF.2 Access Authorisation

User Application Notes

251 This component provides requirements for the access control security functions to be able to explicitly authorise access to an object based upon security attributes.

Operations

Assignment:

252 **In FDP_ACF.2.1, the PP/ST author should specify the [*access control SFP*] name related to an access control SF which will include the special rule that requires access to be explicitly authorised.**

253 **In FDP_ACF.2.1, the PP/ST author should specify the [*value of security attributes of subjects and objects*] that will be used to explicitly authorise access. An example is a privilege vector associated with a subject that always grants access to objects covered by the access control SFP that has been specified.**

D R A F T

FDP_ACF.3 Access Authorisation and Denial

User Application Notes

254 This component provides requirements for the access control security functions to be able to explicitly authorise and deny access to an object based upon security attributes

Operations

Assignment:

255 In FDP_ACF.3.1, the PP/ST author should specify the [*access control SFP*] name related to an access control SF which will include the special rule that requires access to be explicitly authorised.

256 In FDP_ACF.3.1, the PP/ST author should specify the [*value of security attributes of subjects and objects*] that will be used to explicitly authorise access. An example is a privilege vector associated with a subject that always grants access to objects covered by the access control SFP that has been specified.

257 **In FDP_ACF.3.2, the PP/ST author should specify the [*access control SFP*] name related to an access control SF which will include the special rule that requires access to be explicitly denied.**

258 **In FDP_ACF.3.2, the PP/ST author should specify the [*value of security attributes of subjects and objects*] that will be used to explicitly deny access. An example is a privilege vector associated with a subject that always denies access to objects covered by the access control SFP that has been specified.**

FDP_ACF.4 Fixed Access Control

User Application Notes

259 This component ensures that the access control security attribute of a given SFP cannot be modified. Therefore subjects rights to the file cannot be changed. And in effect a static fixed access control policy is created

260 For example, the “message of the day” function typically provided by many multi-user TOEs is covered by a fixed access control policy. The access control for the read and write operations can not be changed as they are built into the security function which provides only read access for users and only read/write access for administrators.

261 It is remarked that the user attributes and the object attributes could both control the access control between subjects and objects. It depends, for example, on whether an ACL (Object attributes) or Capability Lists (Subject attributes) which set of attributes should be fixed.

262 Since this component deals specifically with a fixed set of security attributes, audit with respect to this component is unnecessary. Other Access Control Functions would make use of those security attributes to enforce the SFP, and therefore, auditing would be covered by those other components.

Operations

Assignment:

263 **In FDP_ACF.4.1, the PP/ST author should specify the [*access control SFP*] name which the TSF is to enforce for a fixed set of security attributes.**

D R A F T

FDP_DAU Data Authentication

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

User notes

265 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

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

Operations

Assignment:

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

268 **In FDP_DAU.1.2, the PP/ST author should specify the [assignment: *list of subjects*] that will have the ability to verify data authentication evidence for the objects identified in the previous element.**

FDP_DAU.2 Data Authentication with Identity of Guarantor

User Application Notes

269 This component additionally requires the ability to verify the identity of the entity which provided the guarantee of authenticity (e.g., a trusted third party).

Operations

Assignment:

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

271 In FDP_DAU.2.2, the PP/ST author should specify the [assignment: *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 subject that created the data authentication evidence**.

D R A F T

FDP_ETC Export to Outside TSF Control

272 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.

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

User notes

274 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.

275 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.

FDP_ETC.1 Export of User Data Without Security Attributes

User Application Notes

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

Operations

Assignment:

277 **In FDP_ETC.1.1, the PP/ST author should specify the [*access control SFP* and/or *information flow control SFP*] 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

278 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. A trusted channel is required to assure that the attributes are correctly received at the other Trusted IT Product.

Operations

Assignment:

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

280 **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.**

D R A F T

FDP_IFC Information Flow Control Policy

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

282 Examples of security policies that might satisfy this objective are:

- Bell and La Padula Security model [B&L];
- Biba Integrity model [Biba].

User notes

283 The components in this family are capable of implementing the traditional Mandatory Access Control mechanisms. However, they 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 strengths of the information flow control components also permit different degrees of exception to the policy.

284 Each SFP covers a set of triplets: subject, object, and operations. In the second component (FDP_IFC.2 Complete Information Flow Control), each information flow control SFP will cover all possible operations between a subject and an object covered by that SFP. Furthermore, each object will need to be covered by a SFP. Therefore for each action on an object there will be a set of rules that define whether this action is allowed. If there are multiple SFPs that are applicable for a given action, all involved SFPs must grant access for the action.

285 An information flow control SFP covers a well-defined set of operations. The SFPs coverage may be “complete” with respect to some object, or it may address only some of the operations that affect the object. A critical aspect of an information flow control SFP is that it may be specified; that is, it is based upon some changeable attribute that determines the flow of the information.

286 Information flow control SFPs cover access to the information which differs from access control SFPs which cover access to the objects themselves. Security attributes that are bound to information will flow with the information from container to container.

287 Objects and operations can be expressed at multiple levels. In the case of a ST, the objects and operations might be specified at a system-specific level: files and open. For a PP, the objects and operations might be expressed as types: named objects, data repositories, observe accesses, etc.

288 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 which contain multiple policies, each addressing a particular set of objects, subjects, and operations.

D R A F T

FDP_IFC.1 Subset Information Flow Control

User Application Notes

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

Operations

Assignment:

290 **In FDP_IFC.1.1, the PP/ST author should specify the [*information flow control SFPs*] to be enforced by the TSF.**

291 **In FDP_IFC.1.1, the PP/ST author should specify the [*list of subjects, objects and operations among subjects and objects covered by the SFP*].**

FDP_IFC.2 Complete Information Flow Control

User Application Notes

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

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

Operations

Assignment:

294 In FDP_IFC.2.1, the PP/ST author should specify the [*information flow control SFPs*] to be enforced by the TSF.

295 **In FDP_IFC.2.1, the PP/ST author should specify the [*list of subjects and objects*] that will be covered by the SFP. All operations among those subjects and objects will be covered by the SFP.**

D R A F T

FDP_IFF Information Flow Control Functions

296 This component specifies the requirements on function with respect to the information flow control SFPs. 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

297 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 of the information flows implicit in controlled operations.

298 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 state of the art.

299 The flexibility of these components allow the definition of a privilege policy to allow 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

300 This component requires security attributes on containers of information, and on active recipients of information. It specifies the key rules that are enforced, and describes how security attributes are derived. 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], but these security attributes do not form a hierarchy.

301 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.

302 Upon creation of a subject, the FIA_USB specifies that the object (image) and the user attributes determine the subject security attributes. If the SFP has additional rules on the management of the subject security attributes those can be specified under the additional information flow control SFP rules. If there are specific rules for the object security attributes those can be specified under the additional information flow control SFP rules.

D R A F T

Operations

Assignment:

303 **In FDP_IFF.1.1, the PP/ST author should specify the [*information flow control SFPs*] enforced by the TSF.**

304 **In FDP_IFF.1.1 the PP/ST author should specify [*the minimum number and type of security attributes*] which the mechanism will enforce. The type of security attributes can be things like: subject sensitivity level, subject clearance level, object sensitivity level, etc. The minimum number of each type of security attribute should be sufficient to support the environmental needs.**

305 **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 object security attributes*] that the TSF will enforce.**

306 **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".**

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

FDP_IFF.2 Hierarchical Security Attributes

User Application Notes

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

309 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.

310 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.

311 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 instantiate 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:

312 In FDP_IFF.2.1, the PP/ST author should specify the [*information flow control SFPs*] enforced by the TSF.

313 In FDP_IFF.2.1 the PP/ST author should specify [*the minimum number and type of security attributes*] which the mechanism will enforce. The type of security attributes can be things like: subject sensitivity level, subject clearance level, object sensitivity level, etc. The minimum number of each type of security attribute should be sufficient to support the environmental needs.

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

315 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”.

316 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”.

FDP_IFF.3 Limited Illicit Information Flows

User Application Notes

317 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.

318 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:

319 **In FDP_IFF.3.1 the PP/ST author should specify the [*information flow control SFPs*] enforced by the TSF.**

320 **In FDP_IFF.3.1 the PP/ST author should specify the [*types of illicit information flows*] which are subject to a maximum capacity limitation.**

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

D R A F T

FDP_IFF.4 Partial Elimination of Illicit Information Flows

User Application Notes

322 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:

323 In FDP_IFF.4.1 the PP/ST author should specify the [*information flow control SFPs*] enforced by the TSF.

324 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.

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

326 **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

327 This component should be used when all the SFPs that require control of illicit information flows require elimination of all illicit information flows.

Operations

Assignment:

328 **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.**

FDP_IFF.6 Illicit Information Flow Monitoring

User Application Notes

329 This component should be used when it is desired that the TSF provide the ability to audit the use of illicit information flows that exceed a specified capacity.

D R A F T

Operations

Assignment:

330 **In FDP_IFF.6.1 the PP/ST author should specify the [*information flow control SFPs*] enforced by the TSF.**

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

332 **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.**

FDP_IFF.7 Information Flow Authorisation

User Application Notes

333 This component provides requirements for the information flow control functions to be able to explicitly authorise an information flow based upon security attributes.

Operations

Assignment:

334 **In FDP_IFF.7.1, the PP/ST author should specify the [*information flow control SFP*] name related to an information flow control SF which will include the special rule that requires an information flow to be explicitly authorised.**

335 **In FDP_IFF.7.1, the PP/ST author should specify the [*value of security attributes of subjects and objects*] that will be used to explicitly authorise an information flow. An example is a privilege vector associated with a subject that always allows it to receive an information flow from objects covered by the information flow control SFP that has been specified.**

FDP_IFF.8 Information Flow Authorisation and Denial

User Application Notes

336 This component provides requirements for the information flow control functions to be able to explicitly authorise and deny an information flow based upon security attributes.

D R A F T

Operations

Assignment:

337 In FDP_IFF.8.1, the PP/ST author should specify the [*information flow control SFP*] name related to an information flow control SF which will include the special rule that requires an information flow to be explicitly authorised.

338 In FDP_IFF.8.1, the PP/ST author should specify the [*value of security attributes of subjects and objects*] that will be used to explicitly authorise an information flow. An example is a privilege vector associated with a subject that always allows it to receive an information flow from objects covered by the information flow control SFP that has been specified.

339 **In FDP_IFF.8.2, the PP/ST author should specify the [*information flow control SFP*] name related to an information flow control SF which will include the special rule that requires an information flow to be explicitly denied.**

340 **In FDP_IFF.8.2, the PP/ST author should specify the [*value of security attributes of subjects and objects*] that will be used to explicitly deny an information flow. An example is a privilege vector associated with a subject that always prevents it from receiving an information flow from objects covered by the information flow control SFP that has been specified.**

D R A F T

FDP_ITC Import from Outside TSF Control

341 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.

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

User notes

343 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.

344 A variety of activities might be involved here:

a) Importing user data from an unformatted medium (e.g. floppy disk or tape), 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.

345 This family is not concerned with whether the user data may be imported. It is concerned with the values of the security attributes to associate with the imported user data.

346 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. This family addresses both cases.

347 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).

348 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 which 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).

349 Some of the well know 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.

350 These import requirements may be handled by the TSF with or without human intervention depending on the IT limitations and the organisational security policy. So, for example, if user data is received on a “confidential” channel, the security attributes of the objects will be set to “confidential”.

FDP_ITC.1 Import of User Data Without Security Attributes

User Application Notes

351 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:

352 **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.**

353 **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

354 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:

355 **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**

356 **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.**

D R A F T

FDP_ITT Internal TOE Transfer

357 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

358 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.

359 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.

FDP_ITT.1 Basic Internal Transfer Protection

Operations

Assignment:

360 **In FDP_ITT.1.1, the PP/ST author should specify the [*access control SFP* and/or *information flow control SFP*] covering the information being transferred.**

Selection:

361 **In FDP_ITT.1.1 the PP/ST author should specify the protection the user data should have while in transport. The options are [*disclosure, modification, loss of use*].**

FDP_ITT.2 Transmission Separation by Attribute

User Application Notes

362 One of the ways to achieve separation of channels based on SFP-relevant attributes is through the use of distinct encryption algorithms.

363 For example, this component could be used to provide different protection to information with different clearance levels.

Operations

Assignment:

364 In FDP_ITT.2.1, the PP/ST author should specify the [*access control SFP and/or information flow control SFP*] covering the information being transferred.

Selection:

365 In FDP_ITT.2.1 the PP/ST author should specify the protection the user data should have while in transport. The options are [*disclosure, modification, loss of use*].

Assignment:

366 **In FDP_ITT.2.2, the PP/ST author should specify the [*security attributes that require separate transmission channels*] so that the TSF can properly determine when to transmit the data via separate channels. An example is that the identity of the owner of the user data that has been transmitted is transmitted via a separate channel from the user data itself.**

FDP_ITT.3 Integrity Monitoring

User Application Notes

367 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).

368 The PP/ST author has to specify which types of errors 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.

369 The PP/ST author must specify which actions 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.

D R A F T

Operations

Assignment:

370 **In FDP_ITT.3.1, the PP/ST author should specify the [*access control SFP* and/or *information flow control SFP*] that the TSF will enforce in order to monitor user data transmissions for integrity errors.**

371 **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.**

372 **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.**

FDP_ITT.4 Attribute-Based Integrity Monitoring

373 This component is used in combination with FDP_ITT.2. It ensures that the TSF checks received user data (and their attributes) for integrity.

374 For example, this component could be used to provide different protection to information with different integrity levels such as high integrity required.

375 The PP/ST author has to specify which types of errors 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.

376 The PP/ST author should specify which attributes require a different transmission channel.

377 The PP/ST author must specify which actions 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:

378 In FDP_ITT.4.1, the PP/ST author should specify the [*access control SFP* and/or *information flow control SFP*] that the TSF will enforce in order to monitor user data transmissions for integrity errors.

379 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.

380 **In FDP_ITT.4.1, the PP/ST author should specify a list of [*security attributes that require separate transmission channels*].**

381 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.

D R A F T

FDP_RIP Residual Information Protection

382 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

383 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.

384 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”.

385 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 which exists on the off-line back-up. Therefore that same file is still available.

386 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 RIP selection of “deallocation” cannot 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.

387 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.

388 This family should apply to the objects specified in the access control SFP or the information flow control SFP as specified by the PP/ST author.

FDP_RIP.1 Subset Residual Information Protection

User Application Notes

389 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:

390 **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:

391 **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

392 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:

393 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.

D R A F T

FDP_ROL Rollback

394 This family addresses the need to return to a well defined valid state. For example 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.

395 This family is intended to assist a user in returning to a well defined valid state after the user decided that he wanted the last set of actions undone, or, for example in distributed databases, the return of all of the distributed copies of the databases to the state before an operation failed.

396 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 since there would be no information to roll back. FDP_RIP can only be used 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.

397 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 reverting to 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

398 This component allows a user or subject to undo a set of operations on a predefined set of objects.

399 The undo is only possible within certain limits, for example up to a number of characters or up to a time limit.

D R A F T

Operations

Assignment:

400 **In FDP_ROL.1.1, the PP/ST author should specify the [*access control SFP* and/or *information flow control SFP*] that will be enforced for rollback operations.**

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

402 **In FDP_ROL.1.1 the PP/ST author should specify the [*list of objects*] which are subjected to the rollback policy.**

403 **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

404 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:

405 In FDP_ROL.2.1, the PP/ST author should specify the [*access control SFP* and/or *information flow control SFP*] that will be enforced for rollback operations.

406 In FDP_ROL.2.1 the PP/ST author should specify the [*list of objects*] which are subjected to the rollback policy.

407 **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.**

D R A F T

FDP_SDI Stored Data Integrity

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

User notes

409 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.

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

411 This family differs from FDP_ITT Internal TOE Transfer which protects the user data from integrity errors while being transferred within the TOE.

FDP_SDI.1 Stored Data Integrity Monitoring

User Application Notes

412 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:

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

414 **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

415 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:

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

417 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.

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

D R A F T

FDP_UCT Inter-TSF User Data Confidentiality Transfer Protection

419 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

420 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

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

Operations

Assignment:

422 **In FDP_UCT.1.1, the PP/ST author should specify the [*access control SFP* and/or *information flow control SFP*] which will be enforced when exchanging user data.**

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

FDP_UIT Inter-TSF User Data Integrity Transfer Protection

424 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. Integrity is enforced by preventing unauthorised modification of data in transit between the two end points.

User notes

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

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

FDP_UIT.1 Basic Data Exchange Integrity

User Application Notes

427 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:

428 **In FDP_UIT.1.1, the PP/ST author should specify the [*access control SFP* and/or *information flow control SFP*] which will be enforced on the exchange of data.**

Selection:

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

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

431 **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

432 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.

Operations

Assignment:

433 **In FDP_UIT.2.1, the PP/ST author should specify the [*access control SFP* and/or *information flow control SFP*] which will be enforced when recovering user data.**

434 **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

435 This component provides the ability to recover from a set of identified transmission errors. It accomplishes this task with without help from the source Trusted IT Product.

Operations

Assignment:

436 In FDP_UIT.3.1, the PP/ST author should specify the [*access control SFP* and/or *information flow control SFP*] which will be enforced when recovering user data.

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

Class FIA

Identification and Authentication

438 A common security requirement is to control the access of users to the 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.

439 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).

440 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.

441 The FIA_UID family addresses determining the identity of a user.

442 The FIA_UAU family addresses verifying the identity of a user.

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

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

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

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

D R A F T

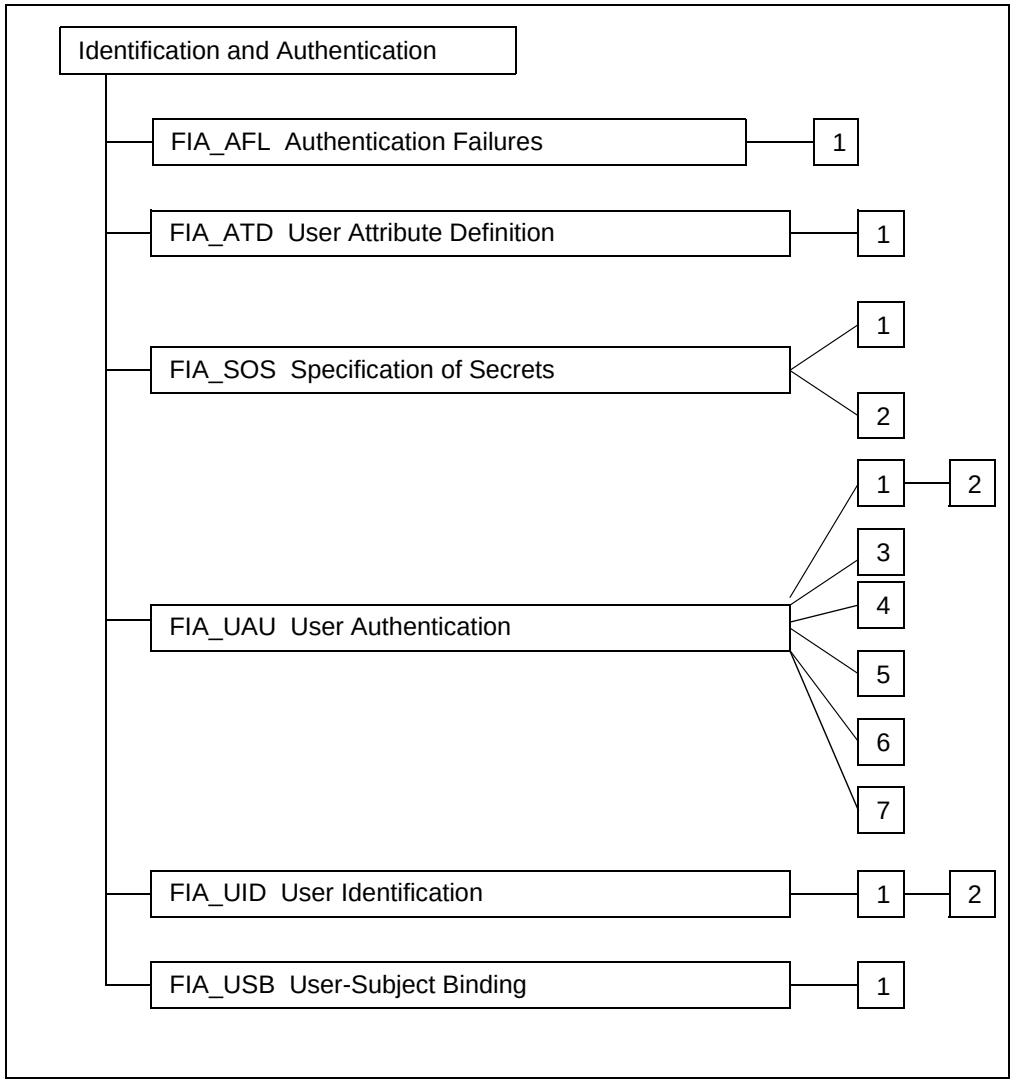

Figure B.8 - Identification and Authentication class decomposition

D R A F T

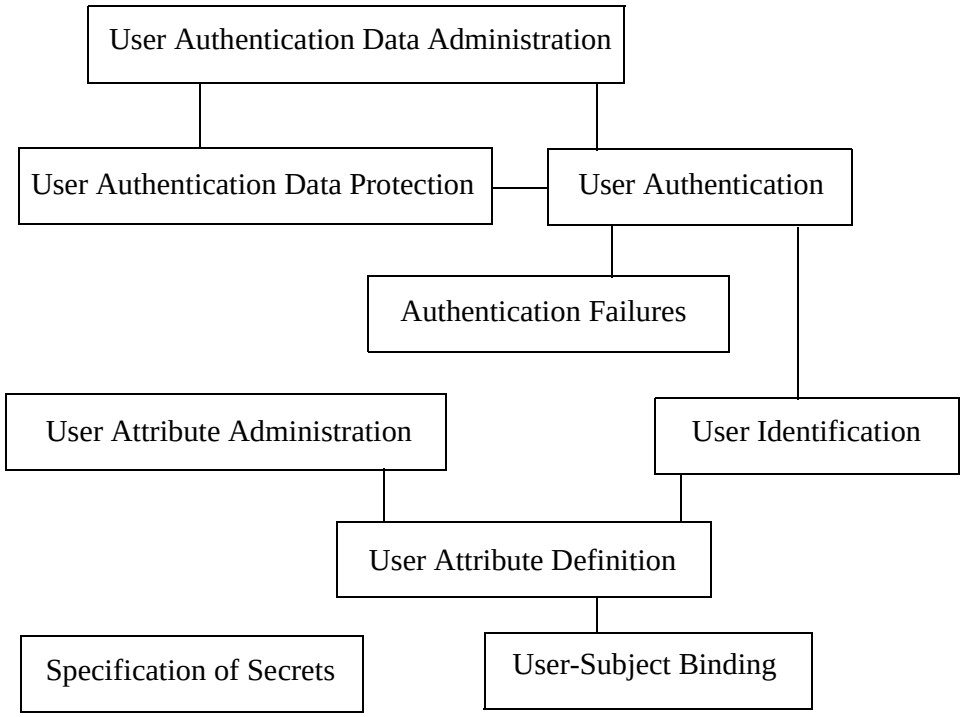

Figure B.9 - Identification and Authentication requirements construction rules

D R A F T

FIA_AFL Authentication Failures

447 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.

448 The meaning of 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 in a row exceed 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 will continue until the condition for re-establishment is satisfied.

FIA_AFL.1 Basic Authentication Failure Handling

User Application Notes

449 It is acceptable for the number of unsuccessful authentication attempts to be specified by the TOE developer. It is also acceptable if this value is also modifiable by a user authorised to perform administrative functions. 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.

450 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 deactivation, administrator alarm. The conditions under which the situation will be restored to normal must be specified on the action.

451 TOEs usually ensure that there is at least one user account that cannot be disabled in order to prevent denial of service. In order to accomplish this for such accounts as these and points of entries like the console, the condition for re-enabling the session establishment procedure could be a zero or very small time-out value that must expire.

452 The 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 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), et cetera.

Operations

Selection:

453 **In FIA_AFL.1.1, the PP/ST author must specify the threshold for the unsuccessful authentication attempts. The PP/ST author has to select either a fixed number, and/or allow the authorised administrator to configure the number. It is not acceptable that none of the two is selected.**

Assignment:

454 **In FIA_AFL.1.1, if the PP/ST author wanted to specify a default number this number must be indicated. This number must be larger than zero.**

455 **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.**

Assignment:

456 **In FIA_AFL.1.2, the PP/ST author must specify the actions to be taken in case the threshold is reached. 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).**

D R A F T

FIA_ATD User Attribute Definition

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

User notes

458 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

459 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 will have no impact on the security attributes of any other user.

460 In case security attributes belong to a group of users (such as Capability List for a group), the user will have a security attribute 'pointer to group'.

Operations

Assignment:

461 **In FIA_ATD.1.1, the PP/ST author must specify the security attributes that are associated to an individual user. This management should not be able to remove security attributes from this list. An example of such a list is {'clearance', 'group identifier', 'rights'}.**

D R A F T

FIA_SOS Specification of Secrets

462 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, automated password generation, etc.

463 A secret can be generated by separated means e.g. selected by the user and introduced in the system. In that case the FIA_SOS.1 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 not used before.

464 Another possibility is that the TOE is expected to create the secret. In that case FIA_SOS.2 can be used to require the TOE to generate the secrets that will adhere to some metrics such as a minimum size, not present in a dictionary, and not used before.

User notes

465 Secrets contain the authentication data provided by the user for an authentication mechanism that is based on knowledge the user possesses. In case cryptographic keys are employed, the class FCS should be used instead of this family.

FIA_SOS.1 Verification of Secrets

User Application Notes

466 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:

467 **In FIA_SOS.1.1, the PP/ST author must 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

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

D R A F T

User Application Notes

469 When a pseudo-random number generator is used in a secret generation algorithm, it should accept as input random data that would provide output which 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:

470 **In FIA_SOS.2.1, the PP/ST author must 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.**

471 **In FIA_SOS.2.2, the PP/ST author must 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.**

D R A F T

FIA_UAU User Authentication

472 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

473 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 considerations 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.

474 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:

475 **In FIA_UAU.1.1, the PP/ST author must 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

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

477 This component includes only minimal form of individual user authentication, and is intended for use in products that will have limited exposure to authentication attacks.

D R A F T

FIA_UAU.3 Unforgeable Authentication

User Application Notes

478 This component addresses requirements for authentication mechanisms which provide protection of authentication data. Authentication data that is copied from another user, or is in some way constructed shall be detected and rejected. This mechanism provides confidence that users authenticated by the TSF are actually who they claim to be.

479 This component may only be useful with authentication mechanisms which 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.

FIA_UAU.4 Single-use Authentication Mechanisms

User Application Notes

480 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 such things as single-use passwords, encrypted time-stamps, random numbers from a secret lookup table.

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

Operations

Assignment:

482 **In FIA_UAU.4.1, the PP/ST author must 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

483 The use of this component allows specification of requirements for more than one authentication mechanism. For each separate 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 authentication mechanism.

484 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.

485 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.

Assignment:

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

Assignment:

487 **In FIA_UAU.5.2, the PP/ST author must specify the rules that describe how the authentication mechanisms provide authentication. This means that for each situation the set of mechanisms that might be used for authenticated 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

488 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:

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

490 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.

491 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:

492 **In FIA_UAU.7.1, the PP/ST author shall 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".**

D R A F T

FIA_UID User Identification

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

FIA_UID.1 Timing of Identification

User Application Notes

494 In this component users will be identified. A user is allowed by the TSF to perform certain specified actions. The control of this component will only be on those actions whose execution the TSF can control before identification.

495 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:

496 **In FIA_UID.1.1, the PP/ST author must 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. This list cannot be empty. 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

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

D R A F T

FIA_USB User-Subject Binding

498 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

499 The phrase "acting on behalf of" has proven to be a contentious issue in the 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 as a result of the identification and authentication process, that subject is acting on behalf of the user who was identified and authenticated. 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 that case the user that created this subject is assumed to be the 'owner'.

D R A F T

Class FMT

Security Management

500 This class specifies the management of the several aspects of the TSF. The aspects consist of the security attributes, TSF data and functions. The different roles with respect to management and their interaction, such as separation of capability, can also be specified.

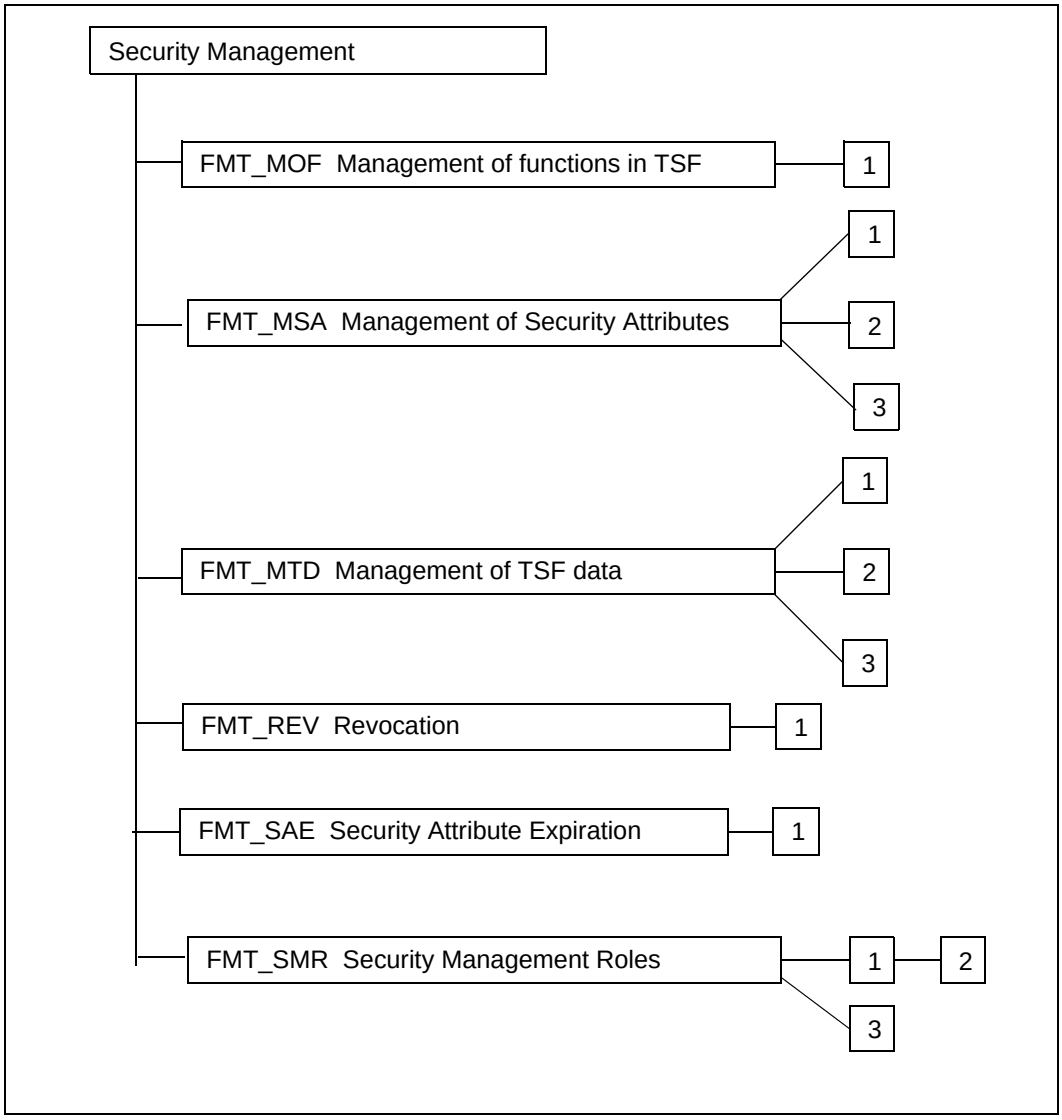

Figure B.10 - Security Management class decomposition

501 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. In such situations, use of components from FPT_TRC is advisable.

D R A F T

FMT_MOF Management of functions in TSF

502 The TSF management functions enable authorised users to set up and control the secure operation of the product. 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 auditing system controls (e.g. selection of audit events, management of audit trails, audit trail analysis, and audit report generation) and functions that define and update per-user policy attributes (such as user clearance), known system access control labels, 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.

503 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.

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

FMT_MOF.1 Management of security functions behaviour

505 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 the changing of authentication mechanisms.

Operations

Selection:

506 **In FMT_MOF.1.1 the PP/ST author should select the actions from the list *determine behaviour, disable, enable, and/or modify* that can be performed on security functions.**

Assignment:

507 **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.**

Assignment:

508 **In FMT_MOF.1.1 the PP/ST author should specify the functions that can be modified by the identified roles. Examples include auditing, or time determination.**
If the behaviour of the security function can be modified, as specified by the selection, the range of modification should be indicated. An example of such a range is "authentication function by selecting a different mechanism".

D R A F T

FMT_MSA Management of Security Attributes

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

510 Users, subjects and objects have security attributes attached which 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 or the priority of a process (subject). These security attributes need to be managed by the user, a subject or an authorised administrator.

511 FMT_MSA.2 can be used to ensure that all combinations of security attributes do not result in an insecure state. The definition of what “secure” means is left to the TOE guidance and the TSP model.

512 In some instances subjects, objects or users are created. If no explicit values for certain security attributes are given, default values need to be used. These default values can be managed by FMT_MSA.1 by specifying in the operations that the default values of the security attributes can be modified.

FMT_MSA.1 Management of security attributes

513 This component allows users with a certain role to modify values of security attributes. The users are assigned to a role within the component FMT_SMR.1.

514 The default value of a parameter is the value the parameter would take when the parameter is instantiated without specifically assigned values. An initial value is provided during the instantiation (creation) of a parameter and is meant to override the default value.

Operations

Selection:

515 **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), read or modify the security attribute, or delete the security attributes entirely.**

Assignment:

516 **In FMT_MSA.1.1 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.**

Assignment:

517 **In FMT_MSA.1.1, list the *access control SFP or the information flow control SFP* for which the object security attributes are applicable.**

Assignment:

518 **In FMT_MSA.1.1 the PP/ST author should specify the security attributes that can be modified 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.**

FMT_MSA.2 Safe security attributes

519 This component covers 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.

520 The definition of what 'safe' 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

521 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:

522 **In FMT_MSA.3.1, list the *access control SFP or the information flow control SFP* for which the object security attributes are applicable.**

Assignment:

523 **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:

524 **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.**

D R A F T

FMT_MTD Management of TSF data

525 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

526 This component allows users with a certain role to modify values of TSF data. The users are assigned to a role within the component FMT_SMR.1.

527 The default value of a parameter is the value the parameter would take when the parameter is instantiated without specifically assigned values. An initial value is provided during the instantiation (creation) of a parameter and is meant to override the default value.

Operations

Selection:

528 **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, read or modify the TSF data, or delete the TSF data entirely. To clarify clear a TSF data means that the values are removed but that the entity itself remain in the system.**

Assignment:

529 **In FMT_MTD.1.1 the PP/ST author should specify the roles that are allowed to modify the values of the TSF data. The possible roles are specified in FMT_SMR.1.**

Assignment:

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

FMT_MTD.2 Management of limits on TSF data

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

Operations

Assignment:

532 **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.**

Assignment:

533 **In FMT_MTD.2.1 the PP/ST author should specify the TSF data that can have limits and should result in the specified actions if the limit is exceeded. An example of such TSF data is the number of users logged-in.**

Assignment:

534 **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 administrator is informed and an audit record is generated.**

FMT_MTD.3 Safe TSF data

535 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.

536 The definition of what ‘safe’ 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.

D R A F T

FMT_REV Revocation

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

Documentation notes

538 AGD_ADM Administrator Guidance must describe the timing aspects of the revocation. This is especially important for TSFs with distributed architecture.

FMT_REV.1 Revocation

539 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.

d) Revocation will take place when new data of the file is requested.

Operations

Selection:

540 **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 latter option is chosen, then the PP/ST author should refine to define the resources.**

Assignment:

541 **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.**

Assignment:

542 **In FMT_REV.1.2, the PP/ST author should specify the [*revocation rules*]. Examples of this specification could include: prior to the next operation on the associated resource, or for all new subject creations.**

D R A F T

FMT_SAE Security Attribute Expiration

543 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, audit attributes, etc.

FMT_SAE.1 Time-Limited Authorisation

Operations

Assignment:

544 **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.**

Assignment:

545 **In FMT_SAE.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.**

Assignment:

546 **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 considered.**

D R A F T

FMT_SMR Security Management Roles

547 This family reduces the likelihood of damage resulting from users and from authorised administrators 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.

548 This family requires that information be maintained to identify whether a user is authorised to use a particular security-relevant administrative function.

549 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.

550 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. Hereby policies like two-person control can be enforced.

FMT_SMR.1 Security roles

551 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. The actions that a role can perform are specified in the other families in this class.

Operations

Assignment:

552 **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, administrator.**

FMT_SMR.2 Restrictions on security roles

553 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.

554 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:

555 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, **assistant,** auditor, administrator.

Assignment:

556 **In FMT_SMR.2.3 the PP/ST author should specify the conditions that should be adhered to. Examples of these conditions are: "an account cannot have the auditor and administrator role" or "a user with the assistant role should also have the owner role".**

FMT_SMR.3 Assuming roles

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

Operations

Assignment:

558 **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.**

D R A F T

Class FPR

Privacy

559 This class is based on the current available knowledge about Privacy techniques. Since research in this area is still on going, in the future these components might need expansion or revision.

560 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.

561 In the components of this class there is flexibility as to whether or not authorised administrators are covered by the required security functions. For example, in some cases a PP/ST author might consider it appropriate not to require protection of the privacy of users against a suitably authorised administrator.

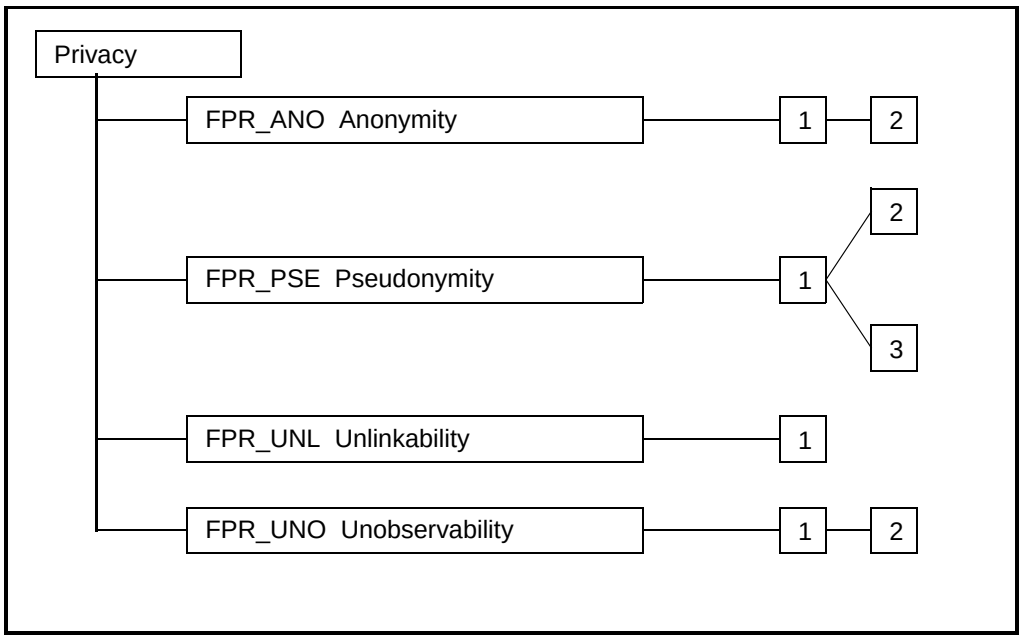

Figure B.11 - Privacy class decomposition

562 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 pose limitations on the use of the components of other classes such as FIA, FAU. For example if authorised administrators are not allowed to see the user identity (e.g. Anonymity or Pseudonymity), it will obviously not be possible to hold individual users to account 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.

563 See also the application notes for class FAU, where it is explained that the definition of ‘identity’ in the context of auditing can also cover an alias or other information which could identify a user.

564 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 threat, pseudonymity will be more appropriate than anonymity e.g. if there is a requirement for auditing.

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

566 All families in this class have components that can be scoped through the operations. The operations allow to state the number of cooperating users/subjects to which the TSF must be resistant, and whether authorised administrators (e.g. the audit authorised administrator, or the I&A authorised administrator) are included or excluded from this set. An example of an instantiation of anonymity could be: “The TSF shall ensure that two cooperating users and/or subjects, excluding authorised administrators, are unable to determine the user identity bound to the teleconsulting application”.

D R A F T

FPR_ANO Anonymity

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

User notes

568 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.

569 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. Therefore the identity of the subject is not protected from disclosure.

570 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.

571 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.

572 The component levelling distinguishes between the users and an authorised administrator. An authorised administrator is often excluded from the component and therefore allowed to retrieve a users identity. However there is no specific requirement that an authorised administrator must be able to have the capability to determine the users identity.

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

574 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.

575 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

576 This component ensures that the identity of a user is protected from disclosure.

D R A F T

Operations

Assignment:

577 **In FPR_ANO.1.1 the PP/ST author should specify the [*set of users and/ or subjects*] against which, if they are working together, the TSF must provide protection. For example if the PP/ST author specifies 'a single user or subject', the TSF must be protected against each individual user or subject but might have some weaknesses with respect to cooperating users.**

Selection:

578 **In FPR_ANO.1.1 the PP/ST author should specify whether authorised administrators are included or excluded from the scope.**

Assignment:

579 **In FPR_ANO.1.1 the PP/ST author should identify the [*list of subjects and/or operations*] where the user identity of the subject should be protected, for example "the voting application".**

FPR_ANO.2 TSF Anonymity

User Application Notes

580 This component is used to prohibit the TSF from accepting any user-identity related information.

Operations

Assignment:

581 In FPR_ANO.2.1 the PP/ST author should specify the [*set of users and/or subjects*] against which, if they are working together, the TSF must provide protection. For example if the PP/ST author specifies 'a single user or subject', the TSF must be protected against each individual user or subject but might have some weaknesses with respect to cooperating users.

Selection:

582 In FPR_ANO.2.1 the PP/ST author should specify whether authorised administrators are included or excluded from the scope.

Assignment:

583 In FPR_ANO.2.1 the PP/ST author should identify the [*list of subjects and/ or operations*] where the users identity of the subject should be protected, for example "the voting application".

Assignment:

584 **For FPR_ANO.2.2 the PP/ST author should identify the [*list of subjects*] where the users identity of the subject should be protected, for example the voting application.**

Assignment:

585 **For FPR_ANO.2.2 the PP/ST author should identify the [*list of operations*] where the users identity should be protected, for example "the accessing of job descriptions".**

D R A F T

FPR_PSE Pseudonymity

586 Pseudonymity ensures that an entity may use a resource or service without disclosing its identity, but can still be accountable for that use. The user can be accountable through directly being related to a reference (alias) held by the TSF, or by providing an alias which will be used for processing purposes such as an account number.

User notes

587 In several respects pseudonymity resembles anonymity. Both pseudonymity and anonymity protect the identity of the user, but in pseudonymity a reference to the users identity is maintained for accountability or other purposes.

588 The component FPR_PSE.1 does not specify the requirements on the alias. For the purpose of specifying requirements on this reference two sets of requirements are presented: FPR_PSE.2 and FPR_PSE.3.

589 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 an advantage to be able to trace the users identity when a check has been issued multiple times (i.e. fraud). In general the users 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 instead.

590 Another usage of the reference is as an alias for a user. For example a user does not wish to be identified, but can provide an account to which the resource utilisation should be charged. In those 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 users 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.

591 Using these constructs above, digital money can be created using FPR_PSE.2 Reversible Pseudonymity. In FPR_PSE.2 Reversible Pseudonymity will specify that the user identity will be protected and, if so specified in the condition, there can be a requirement to trace the user identity if the digital money is spent twice. Thereby when the user is honest, the user identity is protected, and if the user tries to cheat, the user identity can be traced.

592 A different kind of system could be a digital credit card, where the user will provide a pseudonym which indicates an account from which the cash can be subtracted. In that case for example FPR_PSE.3 Alias Pseudonymity could be used. FPR_PSE.3 Alias Pseudonymity will specify that the user identity will be protected and, furthermore this component will specify that the same user will only get assigned values for which he/she has provided money (if so specified in the conditions).

593 It should be realised that especially 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

D R A F T

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.

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

595 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.

596 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 contain methods for authorisation without identification, especially for anonymous payment ("Digital Cash"). This helps providers to get their payment in a secure way while maintaining customer anonymity.

FPR_PSE.1 Pseudonymity

User Application Notes

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

598 This component is dependent on either FPR_PSE.2 or FPR_PSE.3. However, these other components could be located in a separate TOE.

Operations

Assignment:

599 **In FPR_PSE.1.1 the PP/ST author should specify the [*set of users and/ or subjects*] against which, if they are working together, the TSF must provide protection. For example if the PP/ST author specifies 'a single user or subject', the TSF must be protected against each individual user or subject but might have some weaknesses with respect to cooperating users.**

Selection:

600 **In FPR_PSE.1.1 the PP/ST author should specify whether authorised administrators are included or excluded from the scope.**

D R A F T

Assignment:

601 **In FPR_PSE.1.1 the PP/ST author should identify the [*list of subjects and/or operations and/or objects*] where the user identity 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.**

Assignment:

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

603 **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:

604 **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:

605 **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

606 In this component the TSF shall ensure that under specified conditions the user identity related to a provided reference can be determined.

607 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:

608 In FPR_PSE.2.1 the PP/ST author should specify the [*set of users and/or subjects*] against which, if they are working together, the TSF must provide protection. For example if the PP/ST author specifies 'a single user or subject', the TSF must be protected against each individual user or subject but might have some weaknesses with respect to cooperating users.

Selection:

609 In FPR_PSE.2.1 the PP/ST author should specify whether authorised administrators are included or excluded from the scope.

Assignment:

610 In FPR_PSE.2.1 the PP/ST author should identify the [*list of subjects and/ or operations and/or objects*] where the users identity 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.

Assignment:

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

612 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:

613 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:

614 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:

615 **In FPR_PSE.2.4 the PP/ST author should select whether the authorised administrator and/or trusted subjects can determine the user identity.**

Assignment:

616 **In FPR_PSE.2.4 the PP/ST author should identify the *list of trusted subjects* which can obtain the users identity under a specified condition, for example a notary or special administrative role.**

Assignment:

617 **In FPR_PSE.2.4 the PP/ST author should identify the [*list of conditions*] under which the subjects and authorised administrator can determine the users identity 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.**

D R A F T

FPR_PSE.3 Alias Pseudonymity

User Application Notes

618 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.

619 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. These kind of conditions can be specified in this component.

Operations

Assignment:

620 In FPR_PSE.3.1 the PP/ST author should specify the [*set of users and/or subjects*] against which, if they are working together, the TSF must provide protection. For example if the PP/ST author specifies ‘a single user or subject’, the TSF must be protected against each individual user or subject but might have some weaknesses with respect to cooperating users.

Selection:

621 In FPR_PSE.3.1 the PP/ST author should specify whether authorised administrators are included or excluded from the scope.

Assignment:

622 In FPR_PSE.3.1 the PP/ST author should identify the [*list of subjects and/ or operations and/or objects*] where the users identity 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.

Assignment:

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

624 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:

625 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:

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

Assignment:

627 **In FPR_PSE.3.4 the PP/ST author should identify the [*list of conditions*] which indicate when the used reference for the user-identity 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.**

D R A F T

FPR_UNL Unlinkability

628 Unlinkability ensures that an entity 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

629 The requirements for unlinkability are intended to protect the user identity against the use of profiling of the operations. For example in case 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 furthermore 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.

630 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.

631 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 which must be countered.

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

633 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

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

Operations

Assignment:

635 **In FPR_UNL.1.1 the PP/ST author should specify the [*set of users and/ or subjects*] against which, if they are working together, the TSF must provide protection. For example if the PP/ST author specifies 'a single user or subject', the TSF must be protected against each individual user or subject but might have some weaknesses with respect to cooperating users.**

Selection:

636 **In FPR_UNL.1.1 the PP/ST author should specify whether authorised administrators are included or excluded from the scope.**

Assignment:

637 **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:

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

Assignment:

639 **In FPR_UNL.1.1 the PP/ST author might need to identify the [*list of relations*] which should be protected against, for example "originate from the same terminal".**

D R A F T

FPR_UNO Unobservability

640 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

641 Unobservability approaches the user identity from a different direction than the previous families Anonymity, Pseudonymity, and Unlinkability. Instead of not releasing the users identity the fact that somebody is using the resource / service is hidden.

642 Sometimes regular users are not allowed to see the use of a resource, but an authorised administrator must be allowed to see the use of the resource in order to perform his duties. In those cases the FPR_UNO.2 could be requested, which provides the capability for an authorised administrator to see the usage.

643 Examples of potential hostile users or subjects are malicious systems operators or users, who smuggle malicious parts, e.g. Trojan Horses into system. Several countries consider the protection of communications unobservability as essential for the protection of constitutional rights.

FPR_UNO.1 Unobservability

User Application Notes

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

Operations

Assignment:

645 **In FPR_UNO.1.1 the PP/ST author should specify the [*set of users and/ or subjects*] against which, if they are working together, the TSF must provide protection. For example if the PP/ST author specifies ‘a single user or subject’, the TSF must be protected against each individual user or subject but might have some weaknesses with respect to cooperating users.**

Selection:

646 **In FPR_UNO.1.1 the PP/ST author should specify whether authorised administrators are included or excluded from the scope.**

Assignment:

647 **For FPR_UNO.1.1 the PP/ST author should identify the [*list of operations*] which are subjected to the unobservability requirement. In**

D R A F T

other words the other user/subjects cannot observe the operations in the specified list on a covered object, for example reading and writing on the object.

Assignment:

648 **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.**

FPR_UNO.2 Authorised Administrator Observability

User Application Notes

649 This component is used to specify that there will be an authorised administrator with the rights to view the resource utilisation. Without this component, this review is allowed, but not mandated.

Operations

Assignment:

650 In FPR_UNO.2.1 the PP/ST author should specify the [*set of users and/or subjects*] against which, if they are working together, the TSF must provide protection. For example if the PP/ST author specifies ‘a single user or subject’, the TSF must be protected against each individual user or subject but might have some weaknesses with respect to cooperating users.

Selection:

651 In FPR_UNO.2.1 the PP/ST author should specify whether authorised administrators are included or excluded from the scope.

Assignment:

652 For FPR_UNO.2.1 the PP/ST author should identify the [*list of operations*] which are subjected to the unobservability requirement. In other words the other user/subjects cannot observe the operations in the specified list on a covered object, for example reading and writing on the object.

Assignment:

653 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

D R A F T

Class FPT

Protection of the TOE Security Functions

654 This class contains families of functional requirements that relate to the integrity and management of the mechanisms that provide the TSF (independent of TSP-specifics) and to the integrity of TSF data (independent of the specific contents of the TSP data). In some sense, families in this class may appear to duplicate components in the FDP (User Data Protection) class, they may even be implemented using the same mechanisms; however, FDP focuses on user data protection, while FPT focuses on TSF data protection. In fact, components from the FPT class are necessary even in the absence of any user data protection, to provide confidence in the enforcement of other policies (such as accountability) that may be specified in the PP/ST.

D R A F T

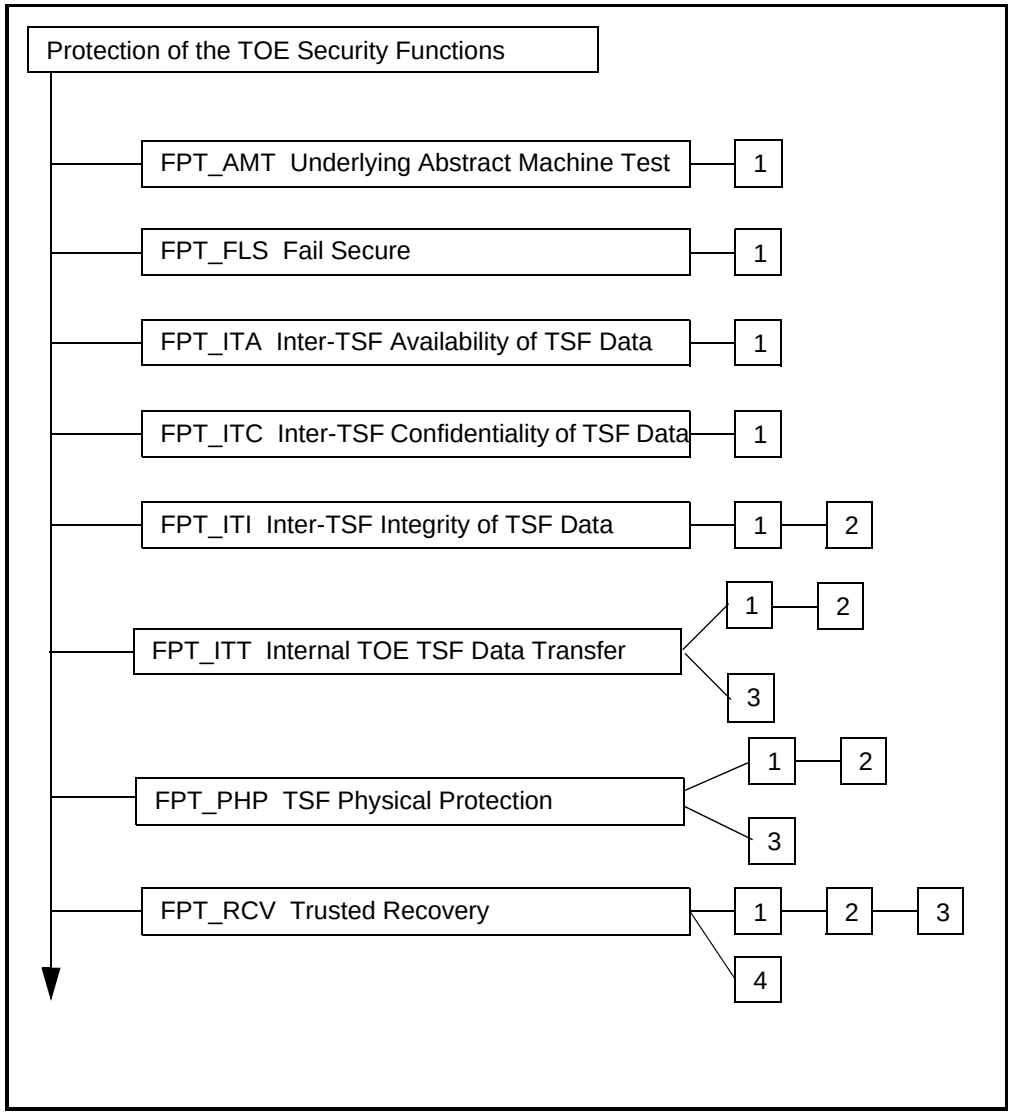

Figure B.12 - Protection of the TOE Security Functions class decomposition

D R A F T

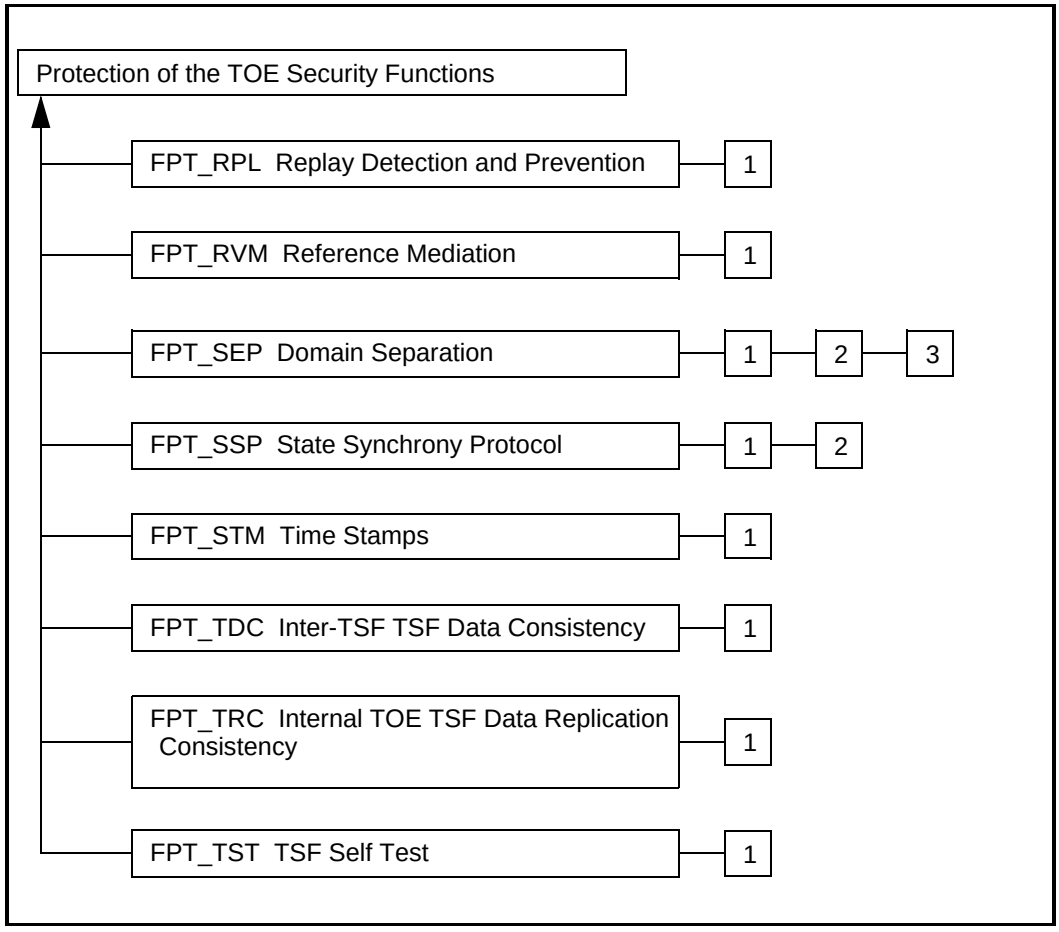

Figure B.13 - Protection of the TOE Security Functions class decomposition (Cont.)

655 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 software under evaluation executes.

b) The TSF's *software*, 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.

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

a) Families that address protection of the TSF mechanisms. These families are:

1) FPT_PHP (TSF Physical Protection) provides the authorised administrator with the ability to detect external attacks on the parts of the TOE that comprise the TSF.

2) FPT_AMT (Underlying Abstract Machine Test) and FPT_TST (TSF Self Test), which provide the authorised administrator 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.

3) 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.”

4) 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.

5) 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.

6) FPT_ITT (Internal TOE TSF Data Transfer) addresses protection of TSF data when it is transmitted between parts of the TOE.

7) FPT_RPL (Replay Detection and Correction), which addresses the replay of various types of information and/or operations.

8) FPT_SSP (State Synchrony Protocol), which addresses the state synchrony required between two parts of the TSF.

9) FPT_STM (Time Stamps), which addresses reliable timing.

b) Families that address the TSF data. This families is:

1) FPT_TDC (Inter-TSF TSF Data Consistency), which addresses the consistency of TSF data shared between TSF of distinct TOEs.

D R A F T

FPT_AMT Underlying Abstract Machine Test

657 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 could be testing hardware page protection, sending sample packets across a network to ensure receipt, verifying the behaviour of the virtual machine interface, etc. 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

658 The term "underlying abstract machine" typically refers to the hardware components upon which the TSF software functions have 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.

658 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 the authorised administrator or be activated by specific conditions. This might include processor component stress tests (logic units, calculation units, etc.) and control memory.

Evaluator notes

659 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

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

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

Evaluator application notes

662 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 administrators.

Operations

Selection:

663 **In FPT_AMT.1.1 the PP/ST author should specify the when the TSF will execute the abstract machine testing, *during initial start-up, periodically during normal operation, at the request of the authorised administrator, other conditions*. In the case of the latter option, the PP/ ST author should refine what those conditions are.**

D R A F T

FPT_FLS Fail Secure

664 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

665 The term “secure state” refers to a state in which the TSF data are consistent and the TSF continues correct enforcement of the TSP.

666 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.

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

Operations

Assignment:

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

D R A F T

FPT_ITA Inter-TSF Availability of TSF Data

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

User Application Notes

670 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 hold responsible for the TSF at the other trusted IT product.

FPT_ITA.1 Inter-TSF Availability Within a Defined Availability Factor

Operations

Assignment:

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

Assignment:

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

Assignment:

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

FPT_ITC Inter-TSF Confidentiality of TSF Data

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

User Application Notes

675 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 hold responsible for the TSF at the other trusted IT product.

FPT_ITC.1 Inter-TSF Confidentiality During Transmission

Evaluator application notes

676 With the technology available at the time of writing of the Common Criteria, the only practical means of satisfying this requirement involves either physical protection of the transmission lines, or the use of cryptographic functions.

D R A F T

FPT_ITI Inter-TSF Integrity of TSF Data

677 This family defines the rules for the protection, from unauthorised modification, of TSF data moving between the TSF and a remote trusted IT product. This data could, for example, be TSF critical data such as passwords, keys, audit data, or TSF executable code.

User notes

678 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.

FPT_ITI.1 Inter-TSF Detection of Modification

User Application Notes

679 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.

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

Evaluator application notes

681 With the technology available at the time of writing of the Common Criteria, the only practical means of satisfying this requirement involves either physical protection of the transmission lines, or the use of cryptographic functions.

Operations

Assignment:

682 **For FPT_ITI.1.1, the PP/ST should specify the *modification metric* which the detection mechanism must satisfy.**

Assignment:

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

D R A F T

FPT_ITI.2 Inter-TSF Detection and Correction of Modification

User Application Notes

684 This component should be used in situations where it is necessary to detect or correct modifications of TSF critical data.

685 The desired strength of modification detection is a function of the algorithm used, ranging from weak checksumming 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 mechanisms well known in the public literature (e.g. the strength must be conformant to the strength offered by Secure Hash Algorithm).

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

Evaluator application notes

687 With the technology available at the time of writing of the Common Criteria, the only practical means of satisfying this requirement involves the use of cryptographic functions, protecting the transmission itself (e.g. anti-jamming) or some form of checksum.

Operations

Assignment:

688 For FPT_ITI.2.1, the PP/ST should specify the *modification metric* which the detection mechanism should satisfy.

Assignment:

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

Assignment:

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

D R A F T

FPT_ITT Internal TOE TSF Data Transfer

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

User notes

692 The determination of the degree of physical separation which 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. In more benign environments, the transfers may be across more traditional network media.

Evaluator notes

693 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:

694 **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

695 One of the ways to achieve separation of channels based on SFP-relevant attributes is through the use of distinct encryption algorithms.

Operations

Selection:

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

D R A F T

FPT_ITT.3 TSF Data Integrity Monitoring

Operations

Selection:

697 **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*. If the latter option is selected, the PP/ST author should refine those other errors.**

Assignment:

698 **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.**

Assignment:

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

D R A F T

FPT_PHP TSF Physical Protection

700 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.

701 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.

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

User notes

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

704 Although there is only limited audit, 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 an Light Emitting Diode (LED) if the circuit is broken when a button is pressed by the authorised administrator). Nevertheless, a PP/ST author may determine that for a particular anticipated threat environment there is a need to audit physical attacks. 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

705 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. The authorised administrator

is given the function to verify whether an attack took place. 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

706 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 attacks. It addresses the threat that physical tampering with TSF elements, although detected, may not be noticed.

Operations

Assignment:

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

Assignment:

708 **For FPT_PHP.2.3, the PP/ST author should identify the type of administrative *user or role* that is to be notified when tampering is detected. The administrative 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

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

User Application Notes

710 This component should be used when TSF devices and elements are expected to operate in an environment where a physical attack (e.g. observation, analysis, or modification) of the internals of a TSF device or element itself is a threat. This component partially addresses the threat of the TSF violating the TSP as the result of a physical attack, by providing increased resistance to attack.

Evaluator application notes

711 The determination of acceptable work factors is by its very nature somewhat qualitative, and cannot always be evaluated in a reasonable time or in a repeatable fashion. Evaluator judgement will be required to determine if a particular attack scenario resistance mechanism would require the indicated level of effort.

Operations

Assignment:

712 **For FPT_PHP.3.1, the PP/ST author should specify both the devices/ elements for which the TSF should resist physical tampering attacks, and the specific attack scenario that should be countered. 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.**

Assignment:

713 **For FPT_PHP.3.2, the PP/ST author should specify both the devices/ elements for which the TSF should automatically respond to physical tampering attacks, and the specific attack scenarios that should be countered. 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. 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 to that the protected information may not be retrieved.**

D R A F T

FPT_RCV Trusted Recovery

714 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. Satisfying the requirements of this family establishes that the initial and recovered states of the TSF satisfy the requirements. This family is important because the start-up state of the TSF determines the protection of subsequent states.

715 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).

716 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.

717 This family identifies a maintenance mode. In this maintenance mode normal operation might be impossible or severely restricted since otherwise insecure situations might occur. Typically only authorised administrators are allowed access.

718 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

719 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.

FPT_RCV.1 Manual Recovery

720 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

721 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

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

FPT_RCV.2 Automated Recovery

723 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

724 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

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

726 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.

727 It is assumed that the robustness of the automated recovery mechanisms will be verified.

Operations

Assignment:

728 **For FPT_RCV.2.3, the PP/ST author should specify the *list of failures or other discontinuities* for which automated recovery shall be possible.**

FPT_RCV.3 Automated Recovery without Undue Loss

729 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

730 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.

731 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

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

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

Operations

Assignment:

734 For FPT_RCV.3.3, the PP/ST author should specify the *list of failures or other discontinuities* for which automated recovery shall be possible.

Assignment:

735 **For FPT_RCV.3.4, 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

735 For selected SFs in the TSF, it is necessary that the SF fail in a manner that does not result in compromised TSF data.

Operations

Assignment:

736 **In FPT_RCV.4.1, the PP/ST author should *list the SFs and failure scenarios* for which the TSF should return to its state immediately before SF invocation.**

D R A F T

FPT_RPL Replay Detection and Prevention

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

FPT_RPL.1 Replay Detection and Prevention

User Application Notes

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

Operations

Assignment:

739 **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.**

740 **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, terminating the subject from which the re-played entity originated.**

D R A F T

FPT_RVM Reference Mediation

741 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 a given SFP, 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.

742 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.

743 This component states that, “the TSF shall ensure that TSP enforcement functions are invoked and succeed before any operation 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.

744 A TSF that implements a SFP provides effective protection against unauthorized operation 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.

D R A F T

FPT_RVM.1 Non-Bypassability of the TSP

User Application Notes

745 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 must cover all objects.

D R A F T

FPT_SEP Domain Separation

746 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.

747 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 transfers between domains are controlled such that arbitrary entry to, or return from, the protected domain is not possible.

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

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

User notes

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

749 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 must cover all objects.

FPT_SEP.1 TSF Domain Separation

750 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.

D R A F T

FPT_SEP.2 SFP Domain Separation

751 The most important function provided by a TSF is the enforcement of its SFPs. In order to ensure 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

752 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.

753 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.

754 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:

755 **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

756 The most important function provided by a TSF is the enforcement of its SFPs. In order to ensure that the TSF exhibits the characteristics of a reference monitor (RM), in particular, being tamperproof, all access control and/or information flow control FSPs must be enforced in a domain distinct from the remainder of the TSF.

Evaluator application notes

757 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.

758 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.

D R A F T

FPT_SSP State Synchrony Protocol

758 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.

758 FPT_SSP establishes the requirement for certain critical security functions of the TSF to use this 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

759 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

760 In this component, the TSF must supply an acknowledgement to another TSF when requested by that other TSF. This acknowledgement should indicate that the TSF successfully received an unmodified transmission from the remote trusted IT product.

FPT_SSP.2 Mutual Trusted Acknowledgement

User Application Notes

761 In this component, in addition to being able to provide an acknowledgement for the receipt of a data transmission, the TSF must comply with a remote trusted IT product's request for an acknowledgement to the acknowledgement.

762 For example, the local TSF transmits some data to a remote trusted IT product. The remote trusted IT product 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 TSFs involved in the data transmission know that the transmission completed successfully.

D R A F T

FPT_STM Time Stamps

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

User notes

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

FPT_STM.1 Reliable Time Stamps

Operation : No permitted operation.

D R A F T

FPT_TDC Inter-TSF TSF Data Consistency

763 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.

User notes

764 The components in this family are intended to provide requirements for automated support for TSF data consistency when it is transferred between the TSF of this TOE and that of 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.

765 This family is different from FDP_ETC and FDP_ITC because those two families are concerned with resolving the security attributes between the TSF and its import/ export medium only.

766 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

767 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 for the TSFs 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 on the sending TSF) by the receiving TSF, the TSFs must use a pre-established protocol to exchange TSF data.

Operations

Assignment:

767 **In FPT_TDC.1.1, the PP/ST author should define the *list of TSF data types* that shall be consistently interpreted when shared between TSFs.**

Assignment:

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

D R A F T

FPT_TRC Internal TOE TSF Data Replication Consistency

768 The requirements of this family are needed to ensure the consistency of TSF data when such data is replicated internal to the TOE. Such data may become inconsistent if the internal channel between parts of the TOE becomes inoperative. If the TOE is internally structured as a network of parts of the TOE, this can occur when parts become disabled, network connections are broken, and so on.

User notes

769 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.

770 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 TOE Data Consistency

Operations

Assignment:

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

D R A F T

FPT_TST TSF Self Test

772 The family defines the requirements for the self-testing of the TSF with respect to some expected correct operation. Examples are calls to enforcement functions, and sample arithmetical operations on critical parts of the TOE. These tests can be carried out at start-up, periodically, at the request of the administrator, 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.

773 The requirements of this family are also needed to detect the corruption of TSF data and code 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.

User notes

774 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

775 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.

776 The checks on the correctness of the TSF executable code 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.

Evaluator application notes

777 It is acceptable for the functions that are available to the authorised administrator 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 administrators.

Operations

Selection:

778 **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 the authorised administrator, 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:

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

780

D R A F T

Class FRU

Resource Utilisation

781 This class provides three families which support the availability of required resources such as processing capability and/or storage capacity when needed. 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..

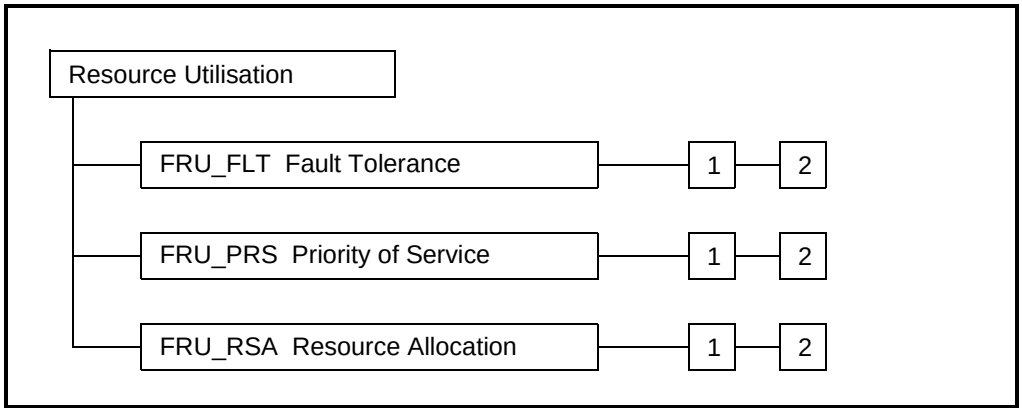

Figure B.14 - Resource Utilisation class decomposition

D R A F T

FRU_FLT Fault Tolerance

782 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

783 Since 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.

784 The mechanisms to provide fault tolerance could be active or passive. In case of an active mechanism, specific functions are in place which 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).

785 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

786 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 overflow of buffer.

Operations

Assignment:

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

Assignment:

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

D R A F T

FRU_FLT.2 Limited Fault Tolerance

User Application Notes

789 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:

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

D R A F T

FRU_PRS Priority of Service

791 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 which are less time critical.

792 This family could be applicable to several types of resources, for example, processing capacity, and communication channel capacity.

793 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

794 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

795 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:

796 **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

797 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.

D R A F T

FRU_RSA Resource Allocation

798 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

799 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.

800 In general, these functions will be implemented through the use of attributes assigned to users and resources.

801 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 a subject is the entity that is allocated the resource (e.g. main memory or CPU cycles). In those instances the components allow that the resource allocation be on the level of subjects.

802 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

803 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:

804 **In FRU_RSA.1.1, the PP/ST author should specify the list of *controlled resources* for which 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:

805 **In FRU_RSA.1.1, the PP/ST author should select whether the maximum quotas apply to *individual users or to a defined group of users* or both.**

Selection:

806 **In FRU_RSA.1.1, the PP/ST author should select whether the maximum quotas can be used at the *simultaneously or whether they apply to a period of time* in which they can be used.**

FRU_RSA.2 Minimum and Maximum Quotas

User Application Notes

807 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:

808 In FRU_RSA.2.1, the PP/ST author should specify the *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:

809 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.

Selection:

810 In FRU_RSA.2.1, the PP/ST author should select whether the maximum quotas can be used at the *simultaneously or whether they apply to a period of time* in which they can be used.

Assignment:

811 **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:

812 **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.**

Selection:

813 **In FRU_RSA.2.2, the PP/ST author should select whether the minimum quotas can be used at the *simultaneously or whether they apply to a period of time* in which they can be used.**

814

D R A F T

Class FTA

TOE Access

815 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.

816 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.

817 Figure B.15 shows the decomposition of this class into its constituent components.

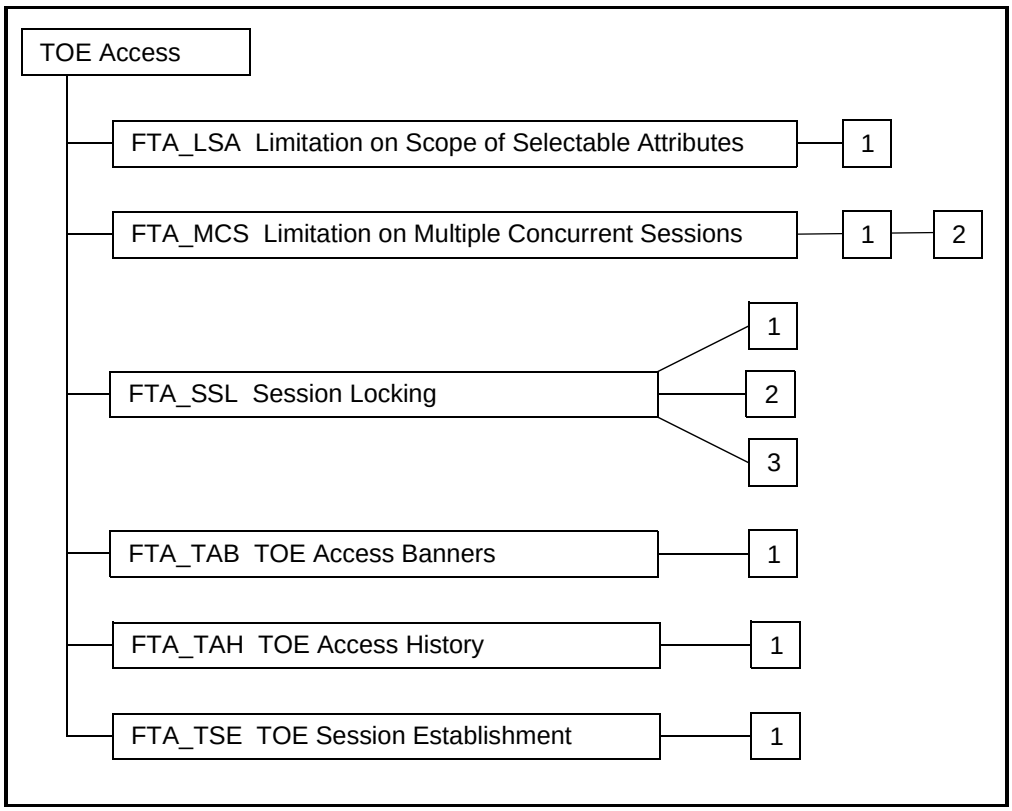

Figure B.15 - TOE Access class decomposition

D R A F T

FTA_LSA Limitation on Scope of Selectable Attributes

818 This family defines requirements that will limit the 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

819 This family provides the capability for a PP/ST authors to specify requirements for the TSF to place limits of 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 multiple times for each attribute. Attributes limitations can be specified in terms of any combination of the following parameters:

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:

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

Assignment:

821 **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.

D R A F T

FTA_MCS Limitation on Multiple Concurrent Sessions

822 This family defines how many sessions a user can 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

823 This component allows the system to limit the number of sessions in order to effectively use the resources of the TOE.

FTA_MCS.2 Per User Attribute Limitation on Multiple Concurrent Sessions

User Application Notes

824 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. This provides protection against actions that cannot be properly monitored or where procedural measures cannot be properly put in place.

Operations

Assignment:

825 **For FTA_MCS.2.1 the PP/ST author should specify the *security attributes* that can be used to specify the maximum number of sessions per user. Examples of these security attributes are group-id, classification level, and location.**

Selection:

826 **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".**

D R A F T

FTA_SSL Session Locking

827 This family defines requirements for the TSF to provide the capability for locking and unlocking of interactive sessions (e.g. keyboard locking).

828 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 must authenticate himself to the TSF, must occur.

829 A user is considered inactive, if he/she has not provided any stimulus to the TOE for a period of time.

830 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

831 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.

832 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:

833 **In FTA_SSL.1.1 the PP/ST author should specify the interval. 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.**

Assignment:

834 **In FTA_SSL.1.2 the PP/ST author should specify the event that should occur before the session is unlocked. Examples of such an event are: "user re-authenticate him/herself", or "user enters unlock key-sequence".**

D R A F T

FTA_SSL.2 User-initiated Locking

User Application Notes

835 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.

Operations

Assignment:

836 **In FTA_SSL.2.2 the PP/ST author should specify the event that should occur before the session is unlocked. Examples of such an event are: "user re-authenticate him/herself", or "user enters unlock key-sequence".**

FTA_SSL.3 TSF-initiated Termination

User Application Notes

837 FTA_SSL.3 TSF-initiated Termination, requires that the TSF shall terminate an interactive user session after a period of inactivity.

838 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:

839 **In FTA_SSL.3.1 the PP/ST author should specify the interval. 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.**

D R A F T

FTA_TAB TOE Access Banners

840 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.

D R A F T

FTA_TAH TOE Access History

841 This family defines requirements for the TSF to display to users, upon successful session establishment to the TOE, a history of successful and 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 successful, and unsuccessful attempts to access the TOE since the last successful access by the identified user.

FTA_TAH.1 TOE Access History

842 This family can provide authorised users with information that may indicate the possible misuse of their user account.

Operations

Selection:

843 **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).**

844 **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).**

D R A F T

FTA_TSE TOE Session Establishment

845 This family provides the ability to place constraints on the establishment of a user session. These constraints can be specified in terms of a user's attributes such as the user identity, role, or confidentiality level.

846 This family defines requirements to deny an authorised 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

847 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. Session establishment constraints can be specified in terms of any combination of the following parameters:

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 may take place at a different location 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.

D R A F T

FTA_TSE.1 TOE Session Establishment

Operations

Assignment:

848 **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).**

Class FTP

Trusted Path/Channels

849 Users often need to perform functions through direct interaction with the TSF. A trusted path ensures 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.

849 Figure B.16 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.

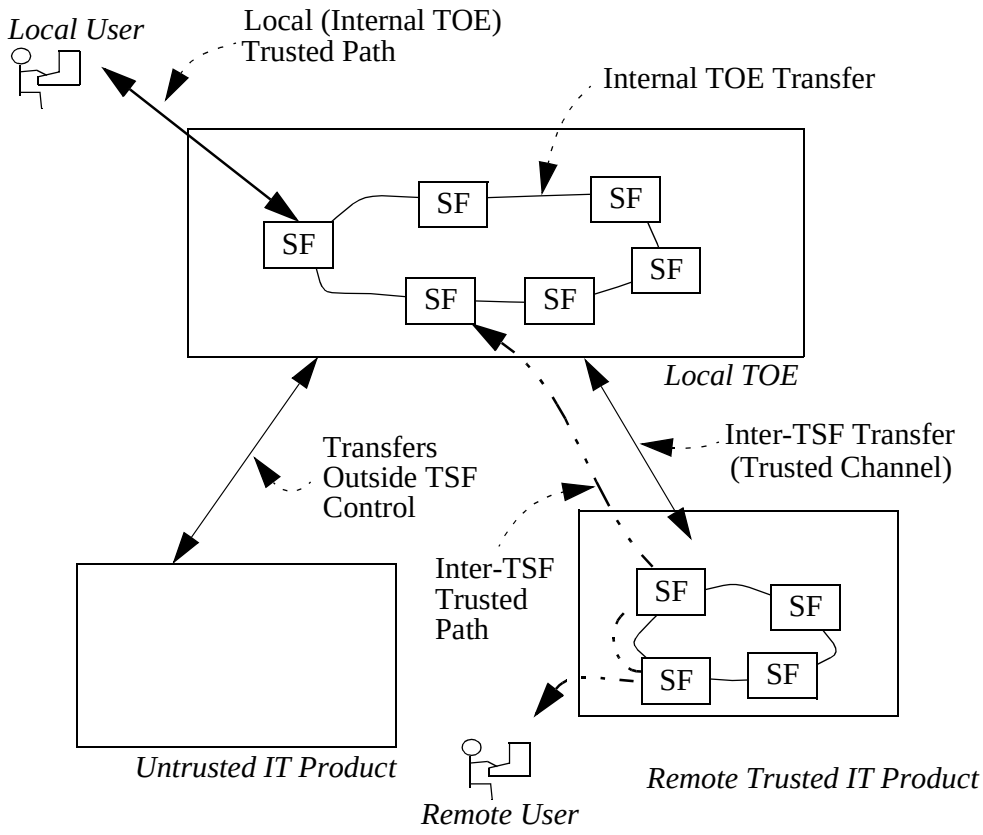

Figure B.16 - Trusted Paths and Trusted Channels

D R A F T

850 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.

851 Figure B.17 shows the decomposition of this class into its constituent components.

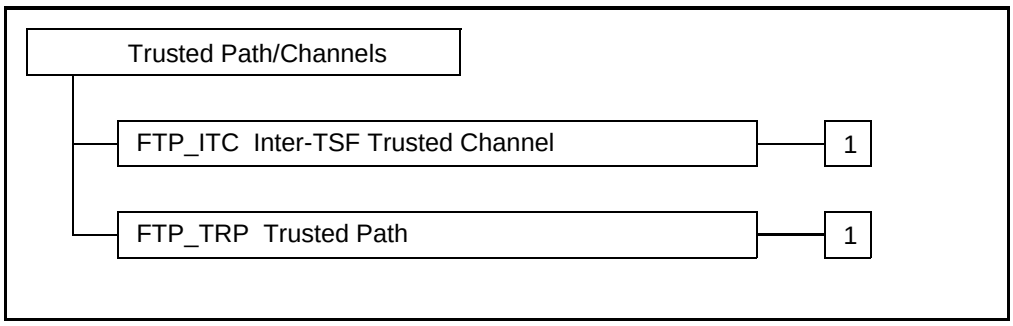

Figure B.17 - Trusted Path / Channels class decomposition

D R A F T

FTP_ITC Inter-TSF Trusted Channel

852 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. Examples of such security critical operations may include 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

853 This component should be used when a trusted communication channel between the TSF and another trusted IT product is required.

Operations

Selection:

854 **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:

855 **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.**

D R A F T

FTP_TRP Trusted Path

856 This component 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

857 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:

858 **In FTP_TRP.1.1, the PP/ST author should specify whether the trusted path must be extended to *remote and/or local* users.**

Selection:

859 **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.**

Selection:

860 **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:

861 **In FTP_TRP.1.3, the PP/ST author should identify *other services for which trusted path is required,* if any.**

D R A F T

Annex C

CC observation report (CCOR)

C.1 Introduction

862 The CC sponsoring organisations welcome feedback from the community and are particularly interested in observations and comments arising out of application of the criteria.

863 The CC sponsoring organisations have set up a body to coordinate and learn from the community experience and to ensure that future issues of the CC can benefit from that experience.

864 Comments, observations, and requests for interpretations should be sent to one of the addresses listed inside the front cover of the CC. If you require feedback on a specific evaluation matter, you should use the contact address which corresponds to the evaluation authority concerned.

C.2 Format of observation report

865 In order to allow for the automated categorisation of the observations, a standard observation format is needed.

866 The following provides a description of each structure of the required comment format and an example of a comment in the required format.

867 If you are submitting one or more observations by electronic mail or other machine readable format, you must use the ASCII text format to guarantee that your submission can be process by an automated tool. You must also insert the tags defined below, each starting in the first column, as this will greatly assist in the automated handling of your input.

868 Each observation report should consist of three parts.

a) The first part consists of a tags **$1:** to **$4:**, which includes the information to allow the unique identification of the originator. This first set of tags is required only once per single observation or batch of observations.

b) The second part consists of tags **$5:** to **$9:**, which includes the information to allow the unique identification and categorisation of the observation, the actual observation itself and suggested solution. The text of each observation should extend to as many lines as are needed to fully express the observation. There can be one or more observations in an observation report.

D R A F T

The set of tags **$5:** to **$9:**, comprising this second part of the observation report, should be repeated for each observation being submitted.

c) The third part consists of a single terminating tag **$$:**. This final tag is required only once per single observation or batch of observations.

C.2.1 Tag definitions for observation report

869 Each tag must start at the first column of a new line.

$1: Originator name

870 The characters "**$1:**" without the quotation marks, followed on the same line by the name of commenter (only required once per message).

$2: Originator organisation

871 The characters "**$2:**" without the quotation marks, followed on the same line by the originator organisation/affiliation (only required once per message).

$3: Return address

872 The characters "**$3:**" without the quotation marks, followed on the same line by the electronic mail or other address for response (only required once per message).

$4: Date

873 The characters "**$4:**" without the quotation marks, followed on the same line by the submission date of observation (only required once per message). The date should be formatted as:

YYMMDD

where YY refers to the last two digits of the calendar year, MM refers to the two digit representation of the month, and DD refers to the two digit representation of the day. For example, 29 December 1997 should be formatted as:

971229

and 5 January 1998 should be formatted as:

980105

$5: Originator report reference identification

874 The characters "**$5:**" without the quotation marks, followed on the same line by the reference for observation which is unique to originator. Please include your initials or similar unique discriminator, e.g. ABC1234.

$6: One line summary/title of observation

875 The characters "**$6:**" without the quotation marks, followed on the same line by the short summary/title for problem (up to 60 characters).

D R A F T

$7: CC document reference

876 The characters "**$7:**" without the quotation marks, followed on the same line by the single reference to the affected area of the CC as detailed as appropriate. The CC version for which the comment is being provided is required. Where possible, part number, section, paragraph, class, family, component, or requirement reference should be provided.

877 The template for CC document reference is as follows:

$7: Version / Part / Document Identifier / Keyword

878 The CC document reference template should be completed as follows (see below for completed example):

a) The characters "**$7:**" without the quotation marks, to indicate the start of an observation.

b) Identification of the Version. The CC Version can be found on the title page of each CC Part. It can also be found in the footer of every internal page within each Part. Some examples are:

Version 1.0
Version 2.0
Version 2.0 Beta
Version 2.0 Draft

c) A "/" character, without the quotes, should be inserted between the Version and the Part identifiers.

d) Part:
Valid identifiers for the CC Part are:

P1 for Part 1
P1A for Part 1 Annex A
P1B for Part 1 Annex B
P1C for Part 1 Annex C
P1D for Part 1 Annex D
P1E for Part 1 Annex E
P2 for Part 2
P2A for Part 2 Annex A
P3 for Part 3
P3A for Part 3 Annex A
P3B for Part 3 Annex B
P3C for Part 3 Annex C

e) A "/" character, without the quotes, should be inserted between the Part and the Specific Document identifiers.

f) The Specific Document Identifier to which the comment applies in the CC. It should be as specific as is possible. The following list of options is

provided in order of decreasing detail, such that if an option applies to your comment (when checking the options in order) then you should follow the directions within that option. If your comment applies to more than one of the options below, then you should consider following the directions in those additional options to determine other document identifiers and separate the resulting list of document identifiers with a comma.

If the comment refers to something within a paragraph, then that paragraph number should be provided (e.g. 232).

If the comment refers to an element then the complete element identifier should be provided (e.g. FIA_ATD.1.1).

If the comment refers to a component then the complete component identifier should be provided (e.g. ADV_FSP.1). Additionally, any relevant page numbers could also be provided (e.g. 123-123).

If the comment refers to a family then the complete family identifier should be provided (e.g. FAU). Additionally, any relevant page numbers could also be provided (e.g. 123-123).

If the comment refers to a section then the complete section identifier, preceded by the word “Section” should be provided (e.g. Section 3.1.1). Additionally, any relevant page numbers could also be provided (e.g. 123-123).

g) A “/” character, without the quotes, should be inserted between the Specific Document identifier and the Keyword (if a keyword is provided).

h) An optional keyword can be provided if the author of the CCOR feels it would be helpful.

$8: Statement of observation

879 The characters “**$8:**” without the quotation marks, followed on the same (or a new) line by the comprehensive statement of observation or query. This field can span several lines. It must contain the actual text of the observation. It should include specific reference to examples of the observation, where appropriate.

$9: Suggested solution

880 The characters “**$9**” without the quotation marks, followed on the same (or a new) line by the proposed solution or solution approach. This field can span several lines. It should include specific replacement text when possible.

$$: Terminating tag

881 The characters “**$$:**” without the quotation marks. This enables an automated handling system to determine the end of the batch of observations (only required once per batch of observations).

D R A F T

C.2.2 Example observations:

$1: A. N. Other
$2: PPs ‘R’ US
$3: another@ppsrus.com
$4: 980131
$5: ano.comment.1
$6: Presentation comment.
$7: P2 / FDP_ACF.1 / Italicise
$8: The operations in the component FDP_ACF.1 should be italicised.
$9: Italicise the operations.
$5: ano.comment.2
$6: Missing requirement for audit.
$7: P2 / FAU, pg. 336 /
$8: The first sentence of this paragraph is incomplete.
$9: The first sentence should include “imminent” violations.
$$: This is the end tag, the contents are immaterial.

D R A F T