

COMMITTEE DRAFT ISO/IEC 3rd CD 15408-2, revision		Reference document: SC 27 N19506	
Date: 2019-07-12		Supersedes document N18804	
THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.			
ISO/IEC JTC 1/SC 27 Information security, cybersecurity and privacy protection Secretariat: Germany	Circulated to P- and O-members, and to technical committees and organizations in liaison for comments by: 2019-09-06 Please submit your comments via the online balloting application by the due date indicated.		
ISO/IEC 3rd CD 15408-2, revision Title: IT Security techniques – Evaluation criteria for IT security — Part 2: Security functional components Project: 1.27.16.02 (ISO/IEC 15408-2, revision)			
Explanatory Report			
Status	SC 27 Decision	Reference documents	
		Input	Output
<i>For details regarding previous development stages refer to 2nd page of this explanatory report.</i>			
ISO/IEC 15408-2 1st WD	54 th WG 3 meeting, April 2017, Recommendations 5, 10, 11, 14 (N17041 = WG 3 N1413).	Results of call f. editor (N17276); SoV (N17026).	Call f. project editor (N17319); Liaisons to: CCDB (WG 3 N1391); The Open Group (WG 3 N1394); ISO/TC 22/SC 32 (N17373); Text f. 1 st WD (WG 3 N1436).
ISO/IEC NP 15408-2 (revision) 2nd WD	55 th WG 3 meeting, October / November 2017, Recommendations 8, 10 (N17666 = WG 3 N1494).	Results of call f. editor (N17389); SoCom (WG 3 N1464); Draft DoC (WG 3 N1501).	Call / NB nomination for /of (N17319 / N17389); Editor's report (WG 3 N1465); Liaisons to: CCDB (WG 3 N1455); ISO/TC 22/SC 32 (N18103); DoC (WG 3 N1462); Text f. 2 nd WD (WG 3 N1466).
ISO/IEC 15408-2 1st CD	56 th WG 3 meeting, April 2018, Recommendations 8, 10 (N18471 = WG 3 N1557) / 30 th SC 27 Plenary, April 2018, Resolution 6 (N18710)	SoCom (WG 3 N1528); Late Com (WG 3 N1563).	Liaison to: CCDB (WG 3 N1521); DoC (WG 3 N1527); Text f. 1 st CD (N18701).
ISO/IEC 15408-2 2nd CD	57 th WG 3 meeting / CRM, Sep / Oct 2018, Recommendations 8, 10 (N18471 = WG 3 N1557) / 30 th SC 27 Plenary, April 2018, Resolution 6 (N18710)	SoV (N18852); Draft DoC (N18924).	Liaison to: CCDB (WG 3 N1619); DoC (N18802); Text f. 2 nd CD (N18804).
ISO/IEC 15408-2 3rd CD	58 th WG 3 meeting / CRM April 2019, Recommendations 12, 14, 17, 21 (N19523 = WG 3 N1676).	SoV (N19488).	Liaison to: CCDB (WG 3 N1680); DoC (N19504); Text f. 3 rd CD (N19506).
3rd CD Consideration In accordance with Recommendation 14 (see SC 27 N19523) of the 58th SC 27/WG 3 meeting / CRM held in Tel Aviv, Israel, 2019-04-01/05 the hereby attached document is circulated for a 8-week 3rd CD letter ballot closing by <b style="text-align: center;">2019-09-06 Medium: http://isotc.iso.org/livelink/livelink/open/jtc1sc27 No. of pages: 2 + 283			

Explanatory Report (2nd page)			
Status	SC 27 Decision	Reference documents	
		Input	Output
Study Period IT security testing, evaluation and assurance standards and techniques	51 st WG 3 meeting, Oct. 2015, Recommendations 5, 6 (N15594 = WG 3 N1251).		Terms of Reference (WG 5 N1258); 1 st /2 nd call f. contr. (WG 3 N1259 /1317)..
	52 nd WG 3 meeting, April 2016, Recommendation 5, 7 (N16026 = WG 3 N1296).	Expert contr. (WG 3 N1299, 1301).	3 rd call f. contr. (WG 3 N1377); Rapporteur's report (WG 3 N1320); Liaison to: CCDB (WG 3 N1266).
ISO/IEC NP 15408-2 (revision) Evaluation criteria for IT security -- Part 2 NWIP	53 rd WG 3 meeting, Oct. 2016, Recommendations 6, 15 (N16800 = WG 5 N600).	Expert contr. (WG 3 N1368, N1371, N1373).	SP report (WG 3 N1363); Call f. editor (WG 3 N1387 = N16886); Liaisons to: CCDB (WG 3 N1330); The Open Group (WG 3 N1332); Text f. NWIP (N16964 [replaces N16883]).

1
2
3
4
5
6
7
8
9

10
11
12
13
14
15
16
17
18
19

ISO/IEC JTC 1/SC 27 N19506

Date: 2019-07-12

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

ISO/IEC JTC 1/SC 27 IT Security techniques

Secretariat: DIN

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

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

CD stage

Warning for WDs and CDs

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and **may** not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

20
21
22
23
24
25
26
27
28
29
30
31
32

READ ME FIRST

Editors general notes for this draft.
Red text in a box are the Editors comments.
In this draft the editors highlighted the keywords relating to the ISO verbal forms, shall, should, may, can and must using green text in order to highlight these words. This convention will be removed before the FDIS level documents.
The editors are aware that the figures are of low quality. In the final documents high quality images will be used. The Editors hope that they are legible in this draft
The Editors thank the WG 3 contributors for their contributions and support during the editing cycle.

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

33	Contents	Page
34	Foreword.....	xxi
35	Introduction.....	xxii
36	1 Scope	1
37	2 Normative references	1
38	3 Terms and Definitions.....	1
39	4 Overview	1
40	4.1 General.....	1
41	4.2 Organization of this document.....	2
42	5 Functional requirements paradigm	2
43	6 Security functional components.....	6
44	6.1 Overview.....	6
45	6.1.1 General.....	6
46	6.1.2 Class structure	6
47	6.1.3 Family structure	7
48	6.1.4 Component structure.....	9
49	6.2 Component catalogue	10
50	6.2.1 General.....	10
51	6.2.2 Component changes highlighting	11
52	7 Class FAU: Security audit	12
53	7.1 Class description	12
54	7.2 Security audit automatic response (FAU_ARP).....	12
55	7.2.1 Family behaviour	12
56	7.2.2 Components leveling and description.....	12
57	7.2.3 Management of FAU_ARP.1.....	13
58	7.2.4 Audit of FAU_ARP.1.....	13
59	7.2.5 FAU_ARP.1 Security alarms	13
60	7.3 Security audit data generation (FAU_GEN).....	13
61	7.3.1 Family behaviour	13
62	7.3.2 Components leveling and description.....	13
63	7.3.3 Management of FAU_GEN.1, FAU_GEN.2.....	14
64	7.3.4 Audit of FAU_GEN.1, FAU_GEN.2	14
65	7.3.5 FAU_GEN.1 Audit data generation.....	14
66	7.3.6 FAU_GEN.2 User identity association	14
67	7.4 Security audit analysis (FAU_SAA).....	14

68	7.4.1	Family behaviour	14
69	7.4.2	Components leveling and description.....	15
70	7.4.3	Management of FAU_SAA.1	15
71	7.4.4	Management of FAU_SAA.2	15
72	7.4.5	Management of FAU_SAA.3	15
73	7.4.6	Management of FAU_SAA.4	15
74	7.4.7	Audit of FAU_SAA.1, FAU_SAA.2, FAU_SAA.3, FAU_SAA.4	16
75	7.4.8	FAU_SAA.1 Potential violation analysis.....	16
76	7.4.9	FAU_SAA.2 Profile based anomaly detection.....	16
77	7.4.10	FAU_SAA.3 Simple attack heuristics.....	17
78	7.4.11	FAU_SAA.4 Complex attack heuristics.....	17
79	7.5	<i>Security audit review (FAU_SAR)</i>	17
80	7.5.1	Family behaviour	17
81	7.5.2	Components leveling and description.....	18
82	7.5.3	Management of FAU_SAR.1	18
83	7.5.4	Management of FAU_SAR.2, FAU_SAR.3	18
84	7.5.5	Audit of FAU_SAR.1	18
85	7.5.6	Audit of FAU_SAR.2	18
86	7.5.7	Audit of FAU_SAR.3	18
87	7.5.8	FAU_SAR.1 Audit review.....	18
88	7.5.9	FAU_SAR.2 Restricted audit review.....	19
89	7.5.10	FAU_SAR.3 Selectable audit review	19
90	7.6	<i>Security audit event selection (FAU_SEL)</i>	19
91	7.6.1	Family behaviour	19
92	7.6.2	Components leveling and description.....	19
93	7.6.3	Management of FAU_SEL.1.....	19
94	7.6.4	Audit of FAU_SEL.1	20
95	7.6.5	FAU_SEL.1 Selective audit.....	20
96	7.7	<i>Security audit data storage (FAU_STG)</i>	21
97	7.7.1	Family behaviour	21
98	7.7.2	Components leveling and description.....	21
99	7.7.3	Management of FAU_STG.1	21
100	7.7.4	Management of FAU_STG.2	21
101	7.7.5	Management of FAU_STG.3	21
102	7.7.6	Management of FAU_STG.4	21
103	7.7.7	Management of FAU_STG.5	22
104	7.7.8	Audit of FAU_STG.1.....	22
105	7.7.9	Audit of FAU_STG.2, FAU_STG.3.....	22

106	7.7.10	Audit of FAU_STG.4.....	22
107	7.7.11	Audit of FAU_STG.5.....	22
108	7.7.12	FAU_STG.1 Audit data storage location	22
109	7.7.13	FAU_STG.2 Protected audit data storage	22
110	7.7.14	FAU_STG.3 Guarantees of audit data availability	23
111	7.7.15	FAU_STG.4 Action in case of possible audit data loss.....	23
112	7.7.16	FAU_STG.5 Prevention of audit data loss.....	23
113	8	Class FCO: Communication.....	24
114	8.1	<i>Class description</i>	24
115	8.2	<i>Non-repudiation of origin (FCO_NRO)</i>	24
116	8.2.1	Family behaviour	24
117	8.2.2	Components leveling and description.....	24
118	8.2.3	Management of FCO_NRO.1, FCO_NRO.2	24
119	8.2.4	Audit of FCO_NRO.1	25
120	8.2.5	Audit of FCO_NRO.2	25
121	8.2.6	FCO_NRO.1 Selective proof of origin.....	25
122	8.2.7	FCO_NRO.2 Enforced proof of origin.....	25
123	8.3	<i>Non-repudiation of receipt (FCO_NRR)</i>	26
124	8.3.1	Family behaviour	26
125	8.3.2	Components leveling and description.....	26
126	8.3.3	Management of FCO_NRR.1, FCO_NRR.2.....	26
127	8.3.4	Audit of FCO_NRR.1.....	26
128	8.3.5	Audit of FCO_NRR.2.....	26
129	8.3.6	FCO_NRR.1 Selective proof of receipt.....	27
130	8.3.7	FCO_NRR.2 Enforced proof of receipt.....	27
131	9	Class FCS: Cryptographic support.....	28
132	9.1	<i>Class description</i>	28
133	9.2	<i>Cryptographic key management (FCS_CKM)</i>	28
134	9.2.1	Family behaviour	28
135	9.2.2	Components leveling and description.....	29
136	9.2.3	Management of FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.5, CKM.6.....	29
137	9.2.4	Audit of FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.5, CKM.6.....	29
138	9.2.5	FCS_CKM.1 Cryptographic key generation.....	29
139	9.2.6	FCS_CKM.2 Cryptographic key distribution	30
140	9.2.7	FCS_CKM.3 Cryptographic key access	30
141	9.2.8	FCS_CKM.4 Cryptographic key destruction.....	31
142	9.2.9	FCS_CKM.5 Cryptographic key derivation.....	31

143	9.2.10	FCS_CKM.6 Timing and event of cryptographic key destruction	31
144	9.3	<i>Cryptographic operation (FCS_COP)</i>	31
145	9.3.1	Family behaviour	31
146	9.3.2	Components leveling and description.....	32
147	9.3.3	Management of FCS_COP.1	32
148	9.3.4	Audit of FCS_COP.1	32
149	9.3.5	FCS_COP.1 Cryptographic operation.....	32
150	9.4	<i>Random bit generation (FCS_RBG)</i>	32
151	9.4.1	Family behaviour	32
152	9.4.2	Components leveling and description.....	33
153	9.4.3	Management of FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5,	
154		FCS_RBG.6.....	33
155	9.4.4	Audit of FCS_RBG.1, FCS_RBG.2	33
156	9.4.5	Audit of FCS_RBG.3, FCS_RBG.4, FCS_RBG.6, FCS_RBG.6.....	33
157	9.4.6	FCS_RBG.1 Random bit generation (RBG).....	34
158	9.4.7	FCS_RBG.2 Random bit generation (external seeding)	34
159	9.4.8	FCS_RBG.3 Random bit generation (internal seeding – single source).....	35
160	9.4.9	FCS_RBG.4 Random bit generation (internal seeding – multiple sources).....	35
161	9.4.10	FCS_RBG.5 Random bit generation (combining entropy sources).....	35
162	9.4.11	FCS_RBG.6 Random bit generation service.....	35
163	9.5	<i>Generation of random numbers (FCS_RNG)</i>	36
164	9.5.1	Family behaviour	36
165	9.5.2	Components leveling and description.....	36
166	9.5.3	Management of FCS_RNG.1	36
167	9.5.4	Audit of FCS_RNG.1.....	36
168	9.5.5	FCS_RNG.1 Random number generation	36
169	10	Class FDP: User data protection	37
170	10.1	<i>Class description</i>	37
171	10.2	<i>Access control policy (FDP_ACC)</i>	38
172	10.2.1	Family behaviour	38
173	10.2.2	Components leveling and description.....	39
174	10.2.3	Management of FDP_ACC.1, FDP_ACC.2	39
175	10.2.4	Audit of FDP_ACC.1, FDP_ACC.2.....	39
176	10.2.5	FDP_ACC.1 Subset access control	39
177	10.2.6	FDP_ACC.2 Complete access control	39
178	10.3	<i>Access control functions (FDP_ACF)</i>	40
179	10.3.1	Family behaviour	40
180	10.3.2	Components leveling and description.....	40

181	10.3.3	Management of FDP_ACF.1	40
182	10.3.4	Audit of FDP_ACF.1.....	40
183	10.3.5	FDP_ACF.1 Security attribute-based access control	40
184	10.4	<i>Data authentication (FDP_DAU)</i>	41
185	10.4.1	Family behaviour	41
186	10.4.2	Components leveling and description.....	41
187	10.4.3	Management of FDP_DAU.1, FDP_DAU.2	41
188	10.4.4	Audit of FDP_DAU.1.....	42
189	10.4.5	Audit of FDP_DAU.2.....	42
190	10.4.6	FDP_DAU.1 Basic Data Authentication.....	42
191	10.4.7	FDP_DAU.2 Data Authentication with Identity of Guarantor	42
192	10.5	<i>Export from the TOE (FDP_ETC)</i>	43
193	10.5.1	Family behaviour	43
194	10.5.2	Components leveling and description.....	43
195	10.5.3	Management of FDP_ETC.1	43
196	10.5.4	Management of FDP_ETC.2	43
197	10.5.5	Audit of FDP_ETC.1, FDP_ETC.2	43
198	10.5.6	FDP_ETC.1 Export of user data without security attributes.....	43
199	10.5.7	FDP_ETC.2 Export of user data with security attributes	44
200	10.6	<i>Information flow control policy (FDP_IFC)</i>	44
201	10.6.1	Family behaviour	44
202	10.6.2	Components leveling and description.....	45
203	10.6.3	Management of FDP_IFC.1, FDP_IFC.2.....	45
204	10.6.4	Audit of FDP_IFC.1, FDP_IFC.2	45
205	10.6.5	FDP_IFC.1 Subset information flow control	45
206	10.6.6	FDP_IFC.2 Complete information flow control.....	45
207	10.7	<i>Information flow control functions (FDP_IFF)</i>	46
208	10.7.1	Family behaviour	46
209	10.7.2	Components leveling and description.....	46
210	10.7.3	Management of FDP_IFF.1, FDP_IFF.2	46
211	10.7.4	Management of FDP_IFF.3, FDP_IFF.4, FDP_IFF.5.....	46
212	10.7.5	Management of FDP_IFF.6.....	47
213	10.7.6	Audit of FDP_IFF.1, FDP_IFF.2, FDP_IFF.5	47
214	10.7.7	Audit of FDP_IFF.3, FDP_IFF.4, FDP_IFF.6	47
215	10.7.8	FDP_IFF.1 Simple security attributes	47
216	10.7.9	FDP_IFF.2 Hierarchical security attributes.....	48
217	10.7.10	FDP_IFF.3 Limited illicit information flows.....	49
218	10.7.11	FDP_IFF.4 Partial elimination of illicit information flows	49

219	10.7.12	FDP_IFF.5 No illicit information flows.....	49
220	10.7.13	FDP_IFF.6 Illicit information flow monitoring.....	49
221	10.8	<i>Information Retention Control (FDP_IRC).....</i>	<i>50</i>
222	10.8.1	Family behaviour	50
223	10.8.2	Components leveling and description.....	50
224	10.8.3	Management of FDP_IRC.1	51
225	10.8.4	Audit of FDP_IRC.1.....	51
226	10.8.5	FDP_IRC.1 Information retention control.....	51
227	10.9	<i>Import from outside of the TOE (FDP_ITC).....</i>	<i>51</i>
228	10.9.1	Family behaviour	51
229	10.9.2	Components leveling and description.....	51
230	10.9.3	Management of FDP_ITC.1, FDP_ITC.2.....	52
231	10.9.4	Audit of FDP_ITC.1, FDP_ITC.2	52
232	10.9.5	FDP_ITC.1 Import of user data without security attributes.....	52
233	10.9.6	FDP_ITC.2 Import of user data with security attributes.....	52
234	10.10	<i>Internal TOE transfer (FDP_ITT).....</i>	<i>53</i>
235	10.10.1	Family behaviour	53
236	10.10.2	Components leveling and description.....	53
237	10.10.3	Management of FDP_ITT.1, FDP_ITT.2.....	54
238	10.10.4	Management of FDP_ITT.3, FDP_ITT.4.....	54
239	10.10.5	Audit of FDP_ITT.1, FDP_ITT.2	54
240	10.10.6	Audit of FDP_ITT.3, FDP_ITT.4.....	54
241	10.10.7	FDP_ITT.1 Basic internal transfer protection.....	54
242	10.10.8	FDP_ITT.2 Transmission separation by attribute.....	54
243	10.10.9	FDP_ITT.3 Integrity monitoring.....	55
244	10.10.10	FDP_ITT.4 Attribute-based integrity monitoring.....	55
245	10.11	<i>Residual information protection (FDP_RIP).....</i>	<i>56</i>
246	10.11.1	Family behaviour	56
247	10.11.2	Components leveling and description.....	56
248	10.11.3	Management of FDP_RIP.1, FDP_RIP.2.....	56
249	10.11.4	Audit of FDP_RIP.1, FDP_RIP.2.....	56
250	10.11.5	FDP_RIP.1 Subset residual information protection.....	56
251	10.11.6	FDP_RIP.2 Full residual information protection.....	56
252	10.12	<i>Rollback (FDP_ROL).....</i>	<i>57</i>
253	10.12.1	Family behaviour	57
254	10.12.2	Components leveling and description.....	57
255	10.12.3	Management of FDP_ROL.1, FDP_ROL.2.....	57
256	10.12.4	Audit of FDP_ROL.1, FDP_ROL.2.....	57

257	10.12.5	FDP_ROL.1 Basic rollback.....	57
258	10.12.6	FDP_ROL.2 Advanced rollback.....	58
259	10.13	<i>Stored data confidentiality (FDP_SDC).....</i>	<i>58</i>
260	10.13.1	Family behaviour	58
261	10.13.2	Components leveling and description.....	58
262	10.13.3	Management of FDP_SDC.1, FDP_SDC.2.....	58
263	10.13.4	Audit of FDP_SDC.1, FDP_SDC.2.....	59
264	10.13.5	FDP_SDC.1 Stored data confidentiality.....	59
265	10.13.6	FDP_SDC.2 Stored data confidentiality with dedicated method.....	59
266	10.14	<i>Stored data integrity (FDP_SDI).....</i>	<i>59</i>
267	10.14.1	Family behaviour	59
268	10.14.2	Components leveling and description.....	59
269	10.14.3	Management of FDP_SDI.1	60
270	10.14.4	Management of FDP_SDI.2	60
271	10.14.5	Audit of FDP_SDI.1.....	60
272	10.14.6	Audit of FDP_SDI.2.....	60
273	10.14.7	FDP_SDI.1 Stored data integrity monitoring	60
274	10.14.8	FDP_SDI.2 Stored data integrity monitoring and action.....	60
275	10.15	<i>Inter-TSF user data confidentiality transfer protection (FDP_UCT).....</i>	<i>61</i>
276	10.15.1	Family behaviour	61
277	10.15.2	Components leveling and description.....	61
278	10.15.3	Management of FDP_UCT.1.....	61
279	10.15.4	Audit of FDP_UCT.1	61
280	10.15.5	FDP_UCT.1 Basic data exchange confidentiality.....	61
281	10.16	<i>Inter-TSF user data integrity transfer protection (FDP_UIT).....</i>	<i>62</i>
282	10.16.1	Family behaviour	62
283	10.16.2	Components leveling and description.....	62
284	10.16.3	Management of FDP_UIT.1, FDP_UIT.2, FDP_UIT.3	62
285	10.16.4	Audit of FDP_UIT.1	62
286	10.16.5	Audit of FDP_UIT.2, FDP_UIT.3	63
287	10.16.6	FDP_UIT.1 Data exchange integrity	63
288	10.16.7	FDP_UIT.2 Source data exchange recovery	63
289	10.16.8	FDP_UIT.3 Destination data exchange recovery.....	64
290	11	Class FIA: Identification and authentication.....	65
291	11.1	<i>Class description</i>	<i>65</i>
292	11.2	<i>Authentication failures (FIA_AFL).....</i>	<i>66</i>
293	11.2.1	Family behaviour	66

294	11.2.2	Components leveling and description.....	66
295	11.2.3	Management of FIA_AFL.1.....	66
296	11.2.4	Audit of FIA_AFL.1.....	66
297	11.2.5	FIA_AFL.1 Authentication failure handling.....	66
298	11.3	<i>Authentication proof of identity (FIA_API).....</i>	<i>67</i>
299	11.3.1	Family behaviour.....	67
300	11.3.2	Components leveling and description.....	67
301	11.3.3	Management of FIA_API.1.....	67
302	11.3.4	Audit of FIA_API.1.....	67
303	11.3.5	FIA_API.1 Authentication proof of identity.....	67
304	11.4	<i>User attribute definition (FIA_ATD).....</i>	<i>67</i>
305	11.4.1	Family behaviour.....	67
306	11.4.2	Components leveling and description.....	68
307	11.4.3	Management of FIA_ATD.1.....	68
308	11.4.4	Audit of FIA_ATD.1.....	68
309	11.4.5	FIA_ATD.1 User attribute definition.....	68
310	11.5	<i>Specification of secrets (FIA_SOS).....</i>	<i>68</i>
311	11.5.1	Family behaviour.....	68
312	11.5.2	Components leveling and description.....	68
313	11.5.3	Management of FIA_SOS.1.....	69
314	11.5.4	Management of FIA_SOS.2.....	69
315	11.5.5	Audit of FIA_SOS.1, FIA_SOS.2.....	69
316	11.5.6	FIA_SOS.1 Verification of secrets.....	69
317	11.5.7	FIA_SOS.2 TSF Generation of secrets.....	69
318	11.6	<i>User authentication (FIA_UAU).....</i>	<i>69</i>
319	11.6.1	Family behaviour.....	69
320	11.6.2	Components leveling and description.....	70
321	11.6.3	Management of FIA_UAU.1.....	70
322	11.6.4	Management of FIA_UAU.2.....	70
323	11.6.5	Management of FIA_UAU.3, FIA_UAU.4, FIA_UAU.7.....	70
324	11.6.6	Management of FIA_UAU.5.....	71
325	11.6.7	Management of FIA_UAU.6.....	71
326	11.6.8	Management of FIA_UAU.7.....	71
327	11.6.9	Audit of FIA_UAU.1.....	71
328	11.6.10	Audit of FIA_UAU.2.....	71
329	11.6.11	Audit of FIA_UAU.3.....	71
330	11.6.12	Audit of FIA_UAU.4.....	71
331	11.6.13	Audit of FIA_UAU.5.....	71

332	11.6.14	Audit of FIA_UAU.6.....	71
333	11.6.15	Audit of FIA_UAU.7.....	72
334	11.6.16	FIA_UAU.1 Timing of authentication.....	72
335	11.6.17	FIA_UAU.2 User authentication before any action.....	72
336	11.6.18	FIA_UAU.3 Unforgeable authentication.....	72
337	11.6.19	FIA_UAU.4 Single-use authentication mechanisms.....	73
338	11.6.20	FIA_UAU.5 Multiple authentication mechanisms.....	73
339	11.6.21	FIA_UAU.6 Re-authenticating.....	73
340	11.6.22	FIA_UAU.7 Protected authentication feedback.....	73
341	11.7	<i>User identification (FIA_UID)</i>	74
342	11.7.1	Family behaviour.....	74
343	11.7.2	Components leveling and description.....	74
344	11.7.3	Management of FIA_UID.1.....	74
345	11.7.4	Management of FIA_UID.2.....	74
346	11.7.5	Audit of FIA_UID.1, FIA_UID.2.....	74
347	11.7.6	FIA_UID.1 Timing of identification.....	74
348	11.7.7	FIA_UID.2 User identification before any action.....	75
349	11.8	<i>User-subject binding (FIA_USB)</i>	75
350	11.8.1	Family behaviour.....	75
351	11.8.2	Components leveling and description.....	75
352	11.8.3	Management of FIA_USB.1.....	75
353	11.8.4	Audit of FIA_USB.1.....	75
354	11.8.5	FIA_USB.1 User-subject binding.....	75
355	12	Class FMT: Security management.....	77
356	12.1	<i>Class description</i>	77
357	12.2	<i>Limited capabilities and availability (FMT_LIM)</i>	78
358	12.2.1	Family behaviour.....	78
359	12.2.2	Components leveling and description.....	78
360	12.2.3	Management of FMT_LIM.1, FMT_LIM.2.....	78
361	12.2.4	Audit of FMT_LIM.1.....	78
362	12.2.5	FMT_LIM.1 Limited capabilities.....	78
363	12.2.6	FMT_LIM.2 Limited availability.....	78
364	12.3	<i>Management of functions in TSF (FMT_MOF)</i>	79
365	12.3.1	Family behaviour.....	79
366	12.3.2	Components leveling and description.....	79
367	12.3.3	Management of FMT_MOF.1.....	79
368	12.3.4	Audit of FMT_MOF.1.....	79

369	12.3.5	FMT_MOF.1 Management of security functions behaviour.....	79
370	12.4	<i>Management of security attributes (FMT_MSA).....</i>	<i>79</i>
371	12.4.1	Family behaviour	79
372	12.4.2	Components leveling and description.....	80
373	12.4.3	Management of FMT_MSA.1	80
374	12.4.4	Management of FMT_MSA.2	80
375	12.4.5	Management of FMT_MSA.3	80
376	12.4.6	Management of FMT_MSA.4	80
377	12.4.7	Audit of FMT_MSA.1.....	80
378	12.4.8	Audit of FMT_MSA.2.....	81
379	12.4.9	Audit of FMT_MSA.3.....	81
380	12.4.10	Audit of FMT_MSA.4.....	81
381	12.4.11	FMT_MSA.1 Management of security attributes.....	81
382	12.4.12	FMT_MSA.2 Secure security attributes.....	81
383	12.4.13	FMT_MSA.3 Static attribute initialization.....	82
384	12.4.14	FMT_MSA.4 Security attribute value inheritance.....	82
385	12.5	<i>Management of TSF data (FMT_MTD).....</i>	<i>82</i>
386	12.5.1	Family behaviour	82
387	12.5.2	Components leveling and description.....	82
388	12.5.3	Management of FMT_MTD.1.....	83
389	12.5.4	Management of FMT_MTD.2.....	83
390	12.5.5	Management of FMT_MTD.3.....	83
391	12.5.6	Audit of FMT_MTD.1	83
392	12.5.7	Audit of FMT_MTD.2	83
393	12.5.8	Audit of FMT_MTD.3.....	83
394	12.5.9	FMT_MTD.1 Management of TSF data.....	83
395	12.5.10	FMT_MTD.2 Management of limits on TSF data	83
396	12.5.11	FMT_MTD.3 Secure TSF data	84
397	12.6	<i>Revocation (FMT_REV).....</i>	<i>84</i>
398	12.6.1	Family behaviour	84
399	12.6.2	Components leveling and description.....	84
400	12.6.3	Management of FMT_REV.1	84
401	12.6.4	Audit of FMT_REV.1	84
402	12.6.5	FMT_REV.1 Revocation	85
403	12.7	<i>Security attribute expiration (FMT_SAE).....</i>	<i>85</i>
404	12.7.1	Family behaviour	85
405	12.7.2	Components leveling and description.....	85
406	12.7.3	Management of FMT_SAE.1.....	85

407	12.7.4	Audit of FMT_SAE.1.....	85
408	12.7.5	FMT_SAE.1 Time-limited authorization.....	85
409	12.8	<i>Specification of Management Functions (FMT_SMF)</i>	86
410	12.8.1	Family behaviour.....	86
411	12.8.2	Components leveling and description.....	86
412	12.8.3	Management of FMT_SMF.1.....	86
413	12.8.4	Audit of FMT_SMF.1.....	86
414	12.8.5	FMT_SMF.1 Specification of Management Functions.....	86
415	12.9	<i>Security management roles (FMT_SMR)</i>	87
416	12.9.1	Family behaviour.....	87
417	12.9.2	Components leveling and description.....	87
418	12.9.3	Management of FMT_SMR.1.....	87
419	12.9.4	Management of FMT_SMR.2.....	87
420	12.9.5	Management of FMT_SMR.3.....	87
421	12.9.6	Audit of FMT_SMR.1.....	87
422	12.9.7	Audit of FMT_SMR.2.....	87
423	12.9.8	Audit of FMT_SMR.3.....	88
424	12.9.9	FMT_SMR.1 Security roles.....	88
425	12.9.10	FMT_SMR.2 Restrictions on security roles.....	88
426	12.9.11	FMT_SMR.3 Assuming roles.....	88
427	13	Class FPR: Privacy	89
428	13.1	<i>Class description</i>	89
429	13.2	<i>Anonymity (FPR_ANO)</i>	89
430	13.2.1	Family behaviour.....	89
431	13.2.2	Components leveling and description.....	89
432	13.2.3	Management of FPR_ANO.1, FPR_ANO.2.....	90
433	13.2.4	Audit of FPR_ANO.1, FPR_ANO.2.....	90
434	13.2.5	FPR_ANO.1 Anonymity.....	90
435	13.2.6	FPR_ANO.2 Anonymity without soliciting information.....	90
436	13.3	<i>Pseudonymity (FPR_PSE)</i>	90
437	13.3.1	Family behaviour.....	90
438	13.3.2	Components leveling and description.....	90
439	13.3.3	Management of FPR_PSE.1, FPR_PSE.2, FPR_PSE.3.....	91
440	13.3.4	Audit of FPR_PSE.1, FPR_PSE.2, FPR_PSE.3.....	91
441	13.3.5	FPR_PSE.1 Pseudonymity.....	91
442	13.3.6	FPR_PSE.2 Reversible pseudonymity.....	91
443	13.3.7	FPR_PSE.3 Alias pseudonymity.....	92

444	13.4	<i>Unlinkability (FPR_UNL)</i>	92
445	13.4.1	Family behaviour	92
446	13.4.2	Components leveling and description.....	92
447	13.4.3	Management of FPR_UNL.1.....	93
448	13.4.4	Audit of FPR_UNL.1	93
449	13.4.5	FPR_UNL.1 Unlinkability of operations	93
450	13.5	<i>Unobservability (FPR_UNO)</i>	93
451	13.5.1	Family behaviour	93
452	13.5.2	Components leveling and description.....	93
453	13.5.3	Management of FPR_UNO.1, FPR_UNO.2	94
454	13.5.4	Management of FPR_UNO.3	94
455	13.5.5	Management of FPR_UNO.4	94
456	13.5.6	Audit of FPR_UNO.1, FPR_UNO.2.....	94
457	13.5.7	Audit of FPR_UNO.3	94
458	13.5.8	Audit of FPR_UNO.4	94
459	13.5.9	FPR_UNO.1 Unobservability	94
460	13.5.10	FPR_UNO.2 Allocation of information impacting unobservability	94
461	13.5.11	FPR_UNO.3 Unobservability without soliciting information.....	95
462	13.5.12	FPR_UNO.4 Authorized user observability.....	95
463	14	Class FPT: Protection of the TSF	96
464	14.1	<i>Class description</i>	96
465	14.2	<i>TOE emanation (FPT_EMS)</i>	97
466	14.2.1	Family behaviour	97
467	14.2.2	Components leveling and description.....	98
468	14.2.3	Management of FPT_EMS.1.....	98
469	14.2.4	Audit of FPT_EMS.1	98
470	14.2.5	FPT_EMS.1 Emanation of TSF and User data.....	98
471	14.3	<i>Fail secure (FPT_FLS)</i>	99
472	14.3.1	Family behaviour	99
473	14.3.2	Components leveling and description.....	99
474	14.3.3	Management of FPT_FLS.1	99
475	14.3.4	Audit of FPT_FLS.1.....	99
476	14.3.5	FPT_FLS.1 Failure with preservation of secure state.....	99
477	14.4	<i>TSF initialization (FPT_INI)</i>	99
478	14.4.1	Family behaviour	99
479	14.4.2	Components leveling and description.....	99
480	14.4.3	Management of FPT_INI.1.....	100

481	14.4.4	Audit of FPT_INI.1	100
482	14.4.5	FPT_INI.1 TSF initialization	100
483	14.5	<i>Availability of exported TSF data (FPT_ITA)</i>	100
484	14.5.1	Family behaviour	100
485	14.5.2	Components leveling and description.....	101
486	14.5.3	Management of FPT_ITA.1.....	101
487	14.5.4	Audit of FPT_ITA.1.....	101
488	14.5.5	FPT_ITA.1 Inter-TSF availability within a defined availability metric	101
489	14.6	<i>Confidentiality of exported TSF data (FPT_ITC)</i>	101
490	14.6.1	Family behaviour	101
491	14.6.2	Components leveling and description.....	101
492	14.6.3	Management of FPT_ITC.1.....	102
493	14.6.4	Audit of FPT_ITC.1	102
494	14.6.5	FPT_ITC.1 Inter-TSF confidentiality during transmission.....	102
495	14.7	<i>Integrity of exported TSF data (FPT_ITI)</i>	102
496	14.7.1	Family behaviour	102
497	14.7.2	Components leveling and description.....	102
498	14.7.3	Management of FPT_ITI.1	102
499	14.7.4	Management of FPT_ITI.2.....	102
500	14.7.5	Audit of FPT_ITI.1	103
501	14.7.6	Audit of FPT_ITI.2	103
502	14.7.7	FPT_ITI.1 Inter-TSF detection of modification	103
503	14.7.8	FPT_ITI.2 Inter-TSF detection and correction of modification	103
504	14.8	<i>Internal TOE TSF data transfer (FPT_ITT)</i>	104
505	14.8.1	Family behaviour	104
506	14.8.2	Components leveling and description.....	104
507	14.8.3	Management of FPT_ITT.1.....	104
508	14.8.4	Management of FPT_ITT.2.....	104
509	14.8.5	Management of FPT_ITT.3.....	104
510	14.8.6	Audit of FPT_ITT.1, FPT_ITT.2	105
511	14.8.7	Audit of FPT_ITT.3	105
512	14.8.8	FPT_ITT.1 Basic internal TSF data transfer protection.....	105
513	14.8.9	FPT_ITT.2 TSF data transfer separation	105
514	14.8.10	FPT_ITT.3 TSF data integrity monitoring.....	105
515	14.9	<i>TSF physical protection (FPT_PHP)</i>	106
516	14.9.1	Family behaviour	106
517	14.9.2	Components leveling and description.....	106
518	14.9.3	Management of FPT_PHP.1	106

519	14.9.4	Management of FPT_PHP.2	106
520	14.9.5	Management of FPT_PHP.3	106
521	14.9.6	Audit of FPT_PHP.1	107
522	14.9.7	Audit of FPT_PHP.2	107
523	14.9.8	Audit of FPT_PHP.3	107
524	14.9.9	FPT_PHP.1 Passive detection of physical attack.....	107
525	14.9.10	FPT_PHP.2 Notification of physical attack.....	107
526	14.9.11	FPT_PHP.3 Resistance to physical attack.....	108
527	14.10	<i>Trusted recovery (FPT_RCV)</i>	108
528	14.10.1	Family behaviour	108
529	14.10.2	Components leveling and description.....	108
530	14.10.3	Management of FPT_RCV.1	108
531	14.10.4	Management of FPT_RCV.2, FPT_RCV.3.....	108
532	14.10.5	Management of FPT_RCV.4	109
533	14.10.6	Audit of FPT_RCV.1, FPT_RCV.2, FPT_RCV.3	109
534	14.10.7	Audit of FPT_RCV.4.....	109
535	14.10.8	FPT_RCV.1 Manual recovery	109
536	14.10.9	FPT_RCV.2 Automated recovery	109
537	14.10.10	FPT_RCV.3 Automated recovery without undue loss.....	109
538	14.10.11	FPT_RCV.4 Function recovery	110
539	14.11	<i>Replay detection (FPT_RPL)</i>	110
540	14.11.1	Family behaviour	110
541	14.11.2	Components leveling and description.....	110
542	14.11.3	Management of FPT_RPL.1.....	110
543	14.11.4	Audit of FPT_RPL.1.....	111
544	14.11.5	FPT_RPL.1 Replay detection.....	111
545	14.12	<i>State synchrony protocol (FPT_SSP)</i>	111
546	14.12.1	Family behaviour	111
547	14.12.2	Components leveling and description.....	111
548	14.12.3	Management of FPT_SSP.1, FPT_SSP.2.....	111
549	14.12.4	Audit of FPT_SSP.1, FPT_SSP.2	112
550	14.12.5	FPT_SSP.1 Simple trusted acknowledgement.....	112
551	14.12.6	FPT_SSP.2 Mutual trusted acknowledgement.....	112
552	14.13	<i>Time stamps (FPT_STM)</i>	112
553	14.13.1	Family behaviour	112
554	14.13.2	Components leveling and description.....	112
555	14.13.3	Management of FPT_STM.1.....	112
556	14.13.4	Management of FPT_STM.2.....	113

557	14.13.5	Audit of FPT_STM.1	113
558	14.13.6	Audit of FPT_STM.2	113
559	14.13.7	FPT_STM.1 Reliable time stamps.....	113
560	14.13.8	FPT_STM.2 Time source	113
561	14.14	<i>Inter-TSF TSF data consistency (FPT_TDC)</i>	113
562	14.14.1	Family behaviour	113
563	14.14.2	Components leveling and description.....	113
564	14.14.3	Management of FPT_TDC.1	114
565	14.14.4	Audit of FPT_TDC.1	114
566	14.14.5	FPT_TDC.1 Inter-TSF basic TSF data consistency	114
567	14.15	<i>Testing of external entities (FPT_TEE)</i>	114
568	14.15.1	Family behaviour	114
569	14.15.2	Components leveling and description.....	114
570	14.15.3	Management of FPT_TEE.1	115
571	14.15.4	Audit of FPT_TEE.1.....	115
572	14.15.5	FPT_TEE.1 Testing of external entities.....	115
573	14.16	<i>Internal TOE TSF data replication consistency (FPT_TRC)</i>	115
574	14.16.1	Family behaviour	115
575	14.16.2	Components leveling and description.....	115
576	14.16.3	Management of FPT_TRC.1	116
577	14.16.4	Audit of FPT_TRC.1.....	116
578	14.16.5	FPT_TRC.1 Internal TSF consistency	116
579	14.17	<i>TSF self-test (FPT_TST)</i>	116
580	14.17.1	Family behaviour	116
581	14.17.2	Components leveling and description.....	116
582	14.17.3	Management of FPT_TST.1.....	117
583	14.17.4	Audit of FPT_TST.1	117
584	14.17.5	FPT_TST.1 TSF self-testing.....	117
585	15	Class FRU: Resource utilization	118
586	15.1	<i>Class description</i>	118
587	15.2	<i>Fault tolerance (FRU_FLT)</i>	118
588	15.2.1	Family behaviour	118
589	15.2.2	Components leveling and description.....	118
590	15.2.3	Management of FRU_FLT.1, FRU_FLT.2.....	118
591	15.2.4	Audit of FRU_FLT.1.....	119
592	15.2.5	Audit of FRU_FLT.2.....	119
593	15.2.6	FRU_FLT.1 Degraded fault tolerance	119

594	15.2.7	FRU_FLT.2 Limited fault tolerance.....	119
595	15.3	<i>Priority of service (FRU_PRS)</i>	<i>119</i>
596	15.3.1	Family behaviour	119
597	15.3.2	Components leveling and description.....	119
598	15.3.3	Management of FRU_PRS.1, FRU_PRS.2.....	120
599	15.3.4	Audit of FRU_PRS.1, FRU_PRS.2.....	120
600	15.3.5	FRU_PRS.1 Limited priority of service	120
601	15.3.6	FRU_PRS.2 Full priority of service.....	120
602	15.4	<i>Resource allocation (FRU_RSA).....</i>	<i>120</i>
603	15.4.1	Family behaviour	120
604	15.4.2	Components leveling and description.....	121
605	15.4.3	Management of FRU_RSA.1.....	121
606	15.4.4	Management of FRU_RSA.2.....	121
607	15.4.5	Audit of FRU_RSA.1, FRU_RSA.2	121
608	15.4.6	FRU_RSA.1 Maximum quotas.....	121
609	15.4.7	FRU_RSA.2 Minimum and maximum quotas	121
610	16	Class FTA: TOE access.....	123
611	16.1	<i>Class description</i>	<i>123</i>
612	16.2	<i>Limitation on scope of selectable attributes (FTA_LSA).....</i>	<i>123</i>
613	16.2.1	Family behaviour	123
614	16.2.2	Components leveling and description.....	123
615	16.2.3	Management of FTA_LSA.1.....	124
616	16.2.4	Audit of FTA_LSA.1	124
617	16.2.5	FTA_LSA.1 Limitation on scope of selectable attributes.....	124
618	16.3	<i>Limitation on multiple concurrent sessions (FTA_MCS).....</i>	<i>124</i>
619	16.3.1	Family behaviour	124
620	16.3.2	Components leveling and description.....	124
621	16.3.3	Management of FTA_MCS.1.....	124
622	16.3.4	Management of FTA_MCS.2.....	125
623	16.3.5	Audit of FTA_MCS.1, FTA_MCS.2.....	125
624	16.3.6	FTA_MCS.1 Basic limitation on multiple concurrent sessions.....	125
625	16.3.7	FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions.....	125
626	16.4	<i>Session locking and termination (FTA_SSL).....</i>	<i>125</i>
627	16.4.1	Family behaviour	125
628	16.4.2	Components leveling and description.....	126
629	16.4.3	Management of FTA_SSL.1	126
630	16.4.4	Management of FTA_SSL.2	126

631	16.4.5	Management of FTA_SSL.3	126
632	16.4.6	Management of FTA_SSL.4	126
633	16.4.7	Audit of FTA_SSL.1, FTA_SSL.2	126
634	16.4.8	Audit of FTA_SSL.3.....	127
635	16.4.9	Audit of FTA_SSL.4.....	127
636	16.4.10	FTA_SSL.1 TSF-initiated session locking.....	127
637	16.4.11	FTA_SSL.2 User-initiated locking.....	127
638	16.4.12	FTA_SSL.3 TSF-initiated termination	128
639	16.4.13	FTA_SSL.4 User-initiated termination.....	128
640	16.5	<i>TOE access banners (FTA_TAB)</i>	128
641	16.5.1	Family behaviour	128
642	16.5.2	Components leveling and description.....	128
643	16.5.3	Management of FTA_TAB.1.....	128
644	16.5.4	Audit of FTA_TAB.1	128
645	16.5.5	FTA_TAB.1 Default TOE access banners	128
646	16.6	<i>TOE access history (FTA_TAH)</i>	129
647	16.6.1	Family behaviour	129
648	16.6.2	Components leveling and description.....	129
649	16.6.3	Management of FTA_TAH.1	129
650	16.6.4	Audit of FTA_TAH.1.....	129
651	16.6.5	FTA_TAH.1 TOE access history	129
652	16.7	<i>TOE session establishment (FTA_TSE)</i>	130
653	16.7.1	Family behaviour	130
654	16.7.2	Components leveling and description.....	130
655	16.7.3	Management of FTA_TSE.1.....	130
656	16.7.4	Audit of FTA_TSE.1.....	130
657	16.7.5	FTA_TSE.1 TOE session establishment.....	130
658	17	Class FTP: Trusted path/channels	131
659	17.1	<i>Class description</i>	131
660	17.2	<i>Inter-TSF trusted channel (FTP_ITC)</i>	132
661	17.2.1	Family behaviour	132
662	17.2.2	Components leveling and description.....	132
663	17.2.3	Management of FTP_ITC.1.....	132
664	17.2.4	Audit of FTP_ITC.1	132
665	17.2.5	FTP_ITC.1 Inter-TSF trusted channel.....	132
666	17.3	<i>Trusted channel protocol (FTP_PRO)</i>	133
667	17.3.1	Family behavior	133

668	17.3.2	Components leveling and description.....	133
669	17.3.3	Management of FTP_PRO.1	133
670	17.3.4	Management of FTP_PRO.2	133
671	17.3.5	Management of FTP_PRO.3	133
672	17.3.6	Audit of FTP_PRO.1	133
673	17.3.7	Audit of FTP_PRO.2	134
674	17.3.8	Audit of FTP_PRO.3	134
675	17.3.9	FTP_PRO.1 Trusted channel protocol.....	134
676	17.3.10	FTP_PRO.2 Trusted channel establishment.....	135
677	17.3.11	FTP_PRO.3 Trusted channel data protection.....	135
678	17.4	<i>Trusted path (FTP_TRP)</i>	<i>135</i>
679	17.4.1	Family behaviour	135
680	17.4.2	Components leveling and description.....	136
681	17.4.3	Management of FTP_TRP.1	136
682	17.4.4	Audit of FTP_TRP.1.....	136
683	17.4.5	FTP_TRP.1 Trusted path.....	136
684		Annex A (informative) Security functional requirements structure of the application notes	
685		137
686		Annex B (informative) Dependency tables for security functional components.....	140
687		Annex C (normative) Class FAU: Security audit - application notes	152
688		Annex D (normative) Class FCO: Communication- application notes	164
689		Annex E (normative) Class FCS: Cryptographic support- application notes	169
690		Annex F (normative) Class FDP: User data protection- application notes.....	178
691		Annex G (normative) Class FIA: Identification and authentication- application notes.....	203
692		Annex H (normative) Class FMT: Security management- application notes	212
693		Annex I (normative) Class FPR: Privacy- application notes.....	221
694		Annex J (normative) Class FPT: Protection of the TSF- application notes	232
695		Annex K (normative) Class FRU: Resource utilization- application notes.....	248
696		Annex L (normative) Class FTA: TOE access- application notes	252
697		Annex M (normative) Class FTP: Trusted path/channels- application notes	258
698			

699 **Foreword**

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

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

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

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

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

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

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

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

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

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

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

735

736 **Introduction**

737 Security functional components, as defined in this document, are the basis for the security
738 functional requirements or components expressed in a Protection Profile (PP), PP-Module,
739 functional package or a Security Target (ST). These requirements describe the desired security
740 behaviour expected of a Target of Evaluation (TOE) and are intended to meet the security
741 objectives as stated in a PP, PP-Module, functional package or an ST. These requirements describe
742 security properties that users **can** detect by direct interaction (i.e. inputs, outputs) with the IT or
743 by the IT response to stimulus.

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

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

- 751 a) Consumers, who use this document when selecting components to express functional
752 requirements which satisfy the security objectives expressed in a PP, PP-Module,
753 functional package or ST. ISO/IEC 15408-1:20XX, Clause 7 provides more detailed
754 information on the relationship between security objectives and security
755 requirements.
- 756 b) Developers, who respond to actual or perceived consumer security requirements in
757 constructing a TOE, **will** find a standardized method to understand those
758 requirements in this document. They **should** also use the contents of this document
759 as a basis for further defining the TOE security functionality and mechanisms that
760 comply with those requirements.
- 761 c) Evaluators, who use the security functional requirements defined in this document in
762 verifying that the TOE functional requirements expressed in the PP, PP-Module,
763 functional package or ST satisfy the IT security objectives and that all dependencies
764 are accounted for and shown to be satisfied. Evaluators **shall** use this document to
765 assist in determining whether a given TOE satisfies stated requirements.

766

767 IT Security techniques — Evaluation criteria for IT security 768 — Part 2: Security functional components

769 1 Scope

770 This document defines the required structure and content of security functional components
771 for the purpose of security evaluation. It includes a catalogue of functional components that will
772 meet the common security functionality requirements of many IT products.

773 2 Normative references

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

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

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

782 3 Terms and Definitions

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

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

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

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

789 4 Overview

790 4.1 General

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

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

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

805 4.2 Organization of this document

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

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

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

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

811 Annex C through Annex M provide the explanatory information for the functional classes. This
812 material **shall** be seen as normative instructions on how to apply relevant operations and select
813 appropriate audit or documentation information; the use of the auxiliary verb “**should**” means
814 that the instruction is strongly preferred, but others **may** be justifiable. Where different options
815 are given, the choice is left to the PP, PP-Module, functional package and ST author.

816 Those who author PPs, PP-Modules, functional packages, or STs **shall** refer to ISO/IEC 15408-
817 1:20XX for relevant structures, rules, and guidance, in particular:

- 818 a) ISO/IEC 15408-1:20XX, Clause 3 defines the terms and definitions used in ISO/IEC
819 15408 (all parts).
- 820 b) ISO/IEC 15408-1:20XX, Clause 7 describes how security functional requirements
821 can be specified using the security functional components.
- 822 c) ISO/IEC 15408-1:20XX, Clause 8 describes how security functional components are
823 organized, and the operations that may be applied to them.
- 824 d) ISO/IEC 15408-1:20XX, Annex A provides further guidance on the structure for
825 security functional packages.
- 826 e) ISO/IEC 15408-1:20XX, Annex B provides further guidance on the structure for
827 PPs.
- 828 f) ISO/IEC 15408-1:20XX, Annex C provides further guidance on the structure of PP-
829 Modules and PP-Configurations.
- 830 g) ISO/IEC 15408-1:20XX, Annex D provides further guidance on the structure for
831 STs.

832 5 Functional requirements paradigm

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

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

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

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

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

850 The TOE **may** be a monolithic product containing hardware, firmware, and software.
 851 Alternatively, a TOE **may** be a distributed product that consists internally of multiple separated
 852 parts. Each of these parts of the TOE provides a particular service for the TOE and is connected
 853 to the other parts of the TOE through an **internal communication channel**. This channel **can**
 854 be as small as a processor bus or **may** encompass a network internal to the TOE.

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

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

- 862 a) The SFRs of the other “trusted IT product” and the SFRs of the TOE have been
 863 administratively coordinated and the other trusted IT product is assumed to
 864 enforce its SFRs correctly (e. g. by being separately evaluated). Exchanges of
 865 information in this situation are called **inter-TSF transfers**, as they are between
 866 the TSFs of distinct trusted products.
- 867 b) The other IT product **may** not be trusted, it **may** be called an “untrusted IT
 868 product”. Therefore, its SFRs are either unknown or their implementation is not
 869 viewed as trustworthy. TSF mediated exchanges of information in this situation are
 870 called **transfers outside of the TOE**, as there is no TSF (or its policy characteristics
 871 are unknown) on the other IT product.

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

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

883 EXAMPLE 1

884 An example of a TOE device is a workstation.

885 A period of interaction between users and the TSF is referred to as a user **session**.
 886 Establishment of user sessions **can** be controlled based on a variety of considerations.

887 EXAMPLE 2

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

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

894 To express requirements that call for the separation of administrator duties, the relevant
 895 security functional components (from family FMT_SMR) explicitly state that administrative
 896 **roles** are required. A role is a pre-defined set of rules establishing the allowed interactions
 897 between a user operating in that role and the TOE. A TOE **may** support the definition of any
 898 number of roles.

899 EXAMPLE 3

900 Roles related to the secure operation of a TOE **may** include “Audit Administrator” and “User Accounts Administrator”.
901 TOEs contain **resources** that **may** be used for the processing and storing of information. The
902 primary goal of the TSF is the complete and correct enforcement of the SFRs over the resources
903 and information that the TOE controls.

904 TOE resources **can** be structured and utilized in many different ways. However, this document
905 makes a specific distinction that allows for the specification of desired security properties. All
906 entities that **can** be created from resources **can** be characterized in one of two ways. The
907 entities **may** be active, meaning that they are the cause of actions that occur internal to the TOE
908 and cause operations to be performed on information. Alternatively, the entities **may** be
909 passive, meaning that they are either the container from which information originates or to
910 which information is stored.

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

913 a) those acting on behalf of an authorized user;

914 EXAMPLE 4

915 UNIX processes

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

918 EXAMPLE 5

919 functions as might be found in client/server architectures

920 c) those acting as part of the TOE itself.

921 EXAMPLE 6

922 processes not acting on behalf of a user

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

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

928 EXAMPLE 7

929 An example of a subject is an inter-process communication

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

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

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

946 EXAMPLE 8

947 User data:

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

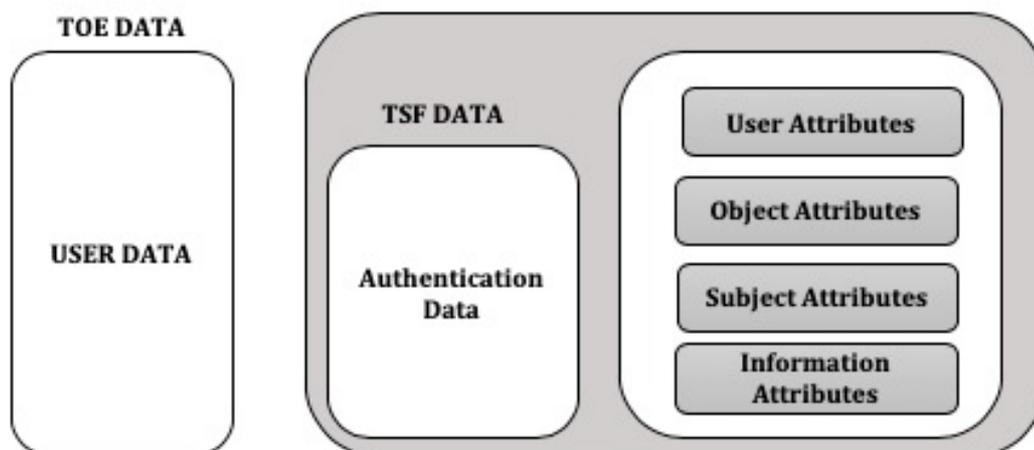
949 TSF data:

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

952 There are several SFPs that apply to data protection such as **access control SFPs** and
953 **information flow control SFPs**. The mechanisms that implement access control SFPs base
954 their policy decisions on attributes of the users, resources, subjects, objects, sessions, TSF status
955 data and operations within the scope of control. These attributes are used in the set of rules that
956 govern operations that subjects **may** perform on objects.

957 The mechanisms that implement information flow control SFPs base their policy decisions on
958 the attributes of the subjects and information within the scope of control and the set of rules
959 that govern the operations by subjects on information. The attributes of the information, which
960 **may** be associated with the attributes of the container or **may** be derived from the data in the
961 container, stay with the information as it is processed by the TSF.

962



963 **Figure 1 — Relationship between user data and TSF data**

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

966 Authentication data is used to verify the claimed identity of a user requesting services from a
967 TOE. The most common form of authentication data is the password, which depends on being
968 kept secret in order to be an effective security mechanism. However, not all forms of
969 authentication data need to be kept secret. Biometric authentication devices do not rely on the
970 fact that the data is kept secret, but rather that the data is something that only one user
971 possesses and that cannot be forged.

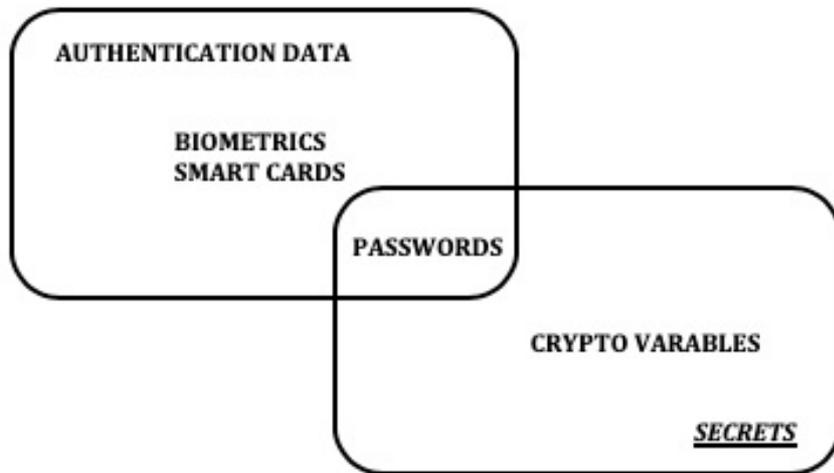
972 EXAMPLE 9

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

974 The term secrets, as used in this document, while applicable to authentication data, is intended
975 to also be applicable to other types of data that must be kept secret in order to enforce a specific
976 SFP.

977 Therefore, some, but not all, authentication data needs to be kept secret and some, but not all,
978 secrets are used as authentication data. Figure 2 shows this relationship between secrets and

979 authentication data. In the Figure, the types of data typically encountered in the authentication
 980 data and the secrets subclauses are indicated.



981
 982 **Figure 2 — Relationship between “authentication data” and “secrets”**

983 **6 Security functional components**

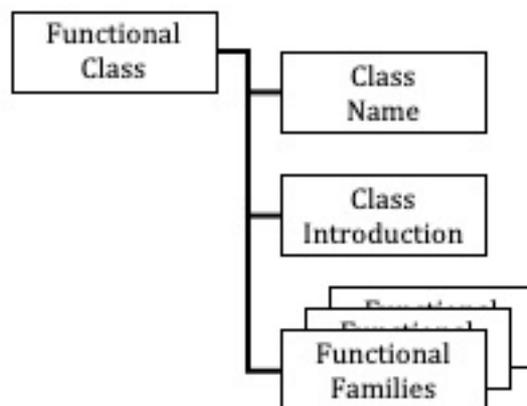
984 **6.1 Overview**

985 **6.1.1 General**

986 Clause 6 defines the content and presentation of the functional requirements of this document
 987 and provides guidance on the organization of the requirements for new, extended components
 988 that may be included in a PP, PP-Module, functional package or ST. As described in ISO/IEC
 989 15408-1 Clause 8, the functional components and requirements are expressed in classes,
 990 families, components and elements.

991 **6.1.2 Class structure**

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



NOTE
 A functional class may contain
 multiple Functional Families

994

Figure 3 — Functional class structure**995 Class name**

996 The class name subclause provides information necessary to identify and categorize a
 997 functional class. Every functional class has a unique name. The categorical information consists
 998 of a short name of three characters. The short name of the class is used in the specification of
 999 the short names of the families of that class.

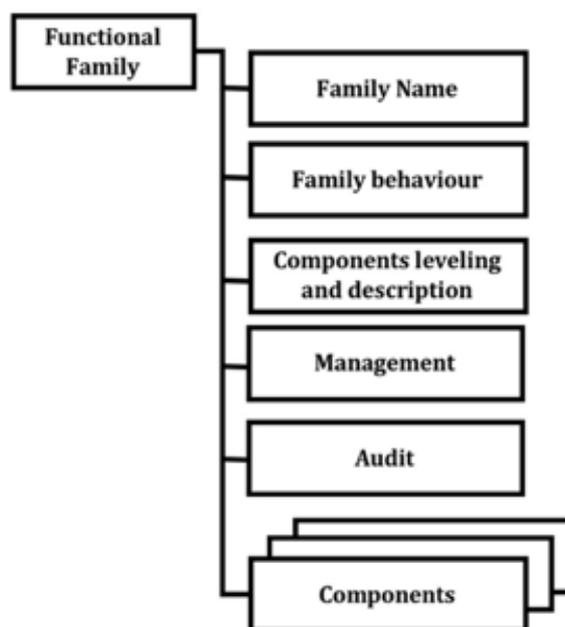
1000 Class introduction

1001 The class introduction expresses the common intent or approach of those families to satisfy
 1002 security objectives. The definition of functional classes does not reflect any formal taxonomy in
 1003 the specification of the requirements.

1004 The class introduction provides a figure describing the families in this class and the hierarchy of
 1005 the components in each family, as explained in 6.2.

1006 6.1.3 Family structure

1007 Figure 4 illustrates the functional family structure in diagrammatic form.



1008

1009

Figure 4 — Functional family structure**1010 Family name**

1011 The family name subclause provides categorical and descriptive information necessary to
 1012 identify and categorize a functional family. Every functional family has a unique name. The
 1013 categorical information consists of a short name of seven characters, with the first three
 1014 identical to the short name of the class followed by an underscore and the short name of the
 1015 family as follows, XXX_YYY. The unique short form of the family name provides the principal
 1016 reference name for the security components.

1017 Family behaviour

1018 The family behaviour is the narrative description of the functional family stating its security
 1019 objective and a general description of the functional requirements. These are described in
 1020 greater detail below:

- 1021 a) The security objectives of the family address a security problem that **may** be solved
 1022 with the help of a TOE that incorporates SFRs derived from a component of this
 1023 family;
- 1024 b) The description of the *functional requirements* summarizes all the requirements
 1025 that are included in the component(s). The description is aimed at authors of STs,
 1026 PPs, PP-Modules or security functional packages who wish to assess whether the
 1027 family is relevant to their specific requirements.

1028 **Components leveling and description**

1029 Functional families contain one or more components, any one of which **may** be selected for
 1030 inclusion in STs, PPs, PP-Modules or security functional packages. The goal of the Components
 1031 leveling and description subclause is to provide information to users in selecting an appropriate
 1032 functional component once the family has been identified as being a necessary or useful part of
 1033 their security requirements.

1034 The functional family description subclause describes the components available, and their
 1035 rationale. The exact details of the components are contained within each component.

1036 The relationships between components within a functional family **may** be hierarchical. A
 1037 component is hierarchical to another if it offers more security.

1038 As explained in 6.2 the descriptions of the families provide a graphical overview of the
 1039 hierarchy of the components in a family.

1040 **Management**

1041 The management clauses contain information for ST, PP, PP-Module, or security functional
 1042 package authors to consider as management activities for a given component. The clauses
 1043 reference components of the management class (FMT) and provide guidance regarding
 1044 potential management activities that **may** be applied via operations to those components.

1045 An author **may** select the indicated management components or **may** include other
 1046 management requirements not listed to detail management activities. As such the information
 1047 **should** be considered informative.

1048 **Audit**

1049 The audit requirements contain auditable events for the authors to select, if requirements from
 1050 the class FAU, are included in the ST, PP, PP-Module, or security functional package. These
 1051 requirements include security relevant events in terms of the various levels of detail supported
 1052 by the components of the Security audit data generation (FAU_GEN) family.

1053 EXAMPLE 1

1054 an audit note might include actions that are in terms of:

- 1055 — Minimal - successful use of the security mechanism;
- 1056 — Basic - any use of the security mechanism as well as relevant information regarding the security attributes
 1057 involved;
- 1058 — Detailed - any configuration changes made to the mechanism, including the actual configuration values
 1059 before and after the change.

1060 It **can** be observed that the categorization of auditable events is hierarchical.

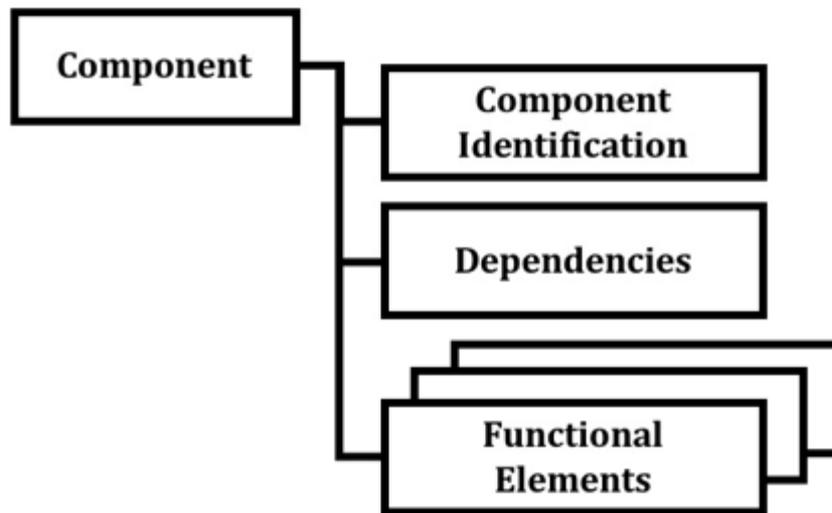
1061 EXAMPLE 2

1062 For example, when Basic Audit Generation is desired, all auditable events identified as being both Minimal and Basic
 1063 are included in the PP, PP-Module, functional package or ST through the use of the appropriate assignment operation,
 1064 except when the higher-level event simply provides more detail than the lower level event. When Detailed Audit
 1065 Generation is desired, all identified auditable events (Minimal, Basic and Detailed) are included in the PP, PP-Module,
 1066 functional package or ST.

1067 In the class FAU the rules governing the audit are explained in more detail.

1068 **6.1.4 Component structure**

1069 Figure 5 illustrates the functional component structure.



1070

1071

Figure 5 — Functional component structure

1072 **Component identification**

1073 The component identification subclause provides descriptive information necessary to identify,
 1074 categorize, register, and cross-reference a component. The following is provided as part of
 1075 every functional component:

1076 *A unique name.* The name reflects the purpose of the component.

1077 *A unique short name.* A unique short form of the functional component name. This short name
 1078 serves as the principal reference name for the categorization, registration, and cross-
 1079 referencing of the component. This short name reflects the class and family to which the
 1080 component belongs and the component number within the family.

1081 *A hierarchical-to list.* A list of other components that this component is hierarchical to and for
 1082 which this component **can** be used to satisfy dependencies to the listed components.

1083 **Functional elements**

1084 A set of elements is provided for each component. Each element is individually defined and is
 1085 self-contained.

1086 When building packages, PPs and/or STs, it is not permitted to select only one or more
 1087 elements from a component. The complete set of elements of a component must be selected for
 1088 inclusion in a PP, PP-Module, security functional package or an ST.

1089 A unique short form of the functional element name is provided.

1090 EXAMPLE

1091 The component name FDP_IFF.4.2 reads as follows:

- 1092 — F - functional requirement,
- 1093 — DP - class "User data protection",
- 1094 — _IFF - family "Information flow control functions",
- 1095 — .4 - 4th component named "Partial elimination of illicit information flows",
- 1096 — .2 - 2nd element of the component.

1097 **Dependencies**

1098 Dependencies among functional components arise when a component is not self-sufficient and
 1099 relies upon the functionality of, or interaction with, another component for its own proper
 1100 functioning.

1101 Each functional component provides a complete list of dependencies to other functional and
 1102 assurance components. Some components **may** list “No dependencies”. The components
 1103 depended upon **may** in turn have dependencies on other components. The list provided in the
 1104 components will be the direct dependencies. That is only references to the other functional
 1105 components that are required for this component to perform its job properly. The indirect
 1106 dependencies, that is the dependencies that result from the depended upon components **can** be
 1107 found in Annex A of this document. It is noted that in some cases the dependency is optional in
 1108 that a number of functional components are provided, where each one of them would be
 1109 sufficient to satisfy the dependency.

1110 EXAMPLE

1111 FDP_UIT.1 Data exchange integrity

1112 The dependency list identifies the minimum functional or assurance components needed to
 1113 satisfy the security requirements associated with an identified component. Components that
 1114 are hierarchical to the identified component **may** also be used to satisfy the dependency.

1115 The dependencies indicated in this document are normative and they **shall** be satisfied within a
 1116 package, PP or ST. In situations where the indicated dependencies are not applicable, the author
 1117 **shall** satisfy the dependency by providing a rationale why it is not applicable and **may** leave the
 1118 depended upon component from the package, PP or ST.

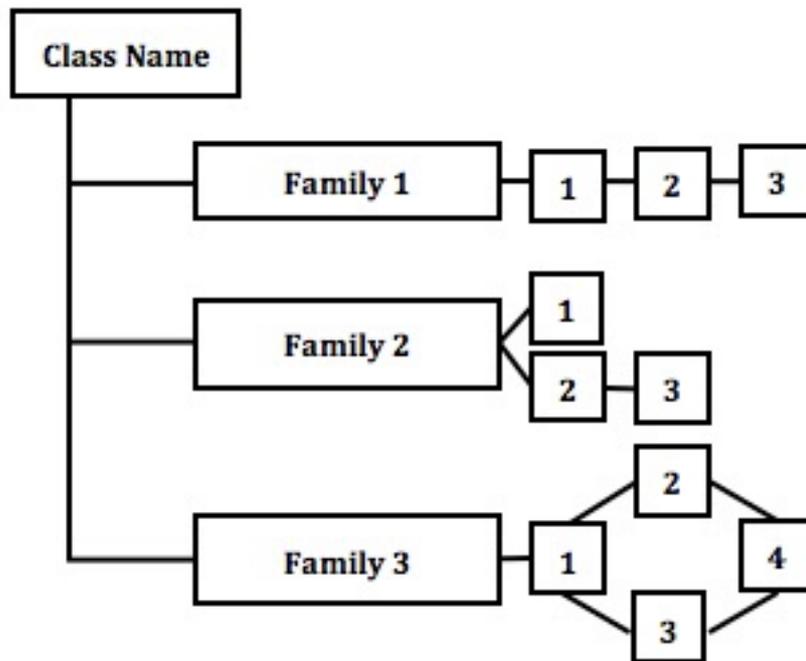
1119 **6.2 Component catalogue**1120 **6.2.1 General**

1121 The grouping of the components in this document does not reflect any formal taxonomy.

1122 This document contains classes of families and components, which are rough groupings on the
 1123 basis of related function or purpose, presented in alphabetic order. At the start of each class is
 1124 an informative diagram that indicates the taxonomy of each class, indicating the families in each
 1125 class and the components in each family. Figure 6 is a useful indicator of the hierarchical
 1126 relationship that **may** exist between components.

1127 In the description of the functional components, a subclause identifies the dependencies
 1128 between the component and any other components.

1129 In each class, a figure describing the family hierarchy similar to Figure 6 is provided. In Figure 6
 1130 the first family, Family 1, contains three hierarchical components, where component 2 and
 1131 component 3 **can** both be used to satisfy dependencies on component 1. Component 3 is
 1132 hierarchical to component 2 and **can** also be used to satisfy dependencies on component 2.



1133

1134

Figure 6 — Sample class decomposition diagram

1135 In Family 2 there are three components not all of which are hierarchical. Components 1 and 2
 1136 are hierarchical to no other components. Component 3 is hierarchical to component 2 and **can**
 1137 be used to satisfy dependencies on component 2, but not to satisfy dependencies on component
 1138 1.

1139 In Family 3, components 2, 3, and 4 are hierarchical to component 1. Components 2 and 3 are
 1140 both hierarchical to component 1, but non-comparable. Component 4 is hierarchical to both
 1141 component 2 and component 3.

1142 These diagrams are meant to complement the text of the families and make identification of the
 1143 relationships easier. They do not replace the “Hierarchical to:” note in each component that is
 1144 the mandatory claim of hierarchy for each component.

1145 6.2.2 Component changes highlighting

1146 The relationship between components within a family is highlighted using a **bolding**
 1147 convention. This bolding convention calls for the bolding of all new requirements. For
 1148 hierarchical components, requirements are bolded when they are enhanced or modified beyond
 1149 the requirements of the previous component. In addition, any new or enhanced permitted
 1150 operations beyond the previous component are also highlighted using **bold** type.

1151

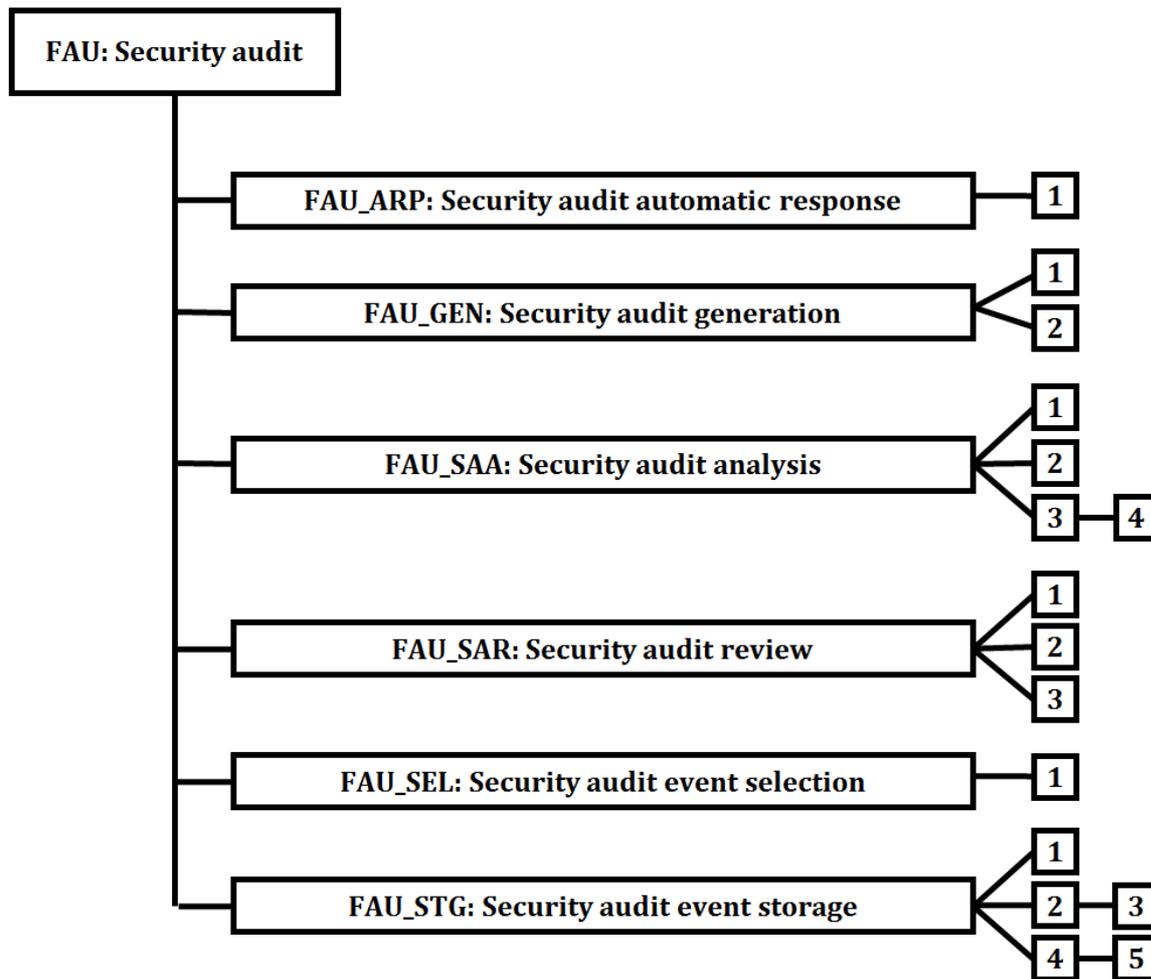
1152 **7 Class FAU: Security audit**

1153 **7.1 Class description**

1154 Security auditing involves recognizing, recording, storing, and analyzing information related to
 1155 security relevant activities (i.e. activities controlled by the TSF). The resulting audit records **can**
 1156 be examined to determine which security relevant activities took place and whom (which user)
 1157 is responsible for them.

1158 Figure 7 shows the decomposition of this class, it’s families and components. Elements are not
 1159 shown in the figure.

1160 Annex C provides explanatory information for this class and **should** be consulted when using
 1161 the components identified in this class.



1162
 1163 **Figure 7 — FAU: Security audit class decomposition**

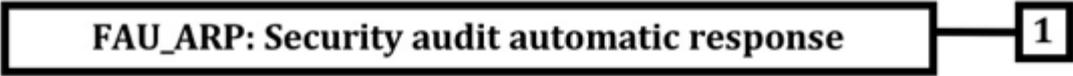
1164 **7.2 Security audit automatic response (FAU_ARP)**

1165 **7.2.1 Family behaviour**

1166 This family defines the response to be taken in case of detected events indicative of a potential
 1167 security violation.

1168 **7.2.2 Components leveling and description**

1169 Figure 8 shows the component leveling for this family.



1

1170

1171

Figure 8 — FAU_ARP: Component leveling

1172 At FAU_ARP.1 Security alarms, the TSF **shall** take actions in case a potential security violation is
1173 detected.

7.2.3 Management of FAU_ARP.1

1175 The following actions **could** be considered for the management functions in FMT:

1176 a) the management (addition, removal, or modification) of actions.

7.2.4 Audit of FAU_ARP.1

1178 The following actions should be auditable if FAU_GEN Security audit data generation is included
1179 in the PP, PP-Module, functional package or ST:

1180 a) Minimal: Actions taken due to potential security violations.

7.2.5 FAU_ARP.1 Security alarms**Component relationships**

1183 Hierarchical to: No other components.

1184 Dependencies: FAU_SAA.1 Potential violation analysis

FAU_ARP.1.1

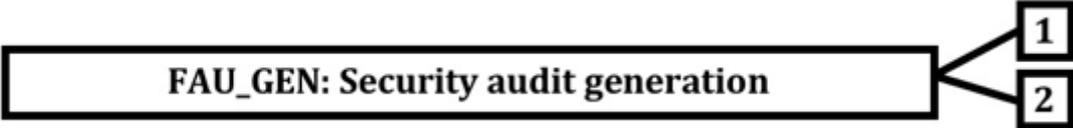
1186 **The TSF shall take [assignment: list of actions] upon detection of a potential security**
1187 **violation.**

7.3 Security audit data generation (FAU_GEN)**7.3.1 Family behaviour**

1190 This family defines requirements for recording the occurrence of security relevant events that
1191 take place under TSF control. This family identifies the level of auditing, enumerates the types
1192 of events that **shall** be auditable by the TSF, and identifies the minimum set of audit-related
1193 information that **should** be provided within various audit record types.

7.3.2 Components leveling and description

1195 Figure 9 shows the component leveling for this family.



1

2

1196

1197

Figure 9 — FAU_GEN: Component leveling

1198 FAU_GEN.1 Audit data generation defines the level of auditable events and specifies the list of
1199 data that **shall** be recorded in each record.

1200 At FAU_GEN.2 User identity association, the TSF **shall** associate auditable events to individual
1201 user identities.

1202 **7.3.3 Management of FAU_GEN.1, FAU_GEN.2**

1203 The following actions **could** be considered for the management functions in FMT:

1204 a) There are no management activities foreseen.

1205 **7.3.4 Audit of FAU_GEN.1, FAU_GEN.2**

1206 The following actions should be auditable if FAU_GEN Security audit data generation is included
1207 in the PP, PP-Module, functional package or ST:

1208 a) There are no auditable events foreseen.

1209 **7.3.5 FAU_GEN.1 Audit data generation**

1210 **Component relationships**

1211 Hierarchical to: No other components.

1212 Dependencies: FPT_STM.1 Reliable time stamps

1213 **FAU_GEN.1.1**

1214 The TSF **shall** be able to generate audit data of the following auditable events:

1215 a) **Start-up and shutdown of the audit functions;**

1216 b) **All auditable events for the [selection, choose one of: *minimum, basic,***
1217 ***detailed, not specified*] level of audit; and**

1218 c) **[assignment: other specifically defined auditable events].**

1219 **FAU_GEN.1.2**

1220 The TSF **shall** record within the audit data at least the following information:

1221 a) **Date and time of the auditable event, type of event, subject identity (if**
1222 **applicable), and the outcome (success or failure) of the event; and**

1223 b) **For each auditable event type, based on the auditable event definitions of the**
1224 **functional components included in the PP, PP-Module, functional package or**
1225 **ST, [assignment: *other audit relevant information*].**

1226 **7.3.6 FAU_GEN.2 User identity association**

1227 **Component relationships**

1228 Hierarchical to: No other components.

1229 Dependencies: FAU_GEN.1 Audit data generation

1230 FIA_UID.1 Timing of identification

1231 **FAU_GEN.2.1**

1232 **For audit events resulting from actions of identified users, the TSF **shall** be able to**
1233 **associate each auditable event with the identity of the user that caused the event.**

1234 **7.4 Security audit analysis (FAU_SAA)**

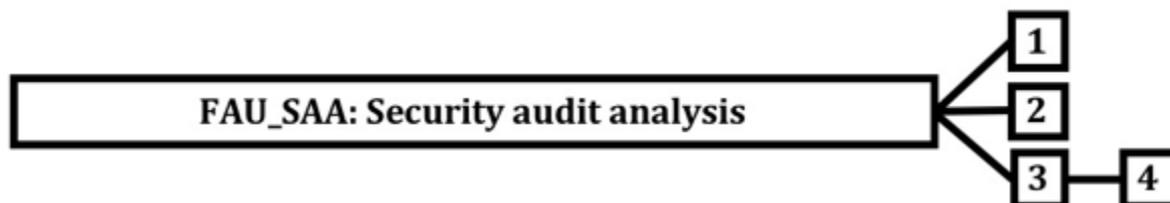
1235 **7.4.1 Family behaviour**

1236 This family defines requirements for automated means that analyze system activity and audit
1237 data looking for possible or real security violations. This analysis **may** work in support of
1238 intrusion detection, or automatic response to a potential security violation.

1239 The actions to be taken based on the detection **can** be specified using the Security audit
1240 automatic response (FAU_ARP) family as desired.

1241 7.4.2 Components leveling and description

1242 Figure 10 shows the component leveling for this family.



1243

1244 **Figure 10 — FAU_SAA: Component leveling**

1245 In FAU_SAA.1 Potential violation analysis, basic threshold detection on the basis of a fixed rule
1246 set is required.

1247 In FAU_SAA.2 Profile based anomaly detection, the TSF maintains individual profiles of system
1248 usage, where a profile represents the historical patterns of usage performed by members of the
1249 profile target group. A profile target group refers to a group of one or more individuals who
1250 interact with the TSF. Each member of a profile target group is assigned an individual suspicion
1251 rating that represents how well that member's current activity corresponds to the established
1252 patterns of usage represented in the profile. This analysis **can** be performed at runtime or
1253 during a post-collection batch-mode analysis.

1254 In FAU_SAA.3 Simple attack heuristics, the TSF **shall** be able to detect the occurrence of
1255 signature events that represent a significant threat to enforcement of the SFRs. This search for
1256 signature events **may** occur in real-time or during a post-collection batch-mode analysis.

1257 In FAU_SAA.4 Complex attack heuristics, the TSF **shall** be able to represent and detect multi-
1258 step intrusion scenarios. The TSF is able to compare system events (possibly performed by
1259 multiple individuals) against event sequences known to represent entire intrusion scenarios.
1260 The TSF **shall** be able to indicate when a signature event or event sequence is found that
1261 indicates a potential violation of the enforcement of the SFRs.

1262 7.4.3 Management of FAU_SAA.1

1263 The following actions **could** be considered for the management functions in FMT:

- 1264 a) Maintenance of the rules by (adding, modifying, deletion) of rules from the set of
1265 rules.

1266 7.4.4 Management of FAU_SAA.2

1267 The following actions **could** be considered for the management functions in FMT:

- 1268 a) Maintenance (deletion, modification, addition) of the group of users in the profile
1269 target group.

1270 7.4.5 Management of FAU_SAA.3

1271 The following actions **could** be considered for the management functions in FMT:

- 1272 a) Maintenance (deletion, modification, addition) of the subset of system events.

1273 7.4.6 Management of FAU_SAA.4

1274 The following actions **could** be considered for the management functions in FMT:

- 1275 a) Maintenance (deletion, modification, addition) of the subset of system events;

1276 b) Maintenance (deletion, modification, addition) of the set of sequences of system
1277 events.

1278 **7.4.7 Audit of FAU_SAA.1, FAU_SAA.2, FAU_SAA.3, FAU_SAA.4**

1279 The following actions should be auditable if FAU_GEN Security audit data generation is included
1280 in the PP, PP-Module, functional package or ST:

1281 a) Minimal: Enabling and disabling of any of the analysis mechanisms;

1282 b) Minimal: Automated responses performed by the tool.

1283 **7.4.8 FAU_SAA.1 Potential violation analysis**

1284 **Component relationships**

1285 Hierarchical to: No other components.

1286 Dependencies: FAU_GEN.1 Audit data generation

1287 **FAU_SAA.1.1**

1288 **The TSF shall be able to apply a set of rules in monitoring the audited events and based**
1289 **upon these rules indicate a potential violation of the enforcement of the SFRs.**

1290 **FAU_SAA.1.2**

1291 **The TSF shall enforce the following rules for monitoring audited events:**

1292 a) **Accumulation or combination of [assignment: *subset of defined auditable***
1293 ***events*] known to indicate a potential security violation;**

1294 b) **[assignment: *any other rules*].**

1295 **7.4.9 FAU_SAA.2 Profile based anomaly detection**

1296 **Component relationships**

1297 Hierarchical to: No other components.

1298 Dependencies: FIA_UID.1 Timing of identification

1299 **FAU_SAA.2.1**

1300 **The TSF shall be able to maintain profiles of system usage, where an individual profile**
1301 **represents the historical patterns of usage performed by the member(s) of [assignment:**
1302 ***the profile target group*].**

1303 **FAU_SAA.2.2**

1304 **The TSF shall be able to maintain a suspicion rating associated with each user whose**
1305 **activity is recorded in a profile, where the suspicion rating represents the degree to**
1306 **which the user's current activity is found inconsistent with the established patterns of**
1307 **usage represented in the profile.**

1308 **FAU_SAA.2.3**

1309 **The TSF shall be able to indicate a possible violation of the enforcement of the SFRs when**
1310 **a user's suspicion rating exceeds the following threshold conditions [assignment:**
1311 ***conditions under which anomalous activity is reported by the TSF*].**

1312 **7.4.10 FAU_SAA.3 Simple attack heuristics**1313 **Component relationships**

1314 Hierarchical to: No other components.

1315 Dependencies: No dependencies.

1316 **FAU_SAA.3.1**

1317 The TSF **shall** be able to maintain an internal representation of the following signature
 1318 events [assignment: *a subset of system events*] that **may** indicate a violation of the
 1319 enforcement of the SFRs.

1320 **FAU_SAA.3.2**

1321 The TSF **shall** be able to compare the signature events against the record of system
 1322 activity discernible from an examination of [assignment: *the information to be used to*
 1323 *determine system activity*].

1324 **FAU_SAA.3.3**

1325 The TSF **shall** be able to indicate a potential violation of the enforcement of the SFRs
 1326 when a system event is found to match a signature event that indicates a potential
 1327 violation of the enforcement of the SFRs.

1328 **7.4.11 FAU_SAA.4 Complex attack heuristics**1329 **Component relationships**

1330 Hierarchical to: FAU_SAA.3 Simple attack heuristics

1331 Dependencies: No dependencies.

1332 **FAU_SAA.4.1**

1333 The TSF **shall** be able to maintain an internal representation of the following **event sequences**
 1334 **of known intrusion scenarios** [assignment: *list of sequences of system events whose*
 1335 *occurrence are representative of known penetration scenarios*] and the following signature
 1336 events [assignment: *a subset of system events*] that **may** indicate a **potential** violation of the
 1337 enforcement of the SFRs.

1338 **FAU_SAA.4.2**

1339 The TSF **shall** be able to compare the signature events **and event sequences** against the record
 1340 of system activity discernible from an examination of [assignment: *the information to be used to*
 1341 *determine system activity*].

1342 **FAU_SAA.4.3**

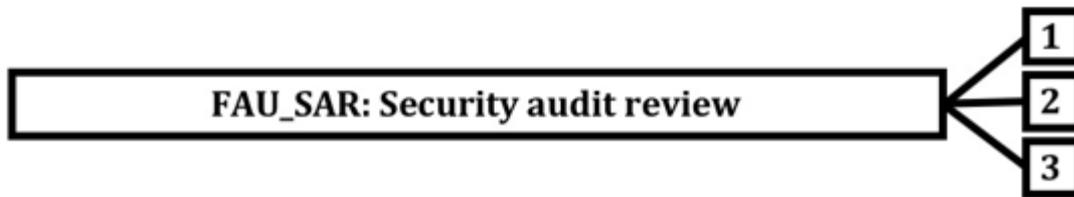
1343 The TSF **shall** be able to indicate a potential violation of the enforcement of the SFRs when
 1344 system **activity** is found to match a signature event **or event sequence** that indicates a
 1345 potential violation of the enforcement of the SFRs.

1346 **7.5 Security audit review (FAU_SAR)**1347 **7.5.1 Family behaviour**

1348 This family defines the requirements for tools that are made available to authorized users to
 1349 assist in the review of audit data.

1350 **7.5.2 Components leveling and description**

1351 Figure 11 shows the component leveling for this family.



1352

1353 **Figure 11 — FAU_SAR: Component leveling**

1354 FAU_SAR.1 Audit review, provides the capability to read information from the audit data.

1355 FAU_SAR.2 Restricted audit review, requires that there are no other users except those that
1356 have been identified in FAU_SAR.1 Audit review that **can** read the information.

1357 FAU_SAR.3 Selectable audit review, requires audit review tools to select the audit data to be
1358 reviewed based on criteria.

1359 **7.5.3 Management of FAU_SAR.1**

1360 The following actions **could** be considered for the management functions in FMT:

- 1361 a) Maintenance (deletion, modification, addition) of the group of users with read
1362 access right to the audit records.

1363 **7.5.4 Management of FAU_SAR.2, FAU_SAR.3**

1364 The following actions **could** be considered for the management functions in FMT:

- 1365 a) There are no management activities foreseen.

1366 **7.5.5 Audit of FAU_SAR.1**

1367 The following actions should be auditable if FAU_GEN Security audit data generation is included
1368 in the PP, PP-Module, functional package or ST:

- 1369 a) Basic: Reading of information from the audit records.

1370 **7.5.6 Audit of FAU_SAR.2**

1371 The following actions should be auditable if FAU_GEN Security audit data generation is included
1372 in the PP, PP-Module, functional package or ST:

- 1373 a) Basic: Unsuccessful attempts to read information from the audit records.

1374 **7.5.7 Audit of FAU_SAR.3**

1375 The following actions should be auditable if FAU_GEN Security audit data generation is included
1376 in the PP, PP-Module, functional package or ST:

- 1377 a) Detailed: the parameters used for the viewing.

1378 **7.5.8 FAU_SAR.1 Audit review**

1379 **Component relationships**

1380 Hierarchical to: No other components.

1381 Dependencies: FAU_GEN.1 Audit data generation

1382 **FAU_SAR.1.1**

1383 The TSF **shall** provide [assignment: *authorized users*] with the capability to read
1384 [assignment: *list of audit information*] from the audit data.

1385 **FAU_SAR.1.2**

1386 The TSF **shall** provide the audit data in a manner suitable for the user to interpret the
1387 information.

1388 **7.5.9 FAU_SAR.2 Restricted audit review**1389 **Component relationships**

1390 Hierarchical to: No other components.

1391 Dependencies: FAU_SAR.1 Audit review

1392 **FAU_SAR.2.1**

1393 The TSF **shall** prohibit all users read access to the audit data, except those users that
1394 have been granted explicit read-access.

1395 **7.5.10 FAU_SAR.3 Selectable audit review**

1396 Hierarchical to: No other components.

1397 Dependencies: FAU_SAR.1 Audit review

1398 **FAU_SAR.3.1**

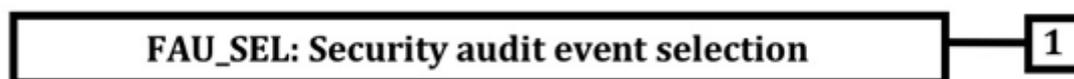
1399 The TSF **shall** provide the ability to apply [assignment: *methods of selection and/or*
1400 *ordering*] of audit data based on [assignment: *criteria with logical relations*].

1401 **7.6 Security audit event selection (FAU_SEL)**1402 **7.6.1 Family behaviour**

1403 This family defines requirements to select the set of events to be audited during TOE operation
1404 from the set of all auditable events.

1405 **7.6.2 Components leveling and description**

1406 Figure 12 shows the component leveling for this family.



1407

1408 **Figure 12 — FAU_SEL: Component leveling**

1409 FAU_SEL.1 Selective audit, requires the ability to select the set of events to be audited from the
1410 set of all auditable events, identified in FAU_GEN.1 Audit data generation, based upon attributes
1411 to be specified by the PP, PP-Module, functional package or ST author.

1412 **7.6.3 Management of FAU_SEL.1**

1413 The following actions **could** be considered for the management functions in FMT:

1414 a) Maintenance of the rights to view/modify the audit data.

1415 **7.6.4 Audit of FAU_SEL.1**

1416 The following actions should be auditable if FAU_GEN Security audit data generation is included
1417 in the PP, PP-Module, functional package or ST:

- 1418 a) Minimal: All modifications to the audit configuration that occur while the audit
1419 collection functions are operating.

1420 **7.6.5 FAU_SEL.1 Selective audit**

1421 **Component relationships**

1422 Hierarchical to: No other components.

1423 Dependencies: FAU_GEN.1 Audit data generation

1424 FMT_MTD.1 Management of TSF data

1425 **FAU_SEL.1.1**

1426 The TSF **shall** be able to select the set of events to be audited from the set of all auditable
1427 events based on the following attributes:

1428 a) [selection: *object identity, user identity, subject identity, host identity, event*
1429 *type*]

1430 b) [assignment: *list of additional attributes that audit selectivity is based upon*]

1431

1432 **7.7 Security audit data storage (FAU_STG)**

1433 **7.7.1 Family behaviour**

1434 This family defines the requirements for the TSF to be able to create and maintain a secure
1435 audit trail. Stored audit data refers to those data stored within an audit trail, and not to any
1436 audit data that has been retrieved (to temporary storage) through selection.

1437 **7.7.2 Components leveling and description**

1438 Figure 13 shows the component leveling for this family.



1439

1440 **Figure 13 — FAU_STG: Component leveling**

1441 FAU_STG.1 Audit data storage location, requires that the storage location(s) for audit data be
1442 specified

1443 FAU_STG.2 Protected audit data storage, requires that protections are placed on the audit data.
1444 It will be protected from unauthorized deletion and/or modification.

1445 FAU_STG.3 Guarantees of audit data availability, specifies the guarantees that the TSF maintains
1446 over the audit data given the occurrence of an undesired condition.

1447 FAU_STG.4 Action in case of possible audit data loss specifies actions to be taken if a threshold
1448 on the stored audit data is exceeded.

1449 FAU_STG.5 Prevention of audit data loss specifies actions to be taken in the case that audit data
1450 storage is full.

1451 **7.7.3 Management of FAU_STG.1**

1452 The following actions **could** be considered for the management functions in FMT:

- 1453 a) Maintenance of remote audit storage locations

1454 **7.7.4 Management of FAU_STG.2**

1455 The following actions **could** be considered for the management functions in FMT:

- 1456 a) There are no management activities foreseen.

1457 **7.7.5 Management of FAU_STG.3**

1458 The following actions **could** be considered for the management functions in FMT:

- 1459 a) Maintenance of the parameters that control the audit data storage capability.

1460 **7.7.6 Management of FAU_STG.4**

1461 The following actions **could** be considered for the management functions in FMT:

- 1462 a) Maintenance (deletion, modification, addition) of actions to be taken in case of
1463 imminent audit data storage failure.

1464 **7.7.7 Management of FAU_STG.5**

1465 The following actions **could** be considered for the management functions in FMT:

- 1466 a) Maintenance (deletion, modification, addition) of actions to be taken in case of
1467 audit data storage failure.

1468 **7.7.8 Audit of FAU_STG.1**

1469 The following actions should be auditable if FAU_GEN Security audit data generation is included
1470 in the PP, PP-Module, functional package or ST:

- 1471 a) Basic: Changes in the location of remote audit data storage.

1472 **7.7.9 Audit of FAU_STG.2, FAU_STG.3**

1473 The following actions should be auditable if FAU_GEN Security audit data generation is included
1474 in the PP, PP-Module, functional package or ST:

- 1475 a) There are no auditable events foreseen.

1476 **7.7.10 Audit of FAU_STG.4**

1477 The following actions should be auditable if FAU_GEN Security audit data generation is included
1478 in the PP, PP-Module, functional package or ST:

- 1479 a) Basic: Actions taken due to exceeding of a threshold.

1480 **7.7.11 Audit of FAU_STG.5**

1481 The following actions should be auditable if FAU_GEN Security audit data generation is included
1482 in the PP, PP-Module, functional package or ST:

- 1483 a) Basic: Actions taken due to the audit data storage failure.

1484 **7.7.12 FAU_STG.1 Audit data storage location**

1485 **Component relationships**

1486	Hierarchical to:	No other components
1487	Dependencies:	FAU_GEN.1 Audit data generation
1488		FTP_ITC.1 Inter-TSF trusted channel

1489 **FAU_STG.1.1**

1490 **The TSF **shall** be able to store generated audit data on the [selection: *TOE itself, transmit*
1491 *the generated audit data to an external IT entity using a trusted channel according to*
1492 *FTP_ITC, [assignment: other storage location(s)].]***

1493 **7.7.13 FAU_STG.2 Protected audit data storage**

1494 **Component relationships**

1495	Hierarchical to:	No other components
1496	Dependencies:	FAU_GEN.1 Audit data generation

1497 **FAU_STG.2.1**

1498 **The TSF **shall** protect the stored audit data in the audit trail from unauthorized deletion.**

1499 **FAU_STG.2.2**

1500 **The TSF shall be able to [selection, choose one of: *prevent, detect*] unauthorized**
 1501 **modifications to the stored audit data in the audit trail.**

1502 **7.7.14 FAU_STG.3 Guarantees of audit data availability**1503 **Component relationships**

1504 Hierarchical to: FAU_STG.2 Protected audit data storage

1505 Dependencies: FAU_GEN.1 Audit data generation

1506 FAU_STG.3.1

1507 The TSF shall protect the stored audit data in the audit trail from unauthorized deletion.

1508 FAU_STG.3.2

1509 The TSF shall be able to [selection, choose one of: *prevent, detect*] unauthorized modifications to
 1510 the stored audit data in the audit trail.

1511 **FAU_STG.3.3**

1512 **The TSF shall ensure that [assignment: *metric for saving audit data*] stored audit data**
 1513 **will be maintained when the following conditions occur: [selection: *audit data storage***
 1514 ***exhaustion, failure, attack*].**

1515 **7.7.15 FAU_STG.4 Action in case of possible audit data loss**1516 **Component relationships**

1517 Hierarchical to: No other components

1518 Dependencies: FAU_STG.2 Protected audit data storage

1519 **FAU_STG.4.1**

1520 The TSF shall [assignment: *actions to be taken in case of possible audit data storage failure*]
 1521 if the audit data storage **exceeds** [assignment: *pre-defined limit*].

1522 **7.7.16 FAU_STG.5 Prevention of audit data loss**1523 **Component relationships**

1524 Hierarchical to: FAU_STG.4 Action in case of possible audit data loss

1525 Dependencies: FAU_STG.2 Protected audit data storage

1526 FAU_GEN.1 Audit data generation

1527 **FAU_STG.5.1**

1528 **The TSF shall [selection: *ignore audited events, "prevent audited events, except those***
 1529 ***taken by the authorized user with special rights", overwrite the oldest stored audit***
 1530 ***records*], [assignment: *other actions to be taken in case of audit storage failure and***
 1531 ***conditions for the actions*] if the audit data storage is full.**

1532

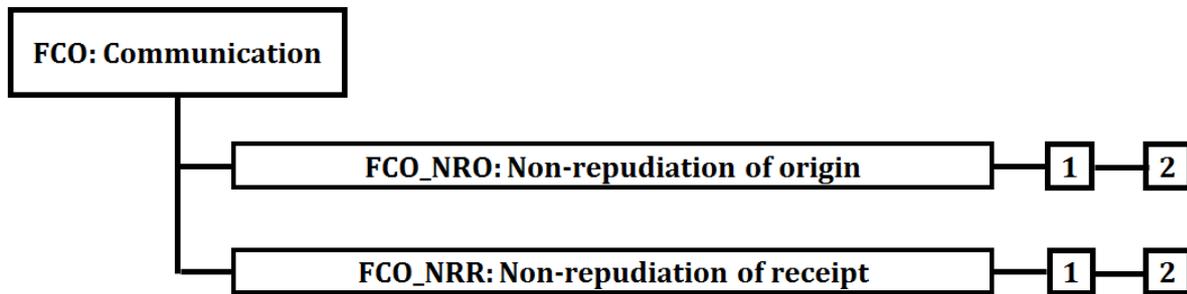
1533 **8 Class FCO: Communication**

1534 **8.1 Class description**

1535 This class provides two families specifically concerned with assuring the identity of a party
 1536 participating in a data exchange. These families are related to assuring the identity of the
 1537 originator of transmitted information (proof of origin) and assuring the identity of the recipient
 1538 of transmitted information (proof of receipt). These families ensure that an originator cannot
 1539 deny having sent the message, nor can the recipient deny having received it. Figure 14 shows
 1540 the decomposition of the class.

1541 Figure 14 shows the decomposition of this class, it's families and components. Elements are not
 1542 shown in the figure.

1543 Annex D provides explanatory information for this class and **should** be consulted when using
 1544 the components identified in this class.



1545
 1546 **Figure 14 — FCO: Communication class decomposition**

1547 **8.2 Non-repudiation of origin (FCO_NRO)**

1548 **8.2.1 Family behaviour**

1549 Non-repudiation of origin ensures that the originator of information cannot successfully deny
 1550 having sent the information. This family requires that the TSF provide a method to ensure that a
 1551 subject that receives information during a data exchange is provided with evidence of the origin
 1552 of the information. This evidence **can** then be verified by either this subject or other subjects.

1553 **8.2.2 Components leveling and description**

1554 Figure 15 shows the component leveling for this family.



1555
 1556 **Figure 15 — FCO_NRO: Component leveling**

1557 FCO_NRO.1 Selective proof of origin, requires the TSF to provide subjects with the capability to
 1558 request evidence of the origin of information.

1559 FCO_NRO.2 Enforced proof of origin, requires that the TSF always generate evidence of origin
 1560 for transmitted information.

1561 **8.2.3 Management of FCO_NRO.1, FCO_NRO.2**

1562 The following actions **could** be considered for the management functions in FMT:

- 1563 a) The management of changes to information types, fields, originator attributes and
- 1564 recipients of evidence.

1565 **8.2.4 Audit of FCO_NRO.1**

1566 The following actions should be auditable if FAU_GEN Security audit data generation is included
1567 in the PP, PP-Module, functional package or ST:

- 1568 a) Minimal: The identity of the user who requested that evidence of origin would be
1569 generated.
- 1570 b) Minimal: The invocation of the non-repudiation service.
- 1571 c) Basic: Identification of the information, the destination, and a copy of the evidence
1572 provided.
- 1573 d) Detailed: The identity of the user who requested a verification of the evidence.

1574 **8.2.5 Audit of FCO_NRO.2**

1575 The following actions should be auditable if FAU_GEN Security audit data generation is included
1576 in the PP, PP-Module, functional package or ST:

- 1577 a) Minimal: The invocation of the non-repudiation service.
- 1578 b) Basic: Identification of the information, the destination, and a copy of the evidence
1579 provided.
- 1580 c) Detailed: The identity of the user who requested a verification of the evidence.

1581 **8.2.6 FCO_NRO.1 Selective proof of origin**1582 **Component relationships**

1583 Hierarchical to: No other components.

1584 Dependencies: FIA_UID.1 Timing of identification

1585 **FCO_NRO.1.1**

1586 **The TSF shall be able to generate evidence of origin for transmitted [assignment: *list of***
1587 ***information types*] at the request of the [selection: *originator, recipient, [assignment: *list****
1588 ***of third parties*]].**

1589 **FCO_NRO.1.2**

1590 **The TSF shall be able to relate the [assignment: *list of attributes*] of the originator of the**
1591 **information, and the [assignment: *list of information fields*] of the information to which**
1592 **the evidence applies.**

1593 **FCO_NRO.1.3**

1594 **The TSF shall provide a capability to verify the evidence of origin of information to**
1595 **[selection: *originator, recipient, [assignment: *list of third parties*]] given [assignment:***
1596 ***limitations on the evidence of origin*].**

1597 **8.2.7 FCO_NRO.2 Enforced proof of origin**1598 **Component relationships**

1599 Hierarchical to: FCO_NRO.1 Selective proof of origin

1600 Dependencies: FIA_UID.1 Timing of identification

1601 **FCO_NRO.2.1**

1602 **The TSF shall enforce the generation of evidence of origin for transmitted [assignment: *list of***
1603 ***information types*] at all times.**

1604 **FCO_NRO.2.2**

1605 The TSF **shall** be able to relate the [assignment: *list of attributes*] of the originator of the
 1606 information, and the [assignment: *list of information fields*] of the information to which the
 1607 evidence applies.

1608 **FCO_NRO.2.3**

1609 The TSF **shall** provide a capability to verify the evidence of origin of information to [selection:
 1610 *originator, recipient, [assignment: list of third parties]*] given [assignment: *limitations on the*
 1611 *evidence of origin*].

1612 **8.3 Non-repudiation of receipt (FCO_NRR)**

1613 **8.3.1 Family behaviour**

1614 Non-repudiation of receipt ensures that the recipient of information cannot successfully deny
 1615 receiving the information. This family requires that the TSF provide a method to ensure that a
 1616 subject that transmits information during a data exchange is provided with evidence of receipt
 1617 of the information. This evidence **can** then be verified by either this subject or other subjects.

1618 **8.3.2 Components leveling and description**

1619 Figure 16 shows the component leveling for this family.



1620

1621 **Figure 16 — FCO_NRR: Component leveling**

1622 FCO_NRR.1 Selective proof of receipt, requires the TSF to provide subjects with a capability to
 1623 request evidence of the receipt of information.

1624 FCO_NRR.2 Enforced proof of receipt, requires that the TSF always generate evidence of receipt
 1625 for received information.

1626 **8.3.3 Management of FCO_NRR.1, FCO_NRR.2**

1627 The following actions **could** be considered for the management functions in FMT:

- 1628 a) The management of changes to information types, fields, originator attributes and
 1629 third-party recipients of evidence.

1630 **8.3.4 Audit of FCO_NRR.1**

1631 The following actions should be auditable if FAU_GEN Security audit data generation is included
 1632 in the PP, PP-Module, functional package or ST:

- 1633 a) Minimal: The identity of the user who requested that evidence of receipt would be
 1634 generated.
- 1635 b) Minimal: The invocation of the non-repudiation service.
- 1636 c) Basic: Identification of the information, the destination, and a copy of the evidence
 1637 provided.
- 1638 d) Detailed: The identity of the user who requested a verification of the evidence.

1639 **8.3.5 Audit of FCO_NRR.2**

1640 The following actions should be auditable if FAU_GEN Security audit data generation is included
 1641 in the PP, PP-Module, functional package or ST:

- 1642 a) Minimal: The invocation of the non-repudiation service.
- 1643 b) Basic: Identification of the information, the destination, and a copy of the evidence
1644 provided.
- 1645 c) Detailed: The identity of the user who requested a verification of the evidence.

1646 8.3.6 FCO_NRR.1 Selective proof of receipt

1647 Component relationships

- 1648 Hierarchical to: No other components.
- 1649 Dependencies: FIA_UID.1 Timing of identification

1650 FCO_NRR.1.1

1651 The TSF **shall** be able to generate evidence of receipt for received [assignment: *list of*
1652 *information types*] at the request of the [selection: *originator, recipient, [assignment: list*
1653 *of third parties*]].

1654 FCO_NRR.1.2

1655 The TSF **shall** be able to relate the [assignment: *list of attributes*] of the recipient of the
1656 information, and the [assignment: *list of information fields*] of the information to which
1657 the evidence applies.

1658 FCO_NRR.1.3

1659 The TSF **shall** provide a capability to verify the evidence of receipt of information to
1660 [selection: *originator, recipient, [assignment: list of third parties]*] given [assignment:
1661 *limitations on the evidence of receipt*].

1662 8.3.7 FCO_NRR.2 Enforced proof of receipt

1663 Component relationships

- 1664 Hierarchical to: FCO_NRR.1 Selective proof of receipt
- 1665 Dependencies: FIA_UID.1 Timing of identification

1666 FCO_NRR.2.1

1667 The TSF **shall enforce the generation of** evidence of receipt for received [assignment: *list of*
1668 *information types*] at **all times**.

1669 FCO_NRR.2.2

1670 The TSF **shall** be able to relate the [assignment: *list of attributes*] of the recipient of the
1671 information, and the [assignment: *list of information fields*] of the information to which the
1672 evidence applies.

1673 FCO_NRR.2.3

1674 The TSF **shall** provide a capability to verify the evidence of receipt of information to [selection:
1675 *originator, recipient, [assignment: list of third parties]*] given [assignment: *limitations on the*
1676 *evidence of receipt*].

1677

1678 **9 Class FCS: Cryptographic support**

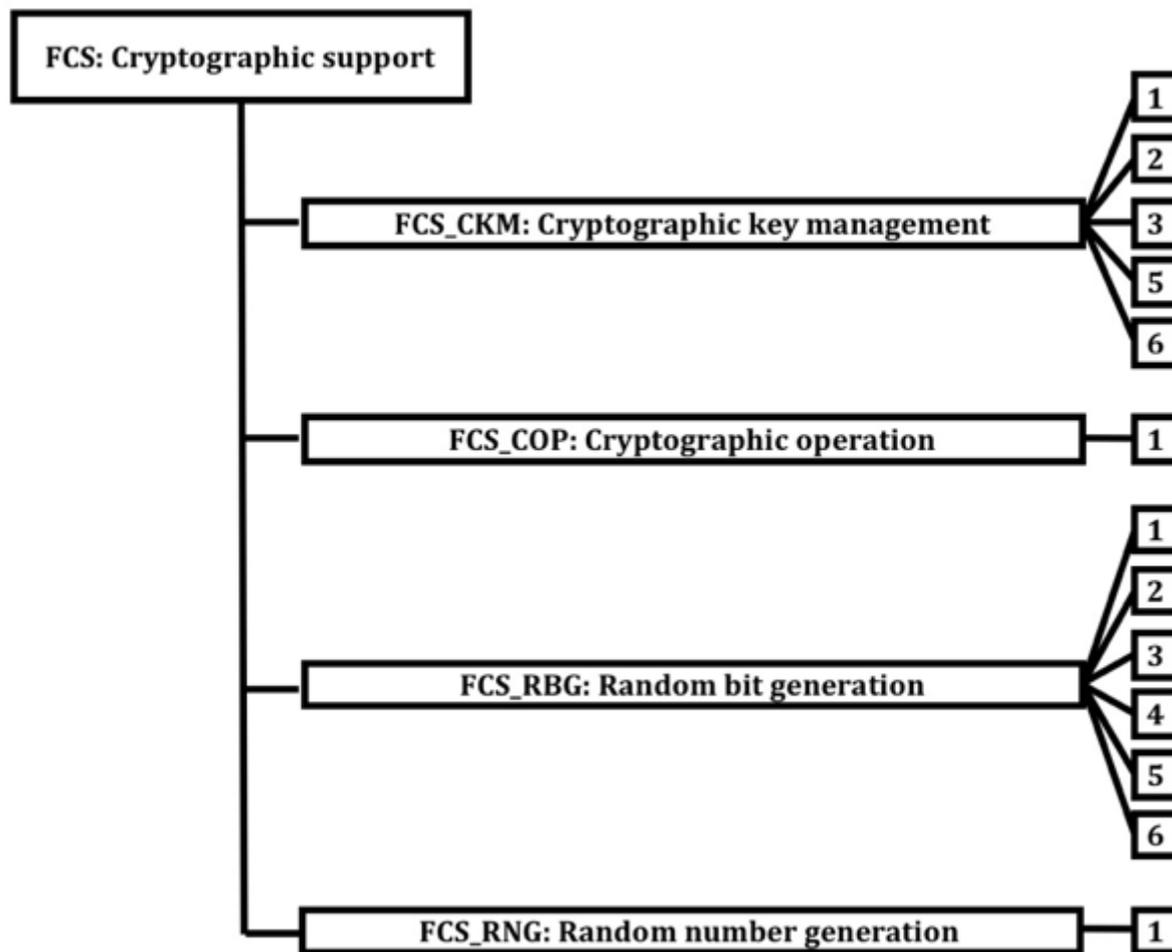
1679 **9.1 Class description**

1680 The TSF **may** employ cryptographic functionality to help satisfy several high-level security
 1681 objectives. These include (but are not limited to): identification and authentication, non-
 1682 repudiation, trusted path, trusted channel, and data separation. This class is used when the TOE
 1683 implements cryptographic functions, the implementation of which **could** be in hardware,
 1684 firmware and/or software.

1685 The FCS: Cryptographic support class is composed of four families.

1686 Figure 17 shows the decomposition of this class, it's families and components. Elements are not
 1687 shown in the figure.

1688 Annex E provides explanatory information for this class and **should** be consulted when using
 1689 the components identified in this class.



1690
 1691 **Figure 17 — FCS: Cryptographic support class decomposition**

1692 **9.2 Cryptographic key management (FCS_CKM)**

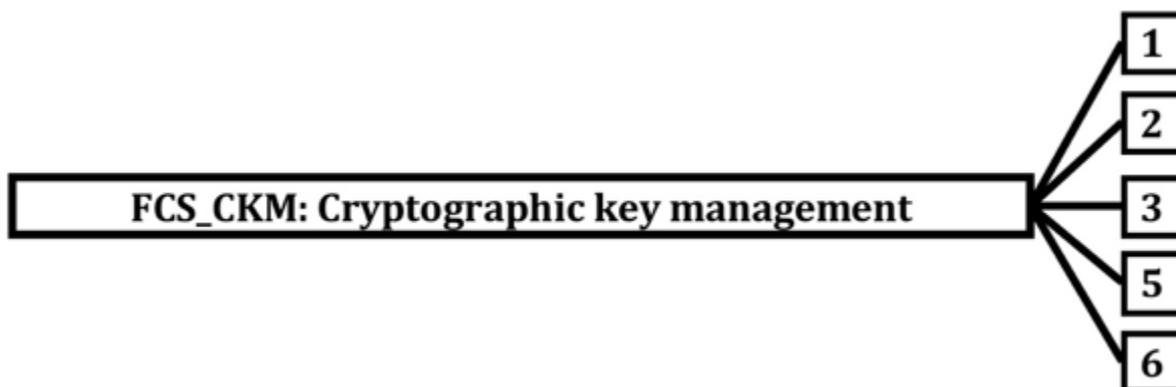
1693 **9.2.1 Family behaviour**

1694 Cryptographic keys must be managed throughout their life cycle. This family is intended to
 1695 support that lifecycle and consequently defines requirements for the following activities:
 1696 cryptographic key generation, cryptographic key derivation, cryptographic key distribution,
 1697 cryptographic key access and timing and event of cryptographic key destruction. This family

1698 **should** be included whenever there are functional requirements for the management of
1699 cryptographic keys.

1700 9.2.2 Components leveling and description

1701 Figure 18 shows the component leveling for this family.



1702

1703 **Figure 18 — FCS_CKM: Component leveling**

1704 FCS_CKM.1 Cryptographic key generation, requires cryptographic keys to be generated in
1705 accordance with a specified algorithm and key sizes which **can** be based on an assigned
1706 standard.

1707 FCS_CKM.2 Cryptographic key distribution, requires cryptographic keys to be distributed in
1708 accordance with a specified distribution method which **can** be based on an assigned standard.

1709 FCS_CKM.3 Cryptographic key access requires access to cryptographic keys to be performed in
1710 accordance with a specified access method which **can** be based on an assigned standard.

1711 FCS_CKM.5 Cryptographic key derivation, requires that the methods, standards, and parameters
1712 for key-derivation are specified.

1713 FCS_CKM.6 Timing and event of cryptographic key destruction, requires cryptographic keys to
1714 be destroyed in accordance with specified destruction methods which **can** be based on an
1715 assigned standard.

1716 NOTE Previous editions of this standard specified FCS_CKM.4 which has been deprecated in this edition of
1717 ISO/IEC 15408-2. In order to preserve consistency when applying different editions of ISO/IEC 15408-2 the
1718 component number has not been re-used.

1719 9.2.3 Management of FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.5, CKM.6

1720 The following actions **could** be considered for the management functions in FMT:

1721 a) There are no management activities foreseen.

1722 9.2.4 Audit of FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.5, CKM.6

1723 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1724 in the PP, PP-Module, functional package or ST:

1725 a) Minimal: Success and failure of the activity.

1726 b) Basic: The object attribute(s), and object value(s) excluding any sensitive
1727 information

1728 9.2.5 FCS_CKM.1 Cryptographic key generation

1729 Component relationships

1730 Hierarchical to: No other components.

1731 Dependencies: [FCS_CKM.2 Cryptographic key distribution, or
 1732 FCS_CKM.5 Cryptographic key derivation, or
 1733 FCS_COP.1 Cryptographic operation]
 1734 FCS_CKM.3 Cryptographic key access
 1735 [FCS_RBG.1 Random bit generation, or
 1736 FCS_RNG.1 Generation of random numbers]
 1737 FCS_CKM.6 Timing and event of cryptographic key
 1738 destruction

1739 **FCS_CKM.1.1**

1740 **The TSF shall generate cryptographic keys in accordance with a specified cryptographic**
 1741 **key generation algorithm [assignment: *cryptographic key generation algorithm*] and**
 1742 **specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the**
 1743 **following: [assignment: *list of standards*].**

1744 **9.2.6 FCS_CKM.2 Cryptographic key distribution**

1745 **Component relationships**

1746 Hierarchical to: No other components.
 1747 Dependencies: [FDP_ITC.1 Import of user data without security
 1748 attributes, or
 1749 FDP_ITC.2 Import of user data with security
 1750 attributes, or
 1751 FCS_CKM.1 Cryptographic key generation or
 1752 FCS_CKM.5 Cryptographic key derivation]
 1753 FCS_CKM.3 Cryptographic key access

1754 **FCS_CKM.2.1**

1755 **The TSF shall distribute cryptographic keys in accordance with a specified cryptographic**
 1756 **key distribution method [assignment: *cryptographic key distribution method*] that meets**
 1757 **the following: [assignment: *list of standards*].**

1758 **9.2.7 FCS_CKM.3 Cryptographic key access**

1759 **Component relationships**

1760 Hierarchical to: No other components.
 1761 Dependencies: [FDP_ITC.1 Import of user data without security
 1762 attributes, or
 1763 FDP_ITC.2 Import of user data with security
 1764 attributes, or
 1765 FCS_CKM.1 Cryptographic key generation or
 1766 FCS_CKM.5 Cryptographic key derivation]

1767 **FCS_CKM.3.1**

1768 **The TSF shall perform [assignment: *type of cryptographic key access*] in accordance with**
 1769 **a specified cryptographic key access method [assignment: *cryptographic key access***
 1770 ***method*] that meets the following: [assignment: *list of standards*].**

1771 **9.2.8 FCS_CKM.4 Cryptographic key destruction**

1772 The component has been deprecated. See FCS_CKM.6 Timing and event of cryptographic key
1773 destruction instead.

1774 **9.2.9 FCS_CKM.5 Cryptographic key derivation**1775 **Component relationships**

1776	Hierarchical to:	No other components.
1777	Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or
1778		FCS_COP.1 Cryptographic operation]
1779		FCS_CKM.6 Timing and event of cryptographic key
1780		destruction

1781 **FCS_CKM.5.1**

1782 **The TSF shall derive cryptographic keys [assignment: *key type*] from [selection: *input***
1783 ***parameters*] in accordance with a specified key derivation algorithm [selection: *key***
1784 ***derivation algorithm*] and specified cryptographic key sizes [selection: *list of key sizes***
1785 **that meet the following: [assignment: *list of standards*].**

1786 NOTE See E.2.6. for information on using this component.

1787 **9.2.10 FCS_CKM.6 Timing and event of cryptographic key destruction**1788 **Component relationships**

1789	Hierarchical to:	No other components
1790	Dependencies:	[FDP_ITC.1 Import of user data without security
1791		attributes, or
1792		FDP_ITC.2 Import of user data with security
1793		attributes, or
1794		FCS_CKM.1 Cryptographic key generation]

1795 **FCS_CKM.6.1**

1796 **The TSF shall destroy [assignment: *list of cryptographic keys (including keying material)*]**
1797 **when [selection: *no longer needed*, [assignment: *other circumstances for key or key***
1798 ***material destruction*]].**

1799 **FCS_CKM.6.2**

1800 **The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1**
1801 **in accordance with a specified cryptographic key destruction method [assignment:**
1802 ***cryptographic key destruction method*] that meets the following: [assignment: *list of***
1803 ***standards*].**

1804 **9.3 Cryptographic operation (FCS_COP)**1805 **9.3.1 Family behaviour**

1806 In order for a cryptographic operation to function correctly, the operation must be performed
1807 in accordance with a specified algorithm and with a cryptographic key of a specified size. This
1808 family **should** be included whenever there are requirements for cryptographic operations to be
1809 performed.

1810 Typical cryptographic operations include data encryption and/or decryption, digital signature
1811 generation and/or verification, cryptographic checksum generation for integrity and/or

1812 verification of checksum, secure hash (message digest), cryptographic key encryption and/or
 1813 decryption, and cryptographic key agreement.

1814 **9.3.2 Components leveling and description**

1815 Figure 19 shows the component leveling for this family.



1816

1817 **Figure 19 — FCS_COP: Component leveling**

1818 FCS_COP.1 Cryptographic operation, requires a cryptographic operation to be performed in
 1819 accordance with a specified algorithm and with a cryptographic key of specified sizes. The
 1820 specified algorithm and cryptographic key sizes **can** be based on an assigned standard.

1821 **9.3.3 Management of FCS_COP.1**

1822 The following actions **could** be considered for the management functions in FCS:

- 1823 a) There are no management activities foreseen.

1824 **9.3.4 Audit of FCS_COP.1**

1825 The following actions should be auditable if FAU_GEN Security audit data generation is included
 1826 in the PP, PP-Module, functional package or ST:

- 1827 a) Minimal: Success and failure, and the type of cryptographic operation.
 1828 b) Basic: Any applicable cryptographic mode(s) of operation, subject attributes and
 1829 object attributes.

1830 **9.3.5 FCS_COP.1 Cryptographic operation**

1831 **Component relationships**

1832	Hierarchical to:	No other components.
1833	Dependencies:	[FDP_ITC.1 Import of user data without security
1834		attributes, or
1835		FDP_ITC.2 Import of user data with security
1836		attributes, or
1837		FCS_CKM.1 Cryptographic key generation, or
1838		FCS_CKM.5 Cryptographic key derivation]
1839		FCS_CKM.3 Cryptographic key access

1840 **FCS_COP.1.1**

1841 **The TSF shall perform [assignment: *list of cryptographic operations*] in accordance with a**
 1842 **specified cryptographic algorithm [assignment: *cryptographic algorithm*] and**
 1843 **cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following:**
 1844 **[assignment: *list of standards*].**

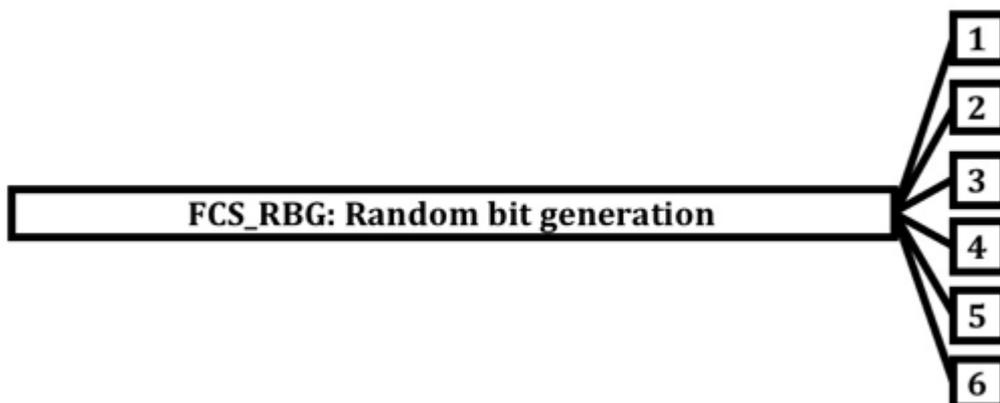
1845 **9.4 Random bit generation (FCS_RBG)**

1846 **9.4.1 Family behaviour**

1847 Components in this family address the requirements for random bit/number generation.

1848 **9.4.2 Components leveling and description**

1849 Figure 20 shows the component leveling for this family.



1850

1851 **Figure 20 — FCS_RBG: Component leveling**1852 FCS_RBG.1 Random bit generation (RBG) requires random bit generation to be performed in
1853 accordance with selected standards.1854 FCS_RBG.2 Random bit generation (external seeding) gives requirements for seeding by an
1855 external (outside the TOE) entropy source.1856 FCS_RBG.3 Random bit generation (internal seeding – single source) gives requirements for
1857 seeding using a TSF entropy source.1858 FCS_RBG.4 Random bit generation (internal seeding – multiple sources) gives requirements for
1859 seeding using multiple TSF entropy sources.1860 FCS_RBG.5 Random bit generation (combining entropy sources) gives requirements for
1861 combining multiple entropy sources (multiple internal sources, internal and external).1862 FCS_RBG.6 Random bit generation service requires random numbers to be supplied over an
1863 external interface as a service to other entities.1864 **9.4.3 Management of FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5,**
1865 **FCS_RBG.6**1866 The following actions **could** be considered for the management functions in FMT:

1867 a) There are no management activities foreseen.

1868 **9.4.4 Audit of FCS_RBG.1, FCS_RBG.2**1869 The following actions should be auditable if FAU_GEN Security audit data generation is included
1870 in the PP, PP-Module, functional package or ST:1871 a) Minimal: failure of the randomization process, failure to initialize or reseed (as
1872 supported by the technology)1873 **9.4.5 Audit of FCS_RBG.3, FCS_RBG.4, FCS_RBG.6, FCS_RBG.6**1874 The following actions should be auditable if FAU_GEN Security audit data generation is included
1875 in the PP, PP-Module, functional package or ST:

1876 a) There are no auditable events foreseen.

1877 **9.4.6 FCS_RBG.1 Random bit generation (RBG)**

1878 **Component relationships**

1879 Hierarchical to: No other components
 1880 Dependencies: [FCS_RBG.2 Random bit generation (external
 1881 seeding), or
 1882 FCS_RBG.3 Random bit generation (internal seeding
 1883 – single source)]
 1884 FPT_FLS.1 Failure with preservation of secure state
 1885 FPT_TST.1 TSF self-testing

1886 **FCS_RBG.1.1**

1887 **The TSF shall perform deterministic random bit generation services using [assignment:**
 1888 **RBG algorithm] in accordance with [assignment: list of standards] after initialization with**
 1889 **a seed.**

1890 **FCS_RBG.1.2**

1891 **The TSF shall initialize and update the RBG state using a noise source under a specified**
 1892 **condition as shown in the RBG State Update Table.**

1893 **RBG State Update Table**

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

1894

1895 **9.4.7 FCS_RBG.2 Random bit generation (external seeding)**

1896 **Component relationships**

1897 Hierarchical to: No other components.
 1898 Dependencies: FCS_RBG.1 Random bit generation (RBG)

1899 **FCS_RBG.2.1**

1900 **The TSF shall be able to accept a minimum input of [assignment: minimum input length**
 1901 **greater than zero] from an external interface for the purpose of seed generation.**

1902 **9.4.8 FCS_RBG.3 Random bit generation (internal seeding – single source)**1903 **Component relationships**

1904 Hierarchical to: No other components

1905 Dependencies: FCS_RBG.1 Random bit generation (RBG)

1906 **FCS_RBG.3.1**

1907 The TSF **shall** be able to seed the RBG using a single [selection: *TSF software-based noise source, TSF hardware-based noise source*] with a minimum of [assignment: *number of bits*] bits of min-entropy.
1908
1909

1910 **9.4.9 FCS_RBG.4 Random bit generation (internal seeding – multiple sources)**1911 **Component relationships**

1912 Hierarchical to: No other components

1913 Dependencies: FCS_RBG.1 Random bit generation (RBG)

1914 FCS_RBG.3 Random bit generation (internal seeding
1915 – single source)1916 **FCS_RBG.4.1**

1917 The TSF **shall** be able to seed the RBG using [selection: *[assignment: number] TSF software-based noise source(s), [assignment: number] TSF hardware-based noise source(s)*].
1918

1919 **9.4.10 FCS_RBG.5 Random bit generation (combining entropy sources)**1920 **Component relationships**

1921 Hierarchical to: No other components.

1922 Dependencies: FCS_RBG.1 Random bit generation (RBG)

1923 [FCS_RBG.2 Random bit generation (external
1924 seeding), or1925 FCS_RBG.3 Random bit generation (internal seeding
1926 – single source)]1927 **FCS_RBG.5.1 Combining entropy sources**

1928 The TSF **shall** [assignment: *combining operation*] [selection: *TSF entropy source(s), TOE external entropy source(s)*] to create the entropy input into the derivation function as
1929 defined in [assignment: *list of standards*], resulting in a minimum of [assignment:
1930 *number of bits*] bits of min-entropy.
1931

1932 **9.4.11 FCS_RBG.6 Random bit generation service**1933 **Component relationships**

1934 Hierarchical to: No other components.

1935 Dependencies: FCS_RBG.1 Random bit generation (RBG)

1936 [FCS_RBG.2 Random bit generation (external
1937 seeding), or1938 FCS_RBG.3 Random bit generation (internal seeding
1939 – single source)]

1940 **FCS_RBG.6.1**

1941 **The TSF shall** provide a [selection: *hardware, software, [assignment: other interface type]*]
 1942 **interface to make the RBG output, as specified in FCS_RBG.1 Random bit generation**
 1943 **(RBG), available as a service to entities outside of the TOE.**

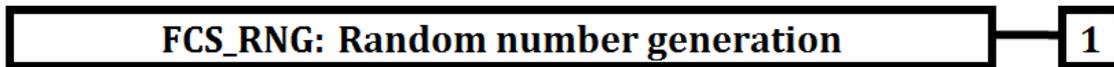
1944 **9.5 Generation of random numbers (FCS_RNG)**

1945 **9.5.1 Family behaviour**

1946 This family defines quality requirements for the generation of random numbers which are
 1947 intended to be use for cryptographic purposes.

1948 **9.5.2 Components leveling and description**

1949 Figure 21 shows the component leveling for this family.



1950

1951 **Figure 21 — FCS_RNG: Component leveling**

1952 FCS_RNG.1 Random number generation requires that random numbers meet a defined quality
 1953 metric.

1954 **9.5.3 Management of FCS_RNG.1**

1955 There are no management activities foreseen.

1956 **9.5.4 Audit of FCS_RNG.1**

1957 There are no actions defined to be auditable.

1958 **9.5.5 FCS_RNG.1 Random number generation**

1959 **Component relationships**

1960 Hierarchical to: No other components.

1961 Dependencies: No dependencies.

1962 **FCS_RNG.1.1**

1963 **The TSF shall** provide a [selection: *physical, non-physical true, deterministic, hybrid*]
 1964 ***physical, hybrid deterministic*] random number generator that implements: [assignment:**
 1965 ***list of security capabilities*].**

1966 **FCS_RNG.1.2**

1967 **The TSF shall** provide [selection: *bits, octets of bits, numbers* [assignment: *format of the*]
 1968 ***numbers*]] that meet [assignment: *a defined quality metric*].**

1969

1970 **10 Class FDP: User data protection**

1971 **10.1 Class description**

1972 This class contains families specifying requirements related to protecting user data. FDP: User
 1973 data protection is split into four groups of families (listed below) that address user data within
 1974 a TOE, during import, export, and storage as well as security attributes directly related to user
 1975 data.

1976 The families in this class are organized into four groups:

1977 a) User data protection security function policies:

1978 — Access control policy (FDP_ACC); and

1979 — Information flow control policy (FDP_IFC).

1980 Components in these families permit the PP, PP-Module, functional package or ST
 1981 author to name the user data protection security function policies and define the
 1982 scope of control of the policy, necessary to address the security objectives. The
 1983 names of these policies are meant to be used throughout the remainder of the
 1984 functional components that have an operation that calls for an assignment or
 1985 selection of an "access control SFP" or an "information flow control SFP". The rules
 1986 that define the functionality of the named access control and information flow
 1987 control SFPs will be defined in the Access control functions (FDP_ACF) and
 1988 Information flow control functions (FDP_IFF) families (respectively).

1989 b) Forms of user data protection:

1990 — Access control functions (FDP_ACF);

1991 — Information flow control functions (FDP_IFF);

1992 — Internal TOE transfer (FDP_ITT);

1993 — Information Retention Control (FDP_IRC)

1994 — Residual information protection (FDP_RIP);

1995 — Rollback (FDP_ROL);

1996 — Stored data confidentiality (FDP_SDC); and

1997 — Stored data integrity (FDP_SDI).

1998 c) Off-line storage, import and export:

1999 — Data authentication (FDP_DAU);

2000 — Export from the TOE (FDP_ETC);

2001 — Import from outside of the TOE (FDP_ITC).

2002 Components in these families address the trustworthy transfer into or out of the
 2003 TOE.

2004 d) Inter-TSF communication:

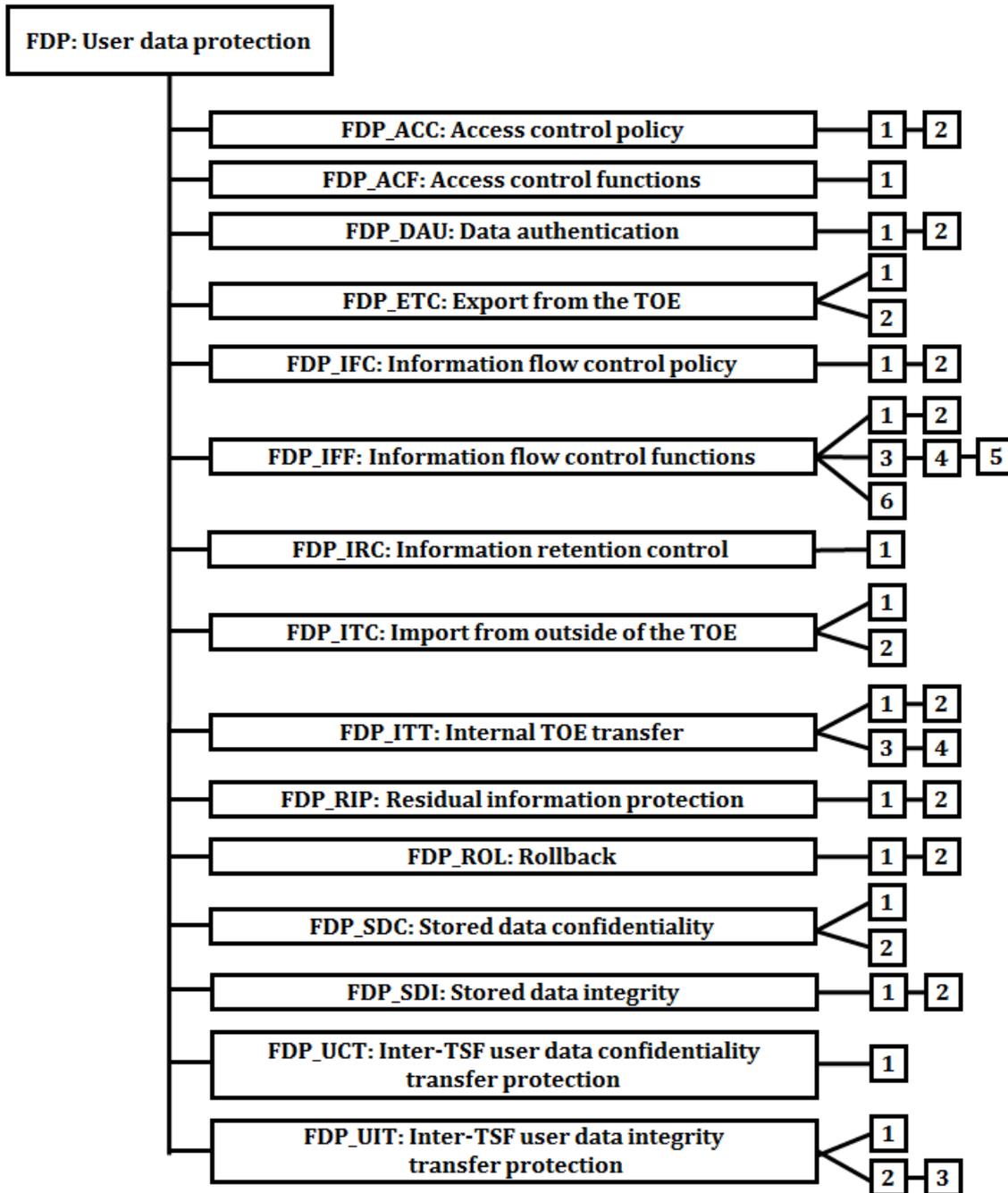
2005 — Inter-TSF user data confidentiality transfer protection (FDP_UCT); and

2006 — Inter-TSF user data integrity transfer protection (FDP_UIT).

2007 — Components in these families address communication between the TSF of the
 2008 TOE and another trusted IT product.

2009 Figure 22 shows the decomposition of this class, it's families and components. Elements are not
 2010 shown in the figure.

2011 Annex F provides explanatory information for this class and **should** be consulted when using
 2012 the components identified in this class.



2013
 2014 **Figure 22 — FDP: User data protection class decomposition**

2015 **10.2 Access control policy (FDP_ACC)**

2016 **10.2.1 Family behaviour**

2017 This family identifies the access control SFPs (by name) and defines the scope of control of the
 2018 policies that form the identified access control portion of the SFRs related to the SFP. This scope
 2019 of control is characterized by three sets: the subjects under control of the policy, the objects
 2020 under control of the policy, and the operations among controlled subjects and controlled
 2021 objects that are covered by the policy. The criteria allow multiple policies to exist, each having a
 2022 unique name. This is accomplished by iterating components from this family once for each

2023 named access control policy. The rules that define the functionality of an access control SFP will
 2024 be defined by other families such as Access control functions (FDP_ACF) and Export from the
 2025 TOE (FDP_ETC). The names of the access control SFPs identified here in Access control policy
 2026 (FDP_ACC) are meant to be used throughout the remainder of the functional components that
 2027 have an operation that calls for an assignment or selection of an “access control SFP.”

2028 10.2.2 Components leveling and description

2029 Figure 23 shows the component leveling for this family.



2030 **Figure 23 — FDP_ACC: Component leveling**

2031 FDP_ACC.1 Subset access control, requires that each identified access control SFP be in place for
 2032 a subset of the possible operations on a subset of the objects in the TOE.

2033 FDP_ACC.2 Complete access control, requires that each identified access control SFP cover all
 2034 operations on subjects and objects covered by that SFP. It further requires that all objects and
 2035 operations protected by the TSF are covered by at least one identified access control SFP.

2036 10.2.3 Management of FDP_ACC.1, FDP_ACC.2

2037 The following actions **could** be considered for the management functions in FMT:

2038 a) There are no management activities foreseen.

2039 10.2.4 Audit of FDP_ACC.1, FDP_ACC.2

2040 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2041 in the PP, PP-Module, functional package or ST:

2042 a) There are no auditable events foreseen.

2043 10.2.5 FDP_ACC.1 Subset access control

2044 Component relationships

2045 Hierarchical to: No other components.

2046 Dependencies: FDP_ACF.1 Security attribute-based access control

2047 FDP_ACC.1.1

2048 The TSF **shall** enforce the [assignment: *access control SFP*] on [assignment: *list of subjects,*
 2049 *objects, and operations among subjects and objects covered by the SFP*].

2050 10.2.6 FDP_ACC.2 Complete access control

2051 Component relationships

2052 Hierarchical to: FDP_ACC.1 Subset access control

2053 Dependencies: FDP_ACF.1 Security attribute-based access control

2054 FDP_ACC.2.1

2055 The TSF **shall** enforce the [assignment: *access control SFP*] on [assignment: *list of subjects and*
 2056 *objects*] and all operations among subjects and objects covered by the **SFP**.

2057 **FDP_ACC.2.2**

2058 **The TSF shall ensure that all operations between any subject controlled by the TSF and**
 2059 **any object controlled by the TSF are covered by an access control SFP.**

2060 **10.3 Access control functions (FDP_ACF)**

2061 **10.3.1 Family behaviour**

2062 This family describes the rules for the specific functions that can implement an access control
 2063 policy named in Access control policy (FDP_ACC). Access control policy (FDP_ACC) specifies the
 2064 scope of control of the policy.

2065 **10.3.2 Components leveling and description**

2066 Figure 24 shows the component leveling for this family.



2067

2068 **Figure 24 — FDP_ACF: Component leveling**

2069 This family addresses security attribute usage and characteristics of policies. The component
 2070 within this family is meant to be used to describe the rules for the function that implements the
 2071 SFP as identified in Access control policy (FDP_ACC). The PP, PP-Module, functional package or
 2072 ST author may also iterate this component to address multiple policies in the TOE.

2073 FDP_ACF.1 Security attribute-based access control Security attribute-based access control
 2074 allows the TSF to enforce access based upon security attributes and named groups of attributes.
 2075 Furthermore, the TSF may have the ability to explicitly authorize or deny access to an object
 2076 based upon security attributes.

2077 **10.3.3 Management of FDP_ACF.1**

2078 The following actions could be considered for the management functions in FMT:

- 2079 a) Managing the attributes used to make explicit access or denial-based decisions.

2080 **10.3.4 Audit of FDP_ACF.1**

2081 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2082 in the PP, PP-Module, functional package or ST:

- 2083 a) Minimal: Successful requests to perform an operation on an object covered by the
 2084 SFP.
- 2085 b) Basic: All requests to perform an operation on an object covered by the SFP.
- 2086 c) Detailed: The specific security attributes used in making an access check.

2087 **10.3.5 FDP_ACF.1 Security attribute-based access control**

2088 **Component relationships**

2089	Hierarchical to:	No other components.
2090	Dependencies:	FDP_ACC.1 Subset access control
2091		FMT_MSA.3 Static attribute

2092 **FDP_ACF.1.1**

2093 The TSF **shall** enforce the [assignment: *access control SFP*] to objects based on the
 2094 following: [assignment: *list of subjects and objects controlled under the indicated SFP, and*
 2095 *for each, the SFP-relevant security attributes, or named groups of SFP-relevant security*
 2096 *attributes*].

2097 **FDP_ACF.1.2**

2098 The TSF **shall** enforce the following rules to determine if an operation among controlled
 2099 subjects and controlled objects is allowed: [assignment: *rules governing access among*
 2100 *controlled subjects and controlled objects using controlled operations on controlled*
 2101 *objects*].

2102 **FDP_ACF.1.3**

2103 The TSF **shall** explicitly authorize access of subjects to objects based on the following
 2104 additional rules: [assignment: *rules, based on security attributes, that explicitly authorize*
 2105 *access of subjects to objects*].

2106 **FDP_ACF.1.4**

2107 The TSF **shall** explicitly deny access of subjects to objects based on the following
 2108 additional rules: [assignment: *rules, based on security attributes, that explicitly deny*
 2109 *access of subjects to objects*].

2110 **10.4 Data authentication (FDP_DAU)**2111 **10.4.1 Family behaviour**

2112 Data authentication permits an entity to accept responsibility for the authenticity of
 2113 information. This family provides a method of providing a guarantee of the validity of a specific
 2114 unit of data that **can** be subsequently used to verify that the information content has not been
 2115 forged or fraudulently modified. In contrast to FAU: Security audit, this family is intended to be
 2116 applied to "static" data rather than data that is being transferred.

2117 **10.4.2 Components leveling and description**

2118 Figure 25 shows the component leveling for this family.



2120 **Figure 25 — FDP_DAU: Component leveling**

2121 FDP_DAU.1 Basic Data Authentication, requires that the TSF is capable of generating a
 2122 guarantee of authenticity of the information content of objects.

2123 FDP_DAU.2 Data Authentication with Identity of Guarantor additionally requires that the TSF is
 2124 capable of establishing the identity of the subject who provided the guarantee of authenticity.

2125 **10.4.3 Management of FDP_DAU.1, FDP_DAU.2**

2126 The following actions **could** be considered for the management functions in FMT:

- 2127 a) The assignment or modification of the objects for which data authentication **may**
 2128 apply **could** be configurable.

2129 **10.4.4 Audit of FDP_DAU.1**

2130 The following actions should be auditable if FAU_GEN Security audit data generation is included
2131 in the PP, PP-Module, functional package or ST:

- 2132 a) Minimal: Successful generation of validity evidence.
- 2133 b) Basic: Unsuccessful generation of validity evidence.
- 2134 c) Detailed: The identity of the subject that requested the evidence.

2135 **10.4.5 Audit of FDP_DAU.2**

2136 The following actions should be auditable if FAU_GEN Security audit data generation is included
2137 in the PP, PP-Module, functional package or ST:

- 2138 a) Minimal: Successful generation of validity evidence.
- 2139 b) Basic: Unsuccessful generation of validity evidence.
- 2140 c) Detailed: The identity of the subject that requested the evidence.
- 2141 d) Detailed: The identity of the subject that generated the evidence.

2142 **10.4.6 FDP_DAU.1 Basic Data Authentication**

2143 **Component relationships**

- 2144 Hierarchical to: No other components.
- 2145 Dependencies: No dependencies.

2146 **FDP_DAU.1.1**

2147 **The TSF shall provide a capability to generate evidence that can be used as a guarantee of**
2148 **the validity of [assignment: *list of objects or information types*].**

2149 **FDP_DAU.1.2**

2150 **The TSF shall provide [assignment: *list of subjects*] with the ability to verify evidence of**
2151 **the validity of the indicated information.**

2152 **10.4.7 FDP_DAU.2 Data Authentication with Identity of Guarantor**

2153 **Component relationships**

- 2154 Hierarchical to: FDP_DAU.1 Basic Data Authentication
- 2155 Dependencies: FIA_UID.1 Timing of identification

2156 **FDP_DAU.2.1**

2157 The TSF shall provide a capability to generate evidence that can be used as a guarantee of the
2158 validity of [assignment: *list of objects or information types*].

2159 **FDP_DAU.2.2**

2160 The TSF shall provide [assignment: *list of subjects*] with the ability to verify evidence of the
2161 validity of the indicated information **and the identity of the user that generated the**
2162 **evidence.**

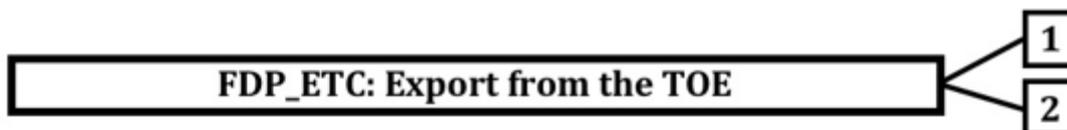
2163 10.5 Export from the TOE (FDP_ETC)

2164 10.5.1 Family behaviour

2165 This family defines functions for TSF-mediated exporting of user data from the TOE such that its
 2166 security attributes and protection either **can** be explicitly preserved or **can** be ignored once it
 2167 has been exported. It is concerned with limitations on export and with the association of
 2168 security attributes with the exported user data.

2169 10.5.2 Components leveling and description

2170 Figure 26 shows the component leveling for this family.



2171

2172 **Figure 26 — FDP_ETC: Component leveling**

2173 FDP_ETC.1 Export of user data without security attributes, requires that the TSF enforces the
 2174 appropriate SFPs when exporting user data outside the TSF. User data that is exported by this
 2175 function is exported without its associated security attributes.

2176 FDP_ETC.2 Export of user data with security attributes, requires that the TSF enforces the
 2177 appropriate SFPs using a function that accurately and unambiguously associates security
 2178 attributes with the user data that is exported.

2179 10.5.3 Management of FDP_ETC.1

2180 The following actions **could** be considered for the management functions in FMT:

2181 a) There are no management activities foreseen.

2182 10.5.4 Management of FDP_ETC.2

2183 The following actions **could** be considered for the management functions in FMT:

2184 a) The additional exportation control rules **could** be configurable by a user in a
 2185 defined role.

2186 10.5.5 Audit of FDP_ETC.1, FDP_ETC.2

2187 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2188 in the PP, PP-Module, functional package or ST:

2189 a) Minimal: Successful export of information.

2190 b) Basic: All attempts to export information.

2191 10.5.6 FDP_ETC.1 Export of user data without security attributes

2192 Component relationships

2193 Hierarchical to: No other components.

2194 Dependencies: [FDP_ACC.1 Subset access control, or
 2195 FDP_IFC.1 Subset information flow control]

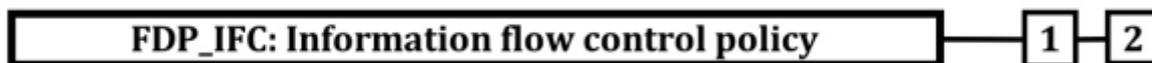
2196 FDP_ETC.1.1

2197 **The TSF shall enforce the [assignment: access control SFP(s) and/or information flow**
 2198 **control SFP(s)] when exporting user data, controlled under the SFP(s), outside of the TOE.**

- 2199 **FDP_ETC.1.2**
- 2200 **The TSF shall export the user data without the user data's associated security attributes.**
- 2201 **10.5.7 FDP_ETC.2 Export of user data with security attributes**
- 2202 **Component relationships**
- | | | |
|------|------------------|--|
| 2203 | Hierarchical to: | No other components. |
| 2204 | Dependencies: | [FDP_ACC.1 Subset access control, or |
| 2205 | | FDP_IFC.1 Subset information flow control] |
- 2206 **FDP_ETC.2.1**
- 2207 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] when exporting user data, controlled under the SFP(s), outside of the TOE.**
- 2208
- 2209 **FDP_ETC.2.2**
- 2210 **The TSF shall export the user data with the user data's associated security attributes.**
- 2211 **FDP_ETC.2.3**
- 2212 **The TSF shall ensure that the security attributes, when exported outside the TOE, are unambiguously associated with the exported user data.**
- 2213
- 2214 **FDP_ETC.2.4**
- 2215 **The TSF shall ensure that interpretation of the security attributes of the exported user data is as intended by the owner of the user data.**
- 2216
- 2217 **FDP_ETC.2.5**
- 2218 **The TSF shall enforce the following rules when user data is exported from the TOE:**
- 2219 **[assignment: *additional exportation control rules*].**
- 2220 **10.6 Information flow control policy (FDP_IFC)**
- 2221 **10.6.1 Family behaviour**
- 2222 This family identifies the information flow control SFPs (by name) and defines the scope of control for each named information flow control SFP. This scope of control is characterized by three sets: the subjects under control of the policy, the information under control of the policy, and operations which cause controlled information to flow to and from controlled subjects covered by the policy. The criteria allow multiple policies to exist, each having a unique name. This is accomplished by iterating components from this family once for each named information flow control policy. The rules that define the functionality of an information flow control SFP will be defined by other families such as Information flow control functions (FDP_IFF) and Export from the TOE (FDP_ETC). The names of the information flow control SFPs identified here in Information flow control policy (FDP_IFC) are meant to be used throughout the remainder of the functional components that have an operation that calls for an assignment or selection of an "information flow control SFP."
- 2223
- 2234 The TSF mechanism controls the flow of information in accordance with the information flow control SFP. Operations that would change the security attributes of information are not generally permitted as this would be in violation of an information flow control SFP. However, such operations may be permitted as exceptions to the information flow control SFP if explicitly specified.
- 2235
- 2236
- 2237
- 2238

2239 **10.6.2 Components leveling and description**

2240 Figure 27 shows the component leveling for this family.

2241 **Figure 27 — FDP_IFC: Component leveling**

2242 FDP_IFC.1 Subset information flow control, requires that each identified information flow
 2243 control SFPs be in place for a subset of the possible operations on a subset of information flows
 2244 in the TOE.

2245 FDP_IFC.2 Complete information flow control, requires that each identified information flow
 2246 control SFP cover all operations on subjects and information covered by that SFP. It further
 2247 requires that all information flows and operations controlled by the TSF are covered by at least
 2248 one identified information flow control SFP.

2249 **10.6.3 Management of FDP_IFC.1, FDP_IFC.2**2250 The following actions **could** be considered for the management functions in FMT:

2251 a) There are no management activities foreseen.

2252 **10.6.4 Audit of FDP_IFC.1, FDP_IFC.2**

2253 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2254 in the PP, PP-Module, functional package or ST:

2255 a) There are no auditable events foreseen.

2256 **10.6.5 FDP_IFC.1 Subset information flow control**2257 **Component relationships**

2258 Hierarchical to: No other components.

2259 Dependencies: FDP_IFF.1 Simple security attributes

2260 **FDP_IFC.1.1**

2261 **The TSF shall enforce the [assignment: information flow control SFP] on [assignment: list**
 2262 **of subjects, information, and operations that cause controlled information to flow to and**
 2263 **from controlled subjects covered by the SFP].**

2264 **10.6.6 FDP_IFC.2 Complete information flow control**2265 **Component relationships**

2266 Hierarchical to: FDP_IFC.1 Subset information flow control

2267 Dependencies: FDP_IFF.1 Simple security attributes

2268 **FDP_IFC.2.1**

2269 The TSF **shall** enforce the [assignment: information flow control SFP] on [assignment: list of
 2270 subjects and **information**] and all operations that cause **that** information to flow to and from
 2271 subjects covered by the **SFP**.

2272 **FDP_IFC.2.2**

2273 **The TSF shall ensure that all operations that cause any information in the TOE to flow to**
 2274 **and from any subject in the TOE are covered by an information flow control SFP.**

2275 10.7 Information flow control functions (FDP_IFF)

2276 10.7.1 Family behaviour

2277 This family describes the rules for the specific functions that **can** implement the information
 2278 flow control SFPs named in Information flow control policy (FDP_IFC), which also specifies the
 2279 scope of control of the policy. It consists of two kinds of requirements: one addressing the
 2280 common information flow function issues, and a second addressing illicit information flows (i.e.
 2281 covert channels). This division arises because the issues concerning illicit information flows are,
 2282 in some sense, orthogonal to the rest of an information flow control SFP. By their nature, they
 2283 circumvent the information flow control SFP resulting in a violation of the policy. As such, they
 2284 require special functions to either limit or prevent their occurrence.

2285 10.7.2 Components leveling and description

2286 Figure 28 shows the component leveling for this family.



2287 **Figure 28 — FDP_IFF: Component leveling**

2288 FDP_IFF.1 Simple security attributes, requires security attributes on information, and on
 2289 subjects that cause that information to flow and on subjects that act as recipients of that
 2290 information. It specifies the rules that must be enforced by the function and describes how
 2291 security attributes are derived by the function.

2292 FDP_IFF.2 Hierarchical security attributes expands on the requirements of FDP_IFF.1 Simple
 2293 security attributes by requiring that all information flow control SFPs in the set of SFRs use
 2294 hierarchical security attributes that form a lattice (as defined in mathematics). FDP_IFF.2.6 is
 2295 derived from the mathematical properties of a lattice. A lattice consists of a set of elements with
 2296 an ordering relationship with the property defined in the first bullet, a least upper bound which
 2297 is the unique element in the set that is greater or equal (in the ordering relationship) than any
 2298 other element of the lattice, and a greatest lower bound, which is the unique element in the set
 2299 that is smaller or equal than any other element of the lattice.

2300 FDP_IFF.3 Limited illicit information flows, requires the SFP to cover illicit information flows,
 2301 but not necessarily eliminate them.

2302 FDP_IFF.4 Partial elimination of illicit information flows, requires the SFP to cover the
 2303 elimination of some (but not necessarily all) illicit information flows.

2304 FDP_IFF.5 No illicit information flows, requires SFP to cover the elimination of all illicit
 2305 information flows.

2306 FDP_IFF.6 Illicit information flow monitoring, requires the SFP to monitor illicit information
 2307 flows for specified and maximum capacities.

2308 10.7.3 Management of FDP_IFF.1, FDP_IFF.2

2309 The following actions **could** be considered for the management functions in FMT:

- 2310 a) Managing the attributes used to make explicit access-based decisions.

2311 10.7.4 Management of FDP_IFF.3, FDP_IFF.4, FDP_IFF.5

2312 The following actions **could** be considered for the management functions in FMT:

- 2313 a) There are no management activities foreseen.

2314 **10.7.5 Management of FDP_IFF.6**2315 The following actions **could** be considered for the management functions in FMT:

- 2316 a) The enabling or disabling of the monitoring function.
- 2317 b) Modification of the maximum capacity at which the monitoring occurs.

2318 **10.7.6 Audit of FDP_IFF.1, FDP_IFF.2, FDP_IFF.5**2319 The following actions should be auditable if FAU_GEN Security audit data generation is included
2320 in the PP, PP-Module, functional package or ST:

- 2321 a) Minimal: Decisions to permit requested information flows.
- 2322 b) Basic: All decisions on requests for information flow.
- 2323 c) Detailed: The specific security attributes used in making an information flow
2324 enforcement decision.
- 2325 d) Detailed: Some specific subsets of the information that has flowed based upon
2326 policy goals.

2327 **10.7.7 Audit of FDP_IFF.3, FDP_IFF.4, FDP_IFF.6**2328 The following actions should be auditable if FAU_GEN Security audit data generation is included
2329 in the PP, PP-Module, functional package or ST:

- 2330 a) Minimal: Decisions to permit requested information flows;
- 2331 b) Basic: All decisions on requests for information flow;
- 2332 c) Basic: The use of identified illicit information flow channels;
- 2333 d) Detailed: The specific security attributes used in making an information flow
2334 enforcement decision;
- 2335 e) Detailed: Some specific subsets of the information that has flowed based upon
2336 policy goals;
- 2337 f) Detailed: The use of identified illicit information flow channels with estimated
2338 maximum capacity exceeding a specified value.

2339 **10.7.8 FDP_IFF.1 Simple security attributes**2340 **Component relationships**

- 2341 Hierarchical to: No other components.
- 2342 Dependencies: FDP_IFC.1 Subset information flow control
- 2343 FMT_MSA.3 Static attribute

2344 **FDP_IFF.1.1**

2345 **The TSF shall enforce the [assignment: *information flow control SFP*] based on the**
2346 **following types of subject and information security attributes: [assignment: *list of***
2347 ***subjects and information controlled under the indicated SFP, and for each, the security***
2348 ***attributes*].**

2349 **FDP_IFF.1.2**

2350 **The TSF shall permit an information flow between a controlled subject and controlled**
2351 **information via a controlled operation if the following rules hold: [assignment: *for each***
2352 ***operation, the security attribute-based relationship that must hold between subject and***
2353 ***information security attributes*].**

2354 **FDP_IFF.1.3**

2355 The TSF **shall** enforce the [assignment: *additional information flow control SFP rules*].

2356 **FDP_IFF.1.4**

2357 The TSF **shall** explicitly authorize an information flow based on the following rules:
 2358 [assignment: *rules, based on security attributes, that explicitly authorize information*
 2359 *flows*].

2360 **FDP_IFF.1.5**

2361 The TSF **shall** explicitly deny an information flow based on the following rules:
 2362 [assignment: *rules, based on security attributes, that explicitly deny information flows*].

2363 **10.7.9 FDP_IFF.2 Hierarchical security attributes**

2364 **Component relationships**

2365	Hierarchical to:	FDP_IFF.1 Simple security attributes
2366	Dependencies:	FDP_IFC.1 Subset information flow control
2367		FMT_MSA.3 Static attribute

2368 **FDP_IFF.2.1**

2369 The TSF **shall** enforce the [assignment: *information flow control SFP*] based on the following
 2370 types of subject and information security attributes: [assignment: *list of subjects and*
 2371 *information controlled under the indicated SFP, and for each, the security attributes*].

2372 **FDP_IFF.2.2**

2373 The TSF **shall** permit an information flow between a controlled subject and controlled
 2374 information via a controlled operation if the following rules, **based on the ordering**
 2375 **relationships between security attributes** hold: [assignment: *for each operation, the security*
 2376 *attribute-based relationship that must hold between subject and information security attributes*].

2377 **FDP_IFF.2.3**

2378 The TSF **shall** enforce the [assignment: *additional information flow control SFP rules*].

2379 **FDP_IFF.2.4**

2380 The TSF **shall** explicitly authorize an information flow based on the following rules:
 2381 [assignment: *rules, based on security attributes, that explicitly authorize information flows*].

2382 **FDP_IFF.2.5**

2383 The TSF **shall** explicitly deny an information flow based on the following rules: [assignment:
 2384 *rules, based on security attributes, that explicitly deny information flows*].

2385 **FDP_IFF.2.6**

2386 The TSF **shall** enforce the following relationships for any two valid information flow
 2387 control security attributes:

- 2388 a) **There exists an ordering function that, given two valid security attributes,**
- 2389 **determines if the security attributes are equal, if one security attribute is**
- 2390 **greater than the other, or if the security attributes are incomparable; and**

2425 **FDP_IFF.6.1**

2426 **The TSF shall enforce the [assignment: *information flow control SFP*] to monitor**
 2427 **[assignment: *types of illicit information flows*] when it exceeds the [assignment: *maximum***
 2428 ***capacity*].**

2429 **10.8 Information Retention Control (FDP_IRC)**

2430 **10.8.1 Family behaviour**

2431 The “Information retention control” family addresses a basic need in secure information
 2432 processing and storage applications for the secure management of data no longer needed by the
 2433 TOE to perform its operations, but that is still stored in the TOE.

2434 The historical view of IT systems as data storage systems suggested that once entered, data
 2435 would seldom be deleted from the system, and if it was deleted, this would mainly be because of
 2436 storage exhaustion problems.

2437 However, in a multilateral or high security environment it is important to minimize the
 2438 replication of data, as well as the time period during which data is stored in the system. It is also
 2439 possible that users could want their IT products to avoid retaining sensitive data that they
 2440 consider to be exploitable by third parties or that could threaten privacy. FDP_IRC may help
 2441 users to gain confidence that the product is secure by deleting every copy of the data when it is
 2442 no longer needed.

2443 The FDP_RIP “Residual information protection” family addresses one side of this problem, but
 2444 an explicit requirement on the management of data that is no longer needed is missing.

2445 Of course, competing requirements may arise, since some data may be needed by the system for
 2446 more operations over a longer time period. Possible solutions to this problem are:

- 2447 — Better protecting the information objects stored in the TOE from access,
- 2448 — Re-requesting the protected information from the user each time it is needed.

2449 Information retention control ensures, that data no longer necessary for the operation of the
 2450 TOE is deleted by the TOE. Components of this family require the PP, PP-Module, functional
 2451 package or ST author to identify the TOE operations, including both simple and complex
 2452 processing and the information objects, that are not to be kept in the TOE, that are the subject of
 2453 those operations.

2454 The TOE is also required to keep track of such stored information objects, and to delete both the
 2455 on-line and the off-line information objects that are no longer required.

2456 This family sets only requirements on information objects requested for specific activities in the
 2457 TOE operation, and not on general data gathering. The policies which control the collection,
 2458 storage, processing, disclosure, and elimination of general user data stored on the TOE must be
 2459 detailed elsewhere, and are domain of the environmental objectives and organizational policies,
 2460 not of the PP, PP-Module, functional package or ST.

2461 When more than one operation requires the presence of a protected object, all operations,
 2462 which refer to the required object shall end before deleting it.

2463 **10.8.2 Components leveling and description**

2464 Figure 29 shows the component leveling for this family.



2465

2466

Figure 29 — FDP_IRC: Component leveling

2467 FDP_IRC.1 Information retention control requires that the TSF ensure that any copy of a defined
 2468 set of objects in the TOE is deleted when no longer strictly necessary for the operation of the
 2469 TOE, and to identify and define the operations for which the object is required.

2470 10.8.3 Management of FDP_IRC.1

2471 The following actions **could** be considered for the management functions in FMT:

2472 a) There are no management actions foreseen.

2473 10.8.4 Audit of FDP_IRC.1

2474 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2475 in the PP, PP-Module, functional package or ST:

2476 a) There are no auditable events foreseen.

2477 10.8.5 FDP_IRC.1 Information retention control

2478 Component relationships

2479 Hierarchical to: No other components.

2480 Dependencies: No dependencies.

2481 FDP_IRC.1.1

2482 The TSF **shall** enforce the [assignment: *information erasure policy*] on a [assignment: *list of*
 2483 *objects*] required for [assignment: *list of operations*] so that the selected objects are deleted
 2484 irreversibly and untraceably from the TOE promptly upon termination of the selected
 2485 operations.

2486 FDP_IRC.1.2

2487 The TSF **shall** ensure that [assignment: *list of objects*] cannot be accessed after their
 2488 release and prior to their irreversible and untraceable deletion.

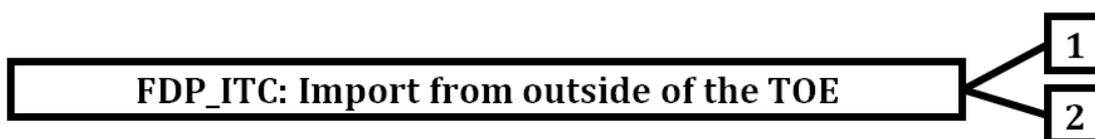
2489 10.9 Import from outside of the TOE (FDP_ITC)

2490 10.9.1 Family behaviour

2491 This family defines the mechanisms for TSF-mediated importing of user data into the TOE such
 2492 that it has appropriate security attributes and is appropriately protected. It is concerned with
 2493 limitations on importation, determination of desired security attributes, and interpretation of
 2494 security attributes associated with the user data.

2495 10.9.2 Components leveling and description

2496 Figure 30 shows the component leveling for this family.



2497

2498 **Figure 30 — FDP_ITC: Component leveling**

2499 FDP_ITC.1 Import of user data without security attributes, requires that the security attributes
 2500 correctly represent the user data and are supplied separately from the object.

2501 FDP_ITC.2 Import of user data with security attributes, requires that security attributes
2502 correctly represent the user data and are accurately and unambiguously associated with the
2503 user data imported from outside the TOE.

2504 **10.9.3 Management of FDP_ITC.1, FDP_ITC.2**

2505 The following actions **could** be considered for the management functions in FMT:

2506 a) The modification of the additional control rules used for import.

2507 **10.9.4 Audit of FDP_ITC.1, FDP_ITC.2**

2508 The following actions should be auditable if FAU_GEN Security audit data generation is included
2509 in the PP, PP-Module, functional package or ST:

2510 a) Minimal: Successful import of user data, including any security attributes.

2511 b) Basic: All attempts to import user data, including any security attributes.

2512 c) Detailed: The specification of security attributes for imported user data supplied by
2513 an authorized user.

2514 **10.9.5 FDP_ITC.1 Import of user data without security attributes**

2515 **Component relationships**

2516 Hierarchical to: No other components.

2517 Dependencies: [FDP_ACC.1 Subset access control, or
2518 FDP_IFC.1 Subset information flow control]

2519 FMT_MSA.3 Static attribute initialization

2520 **FDP_ITC.1.1**

2521 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow***
2522 ***control SFP(s)] when importing user data, controlled under the SFP, from outside of the***
2523 ***TOE.***

2524 **FDP_ITC.1.2**

2525 **The TSF shall ignore any security attributes associated with the user data when imported**
2526 **from outside the TOE.**

2527 **FDP_ITC.1.3**

2528 **The TSF shall enforce the following rules when importing user data controlled under the**
2529 **SFP from outside the TOE: [assignment: *additional importation control rules*].**

2530 **10.9.6 FDP_ITC.2 Import of user data with security attributes**

2531 **Component relationships**

2532 Hierarchical to: No other components.

2533 Dependencies: [FDP_ACC.1 Subset access control, or
2534 FDP_IFC.1 Subset information flow control]

2535 [FTP_ITC.1 Inter-TSF trusted channel, or
2536 FTP_TRP.1 Trusted path]

2537 FPT_TDC.1 Inter-TSF basic TSF data consistency

2538 **FDP_ITC.2.1**

2539 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] when importing user data, controlled under the SFP, from outside of the**
 2540 **TOE.**
 2541

2542 **FDP_ITC.2.2**

2543 **The TSF shall use the security attributes associated with the imported user data.**

2544 **FDP_ITC.2.3**

2545 **The TSF shall ensure that the protocol used provides for the unambiguous association**
 2546 **between the security attributes and the user data received.**

2547 **FDP_ITC.2.4**

2548 **The TSF shall ensure that interpretation of the security attributes of the imported user**
 2549 **data is as intended by the source of the user data.**

2550 **FDP_ITC.2.5**

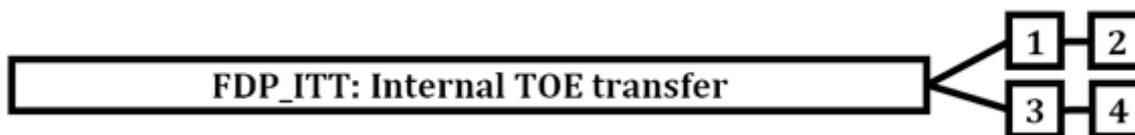
2551 **The TSF shall enforce the following rules when importing user data controlled under the**
 2552 **SFP from outside the TOE: [assignment: *additional importation control rules*].**

2553 **10.10 Internal TOE transfer (FDP_ITT)**2554 **10.10.1 Family behaviour**

2555 This family provides requirements that address protection of user data when it is transferred
 2556 between separated parts of a TOE across an internal channel. This **may** be contrasted with the
 2557 Inter-TSF user data confidentiality transfer protection (FDP_UCT) and Inter-TSF user data
 2558 integrity transfer protection (FDP_UIT) families, which provide protection for user data when it
 2559 is transferred between distinct TSFs across an external channel, and Export from the TOE
 2560 (FDP_ETC) and Import from outside of the TOE (FDP_ITC), which address TSF-mediated
 2561 transfer of data to or from outside the TOE.

2562 **10.10.2 Components leveling and description**

2563 Figure 31 shows the component leveling for this family.



2564

2565 **Figure 31 — FDP_ITT: Component leveling**

2566 FDP_ITT.1 Basic internal transfer protection, requires that user data be protected when
 2567 transmitted between parts of the TOE.

2568 FDP_ITT.2 Transmission separation by attribute, requires separation of data based on the value
 2569 of SFP-relevant attributes in addition to the first component.

2570 FDP_ITT.3 Integrity monitoring, requires that the TSF monitor user data transmitted between
 2571 parts of the TOE for identified integrity errors.

2572 FDP_ITT.4 Attribute-based integrity monitoring expands on the third component by allowing
 2573 the form of integrity monitoring to differ by SFP-relevant attribute.

2574 **10.10.3 Management of FDP_ITT.1, FDP_ITT.2**

2575 The following actions **could** be considered for the management functions in FMT:

- 2576 a) If the TSF provides multiple methods to protect user data during transmission
 2577 between physically separated parts of the TOE, the TSF **could** provide a pre-defined
 2578 role with the ability to select the method that will be used.

2579 **10.10.4 Management of FDP_ITT.3, FDP_ITT.4**

2580 The following actions **could** be considered for the management functions in FMT:

- 2581 a) The specification of the actions to be taken upon detection of an integrity error
 2582 **could** be configurable.

2583 **10.10.5 Audit of FDP_ITT.1, FDP_ITT.2**

2584 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2585 in the PP, PP-Module, functional package or ST:

- 2586 a) Minimal: Successful transfers of user data, including identification of the protection
 2587 method used.
 2588 b) Basic: All attempts to transfer user data, including the protection method used and
 2589 any errors that occurred.

2590 **10.10.6 Audit of FDP_ITT.3, FDP_ITT.4**

2591 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2592 in the PP, PP-Module, functional package or ST:

- 2593 a) Minimal: Successful transfers of user data, including identification of the integrity
 2594 protection method used.
 2595 b) Basic: All attempts to transfer user data, including the integrity protection method
 2596 used and any errors that occurred.
 2597 c) Basic: Unauthorized attempts to change the integrity protection method.
 2598 d) Detailed: The action taken upon detection of an integrity error.

2599 **10.10.7 FDP_ITT.1 Basic internal transfer protection**

2600 **Component relationships**

- 2601 Hierarchical to: No other components.
 2602 Dependencies: [FDP_ACC.1 Subset access control, or
 2603 FDP_IFC.1 Subset information flow control]

2604 **FDP_ITT.1.1**

2605 **The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow*
 2606 *control SFP(s)] to prevent the [selection: *disclosure, modification, loss of use*] of user data
 2607 when it is transmitted between physically-separated parts of the TOE.***

2608 **10.10.8 FDP_ITT.2 Transmission separation by attribute**

2609 **Component relationships**

- 2610 Hierarchical to: FDP_ITT.1 Basic internal transfer protection
 2611 Dependencies: [FDP_ACC.1 Subset access control, or
 2612 FDP_IFC.1 Subset information flow control]

2613 **FDP_ITT.2.1**

2614 The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow control*
 2615 *SFP(s)*] to prevent the [selection: *disclosure, modification, loss of use*] of user data when it is
 2616 transmitted between physically-separated parts of the TOE.

2617 **FDP_ITT.2.2**

2618 **The TSF shall separate data controlled by the SFP(s) when transmitted between**
 2619 **physically-separated parts of the TOE, based on the values of the following: [assignment:**
 2620 ***security attributes that require separation*].**

2621 **10.10.9 FDP_ITT.3 Integrity monitoring**2622 **Component relationships**

2623	Hierarchical to:	No other components.
2624	Dependencies:	[FDP_ACC.1 Subset access control, or
2625		FDP_IFC.1 Subset information flow control]
2626		FDP_ITT.1 Basic internal transfer protection

2627 **FDP_ITT.3.1**

2628 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow***
 2629 ***control SFP(s)*] to monitor user data transmitted between physically-separated parts of**
 2630 **the TOE for the following errors: [assignment: *integrity errors*].**

2631 **FDP_ITT.3.2**

2632 **Upon detection of a data integrity error, the TSF shall [assignment: *specify the action to***
 2633 ***be taken upon integrity error*].**

2634 **10.10.10 FDP_ITT.4 Attribute-based integrity monitoring**2635 **Component relationships**

2636	Hierarchical to:	FDP_ITT.3 Integrity monitoring
2637	Dependencies:	[FDP_ACC.1 Subset access control, or
2638		FDP_IFC.1 Subset information flow control]
2639		FDP_ITT.2 Transmission separation by attribute

2640 **FDP_ITT.4.1**

2641 The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow control*
 2642 *SFP(s)*] to monitor user data transmitted between physically-separated parts of the TOE for the
 2643 following errors: [assignment: *integrity errors*], **based on the following attributes:**
 2644 **[assignment: *security attributes that require separate transmission channels*].**

2645 **FDP_ITT.4.2**

2646 Upon detection of a data integrity error, the TSF **shall** [assignment: *specify the action to be taken*
 2647 *upon integrity error*].

2648 **10.11 Residual information protection (FDP_RIP)**

2649 **10.11.1 Family behaviour**

2650 This family addresses the need to ensure that any data contained in a resource is not available
 2651 when the resource is de-allocated from one object and reallocated to a different object. This
 2652 family requires protection for any data contained in a resource that has been logically deleted
 2653 or released but **may** still be present within the TSF-controlled resource which in turn **may** be re-
 2654 allocated to another object.

2655 **10.11.2 Components leveling and description**

2656 Figure 32 shows the component leveling for this family.



2657 **Figure 32 — FDP_RIP: Component leveling**

2658 FDP_RIP.1 Subset residual information protection, requires that the TSF ensure that any
 2659 residual information content of any resources is unavailable to a defined subset of the objects
 2660 controlled by the TSF upon the resource's allocation or deallocation.

2661 FDP_RIP.2 Full residual information protection, requires that the TSF ensure that any residual
 2662 information content of any resources is unavailable to all objects upon the resource's allocation
 2663 or deallocation.

2664 **10.11.3 Management of FDP_RIP.1, FDP_RIP.2**

2665 The following actions **could** be considered for the management functions in FMT:

- 2666 a) The choice of when to perform residual information protection (i.e. upon allocation
 2667 or deallocation) **could** be made configurable within the TOE.

2668 **10.11.4 Audit of FDP_RIP.1, FDP_RIP.2**

2669 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2670 in the PP, PP-Module, functional package or ST:

- 2671 a) There are no auditable events foreseen.

2672 **10.11.5 FDP_RIP.1 Subset residual information protection**

2673 **Component relationships**

2674 Hierarchical to: No other components.

2675 Dependencies: No dependencies.

2676 **FDP_RIP.1.1**

2677 **The TSF shall ensure that any previous information content of a resource is made**
 2678 **unavailable upon the [selection: *allocation of the resource to, deallocation of the resource***
 2679 **from] the following objects: [assignment: *list of objects*].**

2680 **10.11.6 FDP_RIP.2 Full residual information protection**

2681 **Component relationships**

2682 Hierarchical to: FDP_RIP.1 Subset residual information protection

2683 Dependencies: No dependencies.

2684 **FDP_RIP.2.1**

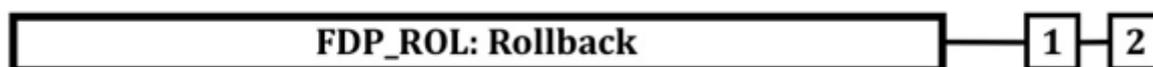
2685 The TSF **shall** ensure that any previous information content of a resource is made unavailable
2686 upon the [selection: *allocation of the resource to, deallocation of the resource from*] **all** objects.

2687 **10.12 Rollback (FDP_ROL)**2688 **10.12.1 Family behaviour**

2689 The rollback operation involves undoing the last operation or a series of operations, bounded
2690 by some limit, such as a period of time, and return to a previous known state. Rollback provides
2691 the ability to undo the effects of an operation or series of operations to preserve the integrity of
2692 the user data.

2693 **10.12.2 Components leveling and description**

2694 Figure 33 shows the component leveling for this family.



2695 **Figure 33 — FDP_ROL: Component leveling**

2696 FDP_ROL.1 Basic rollback addresses a need to roll back or undo a limited number of operations
2697 within the defined bounds.

2698 FDP_ROL.2 Advanced rollback addresses the need to roll back or undo all operations within the
2699 defined bounds.

2700 **10.12.3 Management of FDP_ROL.1, FDP_ROL.2**

2701 The following actions **could** be considered for the management functions in FMT:

- 2702 a) The boundary limit to which rollback **may** be performed **could** be a configurable
2703 item within the TOE.
- 2704 b) Permission to perform a rollback operation **could** be restricted to a well-defined
2705 role.

2706 **10.12.4 Audit of FDP_ROL.1, FDP_ROL.2**

2707 The following actions should be auditable if FAU_GEN Security audit data generation is included
2708 in the PP, PP-Module, functional package or ST:

- 2709 a) Minimal: All successful rollback operations.
- 2710 b) Basic: All attempts to perform rollback operations.
- 2711 c) Detailed: All attempts to perform rollback operations, including identification of the
2712 types of operations rolled back.

2713 **10.12.5 FDP_ROL.1 Basic rollback**2714 **Component relationships**

2715	Hierarchical to:	No other components.
2716	Dependencies:	[FDP_ACC.1 Subset access control, or
2717		FDP_IFC.1 Subset information flow control]

2718 **FDP_ROL.1.1**

2719 The TSF **shall** enforce [assignment: *access control SFP(s) and/or information flow control*
 2720 *SFP(s)*] to permit the rollback of the [assignment: *list of operations*] on the [assignment:
 2721 *information and/or list of objects*].

2722 **FDP_ROL.1.2**

2723 The TSF **shall** permit operations to be rolled back within the [assignment: *boundary limit*
 2724 *to which rollback may be performed*].

2725 **10.12.6 FDP_ROL.2 Advanced rollback**

2726 **Component relationships**

2727	Hierarchical to:	FDP_ROL.1 Basic rollback
2728	Dependencies:	[FDP_ACC.1 Subset access control, or
2729		FDP_IFC.1 Subset information flow control]

2730 **FDP_ROL.2.1**

2731 The TSF **shall** enforce [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
 2732 to permit the rollback of **all** the **operations** on the [assignment: *list of objects*].

2733 **FDP_ROL.2.2**

2734 The TSF **shall** permit operations to be rolled back within the [assignment: *boundary limit to*
 2735 *which rollback may be performed*].

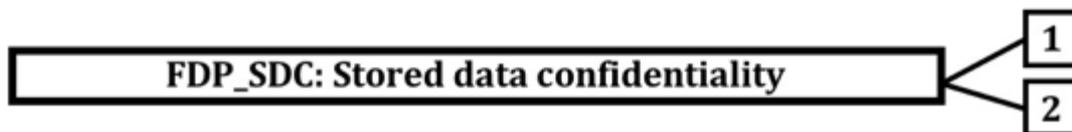
2736 **10.13 Stored data confidentiality (FDP_SDC)**

2737 **10.13.1 Family behaviour**

2738 This family provides requirements that address protection of user data confidentiality while the
 2739 data is stored within memory areas protected by the TSF. The TSF provides access to the data in
 2740 the memory through the specified interfaces only and prevents compromise of their
 2741 information bypassing these interfaces. It complements the family Stored data integrity
 2742 (FDP_SDI) which protects the user data from integrity errors while being stored in the memory.

2743 **10.13.2 Components leveling and description**

2744 Figure 34 shows the component leveling for this family.



2745 **Figure 34 — FDP_SDC: Component leveling**

2746 FDP_SDC.1 Stored data confidentiality, requires the TSF to protect the confidentiality of
 2747 information of the user data in specified memory areas.

2748 FDP_SDC.2 Stored data confidentiality with dedicated method, requires the TSF to protect the
 2749 confidentiality of the user data according to data characteristics leading to specify a dedicated
 2750 method of protection of confidentiality.

2751 **10.13.3 Management of FDP_SDC.1, FDP_SDC.2**

2752 The following actions **could** be considered for the management functions in FMT:

2753 a) No specific management functions are identified

2754 10.13.4 Audit of FDP_SDC.1, FDP_SDC.2

2755 The following actions should be auditable if FAU_GEN Security audit data generation is included
2756 in the PP, PP-Module, functional package or ST:

2757 a) There are no auditable events foreseen.

2758 10.13.5 FDP_SDC.1 Stored data confidentiality

2759 Component relationships

2760 Hierarchical to: No other components.

2761 Dependencies: No dependencies.

2762 FDP_SDC.1.1

2763 The TSF **shall** ensure the confidentiality of [selection: *all user data, the following user data*
2764 *[assignment: list of user data]*] while it is stored in the [selection: *temporary memory,*
2765 *persistent memory, any memory*].

2766 10.13.6 FDP_SDC.2 Stored data confidentiality with dedicated method

2767 Component relationships

2768 Hierarchical to: No other components.

2769 Dependencies: FCS_COP.1.

2770 FDP_SDC.2.1

2771 The TSF **shall** ensure the confidentiality of the [selection: *all user data, the following user*
2772 *data [assignment: list of user data]*] according to [assignment: *data characteristics*] while it
2773 is stored under the control of the TSF.

2774 FDP_SDC.2.2

2775 The TSF **shall** ensure the confidentiality of the user data specified in FDP_SDC.2.1 without
2776 user intervention.

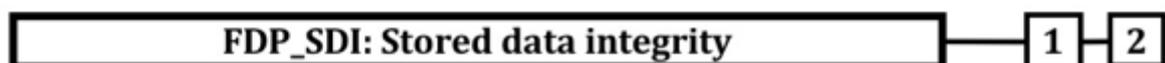
2777 10.14 Stored data integrity (FDP_SDI)

2778 10.14.1 Family behaviour

2779 This family provides requirements that address protection of user data while it is stored within
2780 containers controlled by the TSF. Integrity errors **may** affect user data stored in memory, or in a
2781 storage device. This family differs from Internal TOE transfer (FDP_ITT) which protects the user
2782 data from integrity errors while being transferred within the TOE.

2783 10.14.2 Components leveling and description

2784 Figure 35 shows the component leveling for this family.



2785 **Figure 35 — FDP_SDI: Component leveling**

2786 FDP_SDI.1 Stored data integrity monitoring, requires that the TSF monitor user data stored
2787 within containers controlled by the TSF for identified integrity errors.

2788 FDP_SDI.2 Stored data integrity monitoring and action adds the additional capability to the first
2789 component by allowing for actions to be taken as a result of an error detection.

2790 **10.14.3 Management of FDP_SDI.1**

2791 The following actions **could** be considered for the management functions in FMT:

2792 a) There are no management activities foreseen.

2793 **10.14.4 Management of FDP_SDI.2**

2794 The following actions **could** be considered for the management functions in FMT:

2795 a) The actions to be taken upon the detection of an integrity error **could** be
2796 configurable.

2797 **10.14.5 Audit of FDP_SDI.1**

2798 The following actions should be auditable if FAU_GEN Security audit data generation is included
2799 in the PP, PP-Module, functional package or ST:

2800 a) Minimal: Successful attempts to check the integrity of user data, including an
2801 indication of the results of the check.

2802 b) Basic: All attempts to check the integrity of user data, including an indication of the
2803 results of the check, if performed.

2804 c) Detailed: The type of integrity error that occurred.

2805 **10.14.6 Audit of FDP_SDI.2**

2806 The following actions should be auditable if FAU_GEN Security audit data generation is included
2807 in the PP, PP-Module, functional package or ST:

2808 a) Minimal: Successful attempts to check the integrity of user data, including an
2809 indication of the results of the check.

2810 b) Basic: All attempts to check the integrity of user data, including an indication of the
2811 results of the check, if performed.

2812 c) Detailed: The type of integrity error that occurred.

2813 d) Detailed: The action taken upon detection of an integrity error.

2814 **10.14.7 FDP_SDI.1 Stored data integrity monitoring**

2815 **Component relationships**

2816 Hierarchical to: No other components.

2817 Dependencies: No dependencies.

2818 **FDP_SDI.1.1**

2819 **The TSF shall monitor user data stored in containers controlled by the TSF for**
2820 **[assignment: integrity errors] on all objects, based on the following attributes:**
2821 **[assignment: user data attributes].**

2822 **10.14.8 FDP_SDI.2 Stored data integrity monitoring and action**

2823 Hierarchical to: FDP_SDI.1 Stored data integrity monitoring

2824 Dependencies: No dependencies.

2825 **FDP_SDI.2.1**

2826 The TSF **shall** monitor user data stored in containers controlled by the TSF for [assignment:
2827 *integrity errors*] on all objects, based on the following attributes: [assignment: *user data*
2828 *attributes*].

2829 **FDP_SDI.2.2**

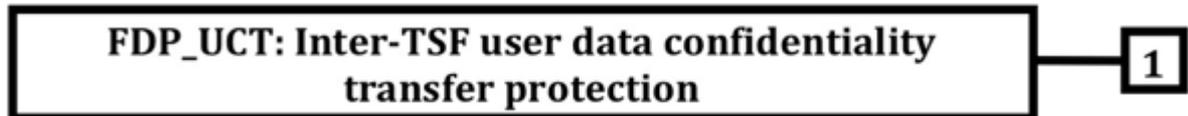
2830 **Upon detection of a data integrity error, the TSF shall** [assignment: *action to be taken*].

2831 **10.15 Inter-TSF user data confidentiality transfer protection (FDP_UCT)**2832 **10.15.1 Family behaviour**

2833 This family defines the requirements for ensuring the confidentiality of user data when it is
2834 transferred using an external channel between the TOE and another trusted IT product.

2835 **10.15.2 Components leveling and description**

2836 Figure 36 shows the component leveling for this family.



2837

2838 **Figure 36 — FDP_UCT: Component leveling**

2839 In FDP_UCT.1 Basic data exchange confidentiality, the goal is to provide protection from
2840 disclosure of user data while in transit.

2841 **10.15.3 Management of FDP_UCT.1**

2842 The following actions **could** be considered for the management functions in FMT:

2843 a) There are no management activities foreseen.

2844 **10.15.4 Audit of FDP_UCT.1**

2845 The following actions should be auditable if FAU_GEN Security audit data generation is included
2846 in the PP, PP-Module, functional package or ST:

2847 a) Minimal: The identity of any user or subject using the data exchange mechanisms.

2848 b) Basic: The identity of any unauthorized user or subject attempting to use the data
2849 exchange mechanisms.

2850 c) Basic: A reference to the names or other indexing information useful in identifying
2851 the user data that was transmitted or received. This **could** include security
2852 attributes associated with the information.

2853 **10.15.5 FDP_UCT.1 Basic data exchange confidentiality**2854 **Component relationships**

2855 Hierarchical to: No other components.

2856 Dependencies: [FTP_ITC.1 Inter-TSF trusted channel, or
2857 FTP_TRP.1 Trusted path]

2858 [FDP_ACC.1 Subset access control, or
2859 FDP_IFC.1 Subset information flow control]

2860 **FDP_UCT.1.1**

2861 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow***
 2862 ***control SFP(s)] to [selection: *transmit, receive*] user data in a manner protected from***
 2863 ***unauthorized disclosure.***

2864 **10.16 Inter-TSF user data integrity transfer protection (FDP_UIT)**

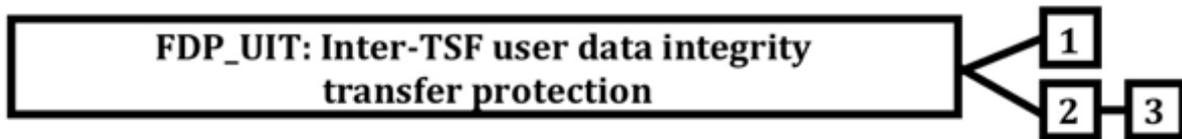
2865 **10.16.1 Family behaviour**

2866 This family defines the requirements for providing integrity for user data in transit between the
 2867 TOE and another trusted IT product and recovering from detectable errors. At a minimum, this
 2868 family monitors the integrity of user data for modifications. Furthermore, this family supports
 2869 different ways of correcting detected integrity errors.

2870 **10.16.2 Components leveling and description**

2871 Figure 37 shows the component leveling for this family.

2872



2873 **Figure 37 — FDP_UIT: Component leveling**

2874 FDP_UIT.1 Data exchange integrity addresses detection of modifications, deletions, insertions,
 2875 and replay errors of the user data transmitted.

2876 FDP_UIT.2 Source data exchange recovery addresses recovery of the original user data by the
 2877 receiving TSF with help from the source trusted IT product.

2878 FDP_UIT.3 Destination data exchange recovery addresses recovery of the original user data by
 2879 the receiving TSF on its own without any help from the source trusted IT product.

2880 **10.16.3 Management of FDP_UIT.1, FDP_UIT.2, FDP_UIT.3**

2881 The following actions could be considered for the management functions in FMT:

2882 a) There are no management activities foreseen.

2883 **10.16.4 Audit of FDP_UIT.1**

2884 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2885 in the PP, PP-Module, functional package or ST:

2886 a) Minimal: The identity of any user or subject using the data exchange mechanisms.

2887 b) Basic: The identity of any user or subject attempting to use the user data exchange
 2888 mechanisms, but who is unauthorized to do so.

2889 c) Basic: A reference to the names or other indexing information useful in identifying
 2890 the user data that was transmitted or received. This could include security
 2891 attributes associated with the user data.

2892 d) Basic: Any identified attempts to block transmission of user data.

2893 e) Detailed: The types and/or effects of any detected modifications of transmitted
 2894 user data.

2895 **10.16.5 Audit of FDP_UIT.2, FDP_UIT.3**

2896 The following actions should be auditable if FAU_GEN Security audit data generation is included
2897 in the PP, PP-Module, functional package or ST:

- 2898 a) Minimal: The identity of any user or subject using the data exchange mechanisms;
2899 b) Minimal: Successful recovery from errors including the type of error that was
2900 detected;
2901 c) Basic: The identity of any user or subject attempting to use the user data exchange
2902 mechanisms, but who is unauthorized to do so;
2903 d) Basic: A reference to the names or other indexing information useful in identifying
2904 the user data that was transmitted or received. This **could** include security
2905 attributes associated with the user data;
2906 e) Basic: Any identified attempts to block transmission of user data;
2907 f) Detailed: The types and/or effects of any detected modifications of transmitted
2908 user data.

2909 **10.16.6 FDP_UIT.1 Data exchange integrity**2910 **Component relationships**

2911	Hierarchical to:	No other components.
2912	Dependencies:	[FDP_ACC.1 Subset access control, or 2913 FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or 2914 FTP_TRP.1 Trusted path] 2915

2916 **FDP_UIT.1.1**

2917 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow***
2918 ***control SFP(s)*] to [selection: *transmit, receive*] user data in a manner protected from**
2919 **[selection: *modification, deletion, insertion, replay*] errors.**

2920 **FDP_UIT.1.2**

2921 **The TSF shall be able to determine on receipt of user data, whether [selection:**
2922 ***modification, deletion, insertion, replay*] has occurred.**

2923 **10.16.7 FDP_UIT.2 Source data exchange recovery**2924 **Component relationships**

2925	Hierarchical to:	No other components.
2926	Dependencies:	[FDP_ACC.1 Subset access control, or 2927 FDP_IFC.1 Subset information flow control] [FDP_UIT.1 Data exchange integrity, or 2928 FTP_ITC.1 Inter-TSF trusted channel] 2929

2930 **FDP_UIT.2.1**

2931 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow***
2932 ***control SFP(s)*] to be able to recover from [assignment: *list of recoverable errors*] with the**
2933 **help of the source trusted IT product.**

2934 **10.16.8 FDP_UIT.3 Destination data exchange recovery**

2935 Hierarchical to: FDP_UIT.2 Source data exchange recovery

2936 Dependencies: [FDP_ACC.1 Subset access control, or
2937 FDP_IFC.1 Subset information flow control]

2938 [FDP_UIT.1 Data exchange integrity, or
2939 FTP_ITC.1 Inter-TSF trusted channel]

2940 **FDP_UIT.3.1**

2941 The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow control*
2942 *SFP(s)*] to be able to recover from [assignment: *list of recoverable errors*] **without any help**
2943 **from** the source trusted IT product.

2944

2945 **11 Class FIA: Identification and authentication**

2946 **11.1 Class description**

2947 Families in this class address the requirements for functions to establish and verify a claimed
 2948 user identity.

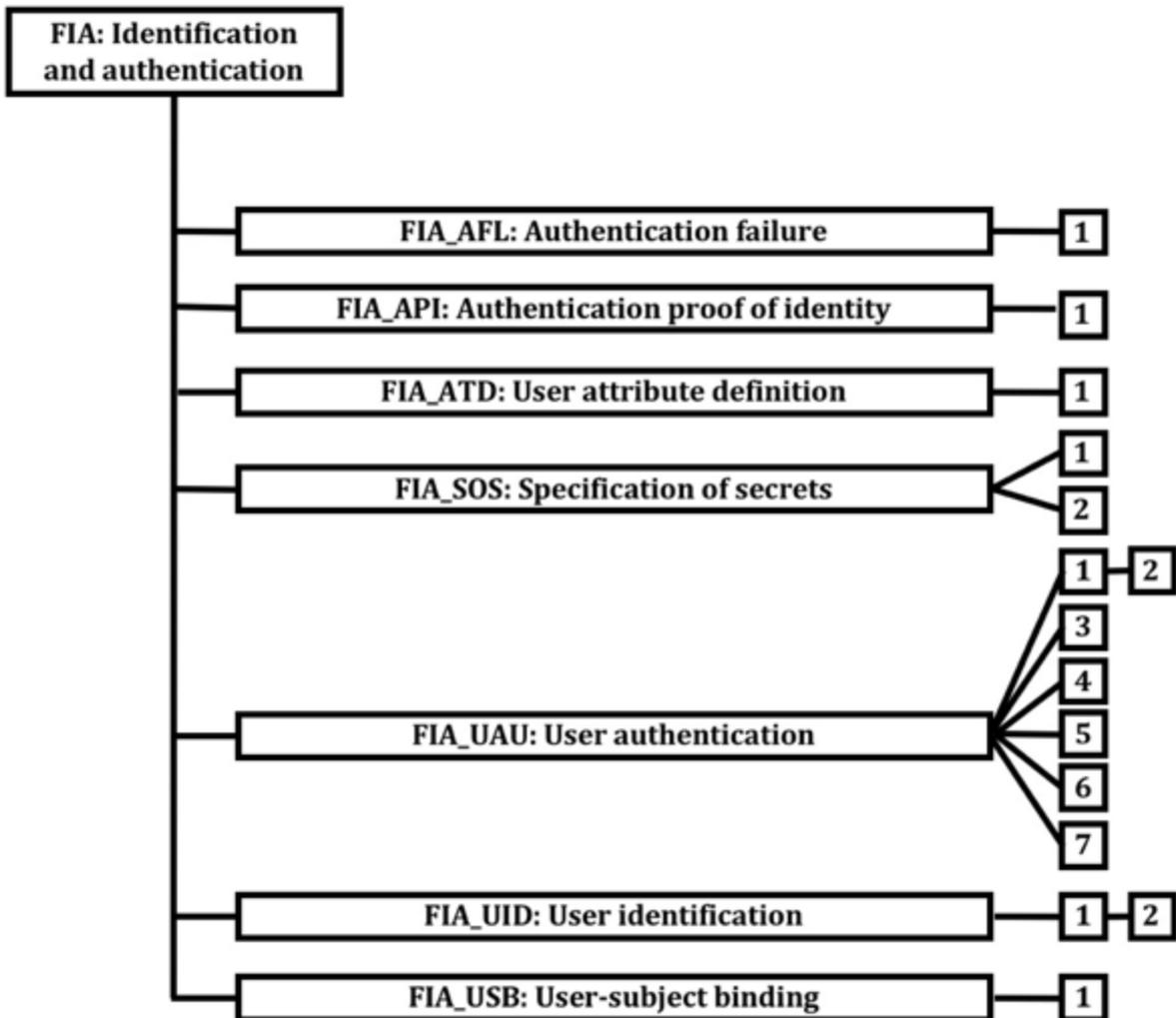
2949 Identification and authentication is required to ensure that users are associated with the proper
 2950 security attributes

2951 The unambiguous identification of authorized users and the correct association of security
 2952 attributes with users and subjects is critical to the enforcement of the intended security
 2953 policies. The families in this class deal with determining and verifying the identity of users,
 2954 determining their authority to interact with the TOE, and with the correct association of
 2955 security attributes for each authorized user. Other classes of requirements are dependent upon
 2956 correct identification and authentication of users in order to be effective.

2957 Figure 38 shows the decomposition of this class, it's families and components. Elements are not
 2958 shown in the figure.

2959 Annex G provides explanatory information for this class and **should** be consulted when using
 2960 the components identified in this class.

2961



2962 **Figure 38 — FIA: Identification and authentication class decomposition**

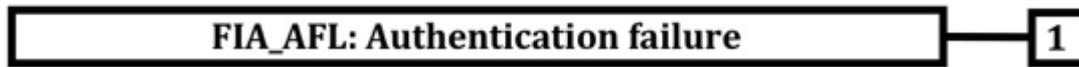
2963 **11.2 Authentication failures (FIA_AFL)**

2964 **11.2.1 Family behaviour**

2965 This family contains requirements for defining values for some number of unsuccessful
 2966 authentication attempts and TSF actions in cases of authentication attempt failures. Parameters
 2967 include, but are not limited to, the number of failed authentication attempts and time
 2968 thresholds.

2969 **11.2.2 Components leveling and description**

2970 Figure 39 shows the component leveling for this family.



2971

2972 **Figure 39 — FIA_AFL: Component leveling**

2973 FIA_AFL.1 Authentication failure handling, requires that the TSF be able to terminate the
 2974 session establishment process after a specified number of unsuccessful user authentication
 2975 attempts. It also requires that, after termination of the session establishment process, the TSF
 2976 be able to disable the user account or the point of entry from which the attempts were made
 2977 until an administrator-defined condition occurs.

2978 **11.2.3 Management of FIA_AFL.1**

2979 The following actions **could** be considered for the management functions in FMT:

- 2980 a) Management of the threshold for unsuccessful authentication attempts;
- 2981 b) Management of actions to be taken in the event of an authentication failure.

2982 **11.2.4 Audit of FIA_AFL.1**

2983 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2984 in the PP, PP-Module, functional package or ST:

- 2985 a) Minimal: the reaching of the threshold for the unsuccessful authentication attempts
 2986 and the actions taken and the subsequent, if appropriate, restoration to the normal
 2987 state.

2988 **11.2.5 FIA_AFL.1 Authentication failure handling**

2989 **Component relationships**

2990	Hierarchical to:	No other components.
2991	Dependencies:	FIA_UAU.1 Timing of authentication

2992 **FIA_AFL.1.1**

2993 **The TSF shall detect when [selection: [assignment: *positive integer number*], an**
 2994 **administrator configurable positive integer within [assignment: *range of acceptable***
 2995 **values]] unsuccessful authentication attempts occur related to [assignment: *list of***
 2996 ***authentication events*].**

2997 **FIA_AFL.1.2**

2998 **When the defined number of unsuccessful authentication attempts has been [selection:**
 2999 ***met, surpassed*], the TSF shall [assignment: *list of actions*].**

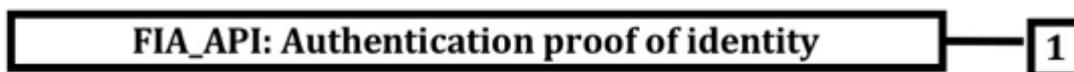
3000 **11.3 Authentication proof of identity (FIA_API)**

3001 **11.3.1 Family behaviour**

3002 This family defines functions provided by the TOE to prove its identity and so allow for
3003 verification of the TOE by an external entity in the TOE's IT environment.

3004 **11.3.2 Components leveling and description**

3005 Figure 40 shows the component leveling for this family.



3006 **Figure 40 — FIA_API: Component leveling**

3007 FIA_API.1 Authentication Proof of Identity, provides proof of the identity of the TOE to an
3008 external entity.

3009 **11.3.3 Management of FIA_API.1**

3010 The following actions **could** be considered for the management functions in FMT:

- 3011 a) Management of authentication information used to prove the claimed identity.

3012 **11.3.4 Audit of FIA_API.1**

3013 The following actions should be auditable if FAU_GEN Security audit data generation is included
3014 in the PP, PP-Module, functional package or ST:

- 3015 a) There are no auditable events foreseen.

3016 **11.3.5 FIA_API.1 Authentication proof of identity**

3017 **Component relationships**

3018 Hierarchical to: No other components.

3019 Dependencies: No dependencies.

3020 **FIA_API.1.1**

3021 The TSF **shall** provide an [assignment: *authentication mechanism*] to prove the identity of
3022 the [assignment: *object, authorized user, or role*] to an external entity.

3023 **Editors' Note:**

3024 Editors observe that in many STs using this component the second completion is for "TOE"
3025 which is neither an object, authorized user or a role. Should FIA_API.1.1 be updated to allow for
3026 the specification of TOE in the assignment?

3027 **11.4 User attribute definition (FIA_ATD)**

3028 **11.4.1 Family behaviour**

3029 All authorized users **may** have a set of security attributes, other than the user's identity, that is
3030 used to enforce the SFRs. This family defines the requirements for associating user security
3031 attributes with users as needed to support the TSF in making security decisions.

3032 **11.4.2 Components leveling and description**

3033 Figure 41 shows the component leveling for this family.



3034 **Figure 41 — FIA_ATD: Component leveling**

3035 FIA_ATD.1 User attribute definition, allows user security attributes for each user to be
3036 maintained individually.

3037 **11.4.3 Management of FIA_ATD.1**

3038 The following actions **could** be considered for the management functions in FMT:

- 3039 a) if so indicated in the assignment, the authorized administrator might be able to
3040 define additional security attributes for users.

3041 **11.4.4 Audit of FIA_ATD.1**

3042 The following actions should be auditable if FAU_GEN Security audit data generation is included
3043 in the PP, PP-Module, functional package or ST:

- 3044 a) There are no auditable events foreseen.

3045 **11.4.5 FIA_ATD.1 User attribute definition**

3046 **Component relationships**

- 3047 Hierarchical to: No other components.
- 3048 Dependencies: No dependencies.

3049 **FIA_ATD.1.1**

3050 The TSF **shall** maintain the following list of security attributes belonging to individual
3051 users: [assignment: *list of security attributes*].

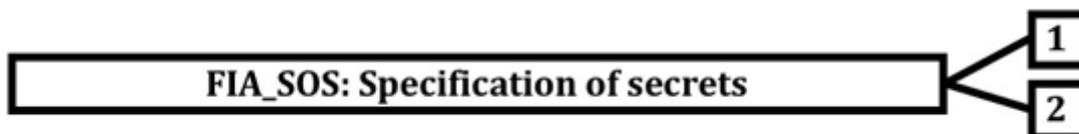
3052 **11.5 Specification of secrets (FIA_SOS)**

3053 **11.5.1 Family behaviour**

3054 This family defines requirements for mechanisms that enforce defined quality metrics on
3055 provided secrets and generate secrets to satisfy the defined metric.

3056 **11.5.2 Components leveling and description**

3057 Figure 42 shows the component leveling for this family.



3058 **Figure 42 — FIA_SOS: Component leveling**

3059 FIA_SOS.1 Verification of secrets, requires the TSF to verify that secrets meet defined quality
3060 metrics.

3061 FIA_SOS.2 TSF Generation of secrets, requires the TSF to be able to generate secrets that meet
3062 defined quality metrics.

3063 **11.5.3 Management of FIA_SOS.1**3064 The following actions **could** be considered for the management functions in FMT:

3065 a) the management of the metric used to verify the secrets.

3066 **11.5.4 Management of FIA_SOS.2**3067 The following actions **could** be considered for the management functions in FMT:

3068 a) the management of the metric used to generate the secrets.

3069 **11.5.5 Audit of FIA_SOS.1, FIA_SOS.2**3070 The following actions should be auditable if FAU_GEN Security audit data generation is included
3071 in the PP, PP-Module, functional package or ST:

3072 a) Minimal: Rejection by the TSF of any tested secret;

3073 b) Basic: Rejection or acceptance by the TSF of any tested secret;

3074 c) Detailed: Identification of any changes to the defined quality metrics.

3075 **11.5.6 FIA_SOS.1 Verification of secrets**3076 **Component relationships**

3077 Hierarchical to: No other components.

3078 Dependencies: No dependencies.

3079 **FIA_SOS.1.1**3080 **The TSF **shall** provide a mechanism to verify that secrets meet [assignment: *a defined***
3081 ***quality metric*].**3082 **11.5.7 FIA_SOS.2 TSF Generation of secrets**3083 **Component relationships**

3084 Hierarchical to: No other components.

3085 Dependencies: No dependencies.

3086 **FIA_SOS.2.1**3087 The TSF **shall** provide a mechanism to **generate** secrets that meet [assignment: *a defined*
3088 *quality metric*].3089 **FIA_SOS.2.2**3090 **The TSF **shall** be able to enforce the use of TSF generated secrets for [assignment: *list of***
3091 ***TSF functions*].**3092 **11.6 User authentication (FIA_UAU)**3093 **11.6.1 Family behaviour**3094 This family defines the types of user authentication mechanisms supported by the TSF. This
3095 family also defines the required attributes on which the user authentication mechanisms must
3096 be based.

3097 **11.6.2 Components leveling and description**

3098 Figure 43 shows the component leveling for this family.



3099 **Figure 43 — FIA_UAU: Component leveling**

3100 FIA_UAU.1 Timing of authentication, allows a user to perform certain actions prior to the
3101 authentication of the user's identity.

3102 FIA_UAU.2 User authentication before any action, requires that users are authenticated before
3103 any other action will be allowed by the TSF.

3104 FIA_UAU.3 Unforgeable authentication, requires the authentication mechanism to be able to
3105 detect and prevent the use of authentication data that has been forged or copied.

3106 FIA_UAU.4 Single-use authentication mechanisms, requires an authentication mechanism that
3107 operates with single-use authentication data.

3108 FIA_UAU.5 Multiple authentication mechanisms, requires that different authentication
3109 mechanisms be provided and used to authenticate user identities for specific events.

3110 FIA_UAU.6 Re-authenticating, requires the ability to specify events for which the user needs to
3111 be re-authenticated.

3112 FIA_UAU.7 Protected authentication feedback, requires that only limited feedback information
3113 is provided to the user during the authentication.

3114 **11.6.3 Management of FIA_UAU.1**

3115 The following actions **could** be considered for the management functions in FMT:

- 3116 a) management of the authentication data by an administrator;
- 3117 b) management of the authentication data by the associated user;
- 3118 c) managing the list of actions that **can** be taken before the user is authenticated.

3119 **11.6.4 Management of FIA_UAU.2**

3120 The following actions **could** be considered for the management functions in FMT:

- 3121 a) management of the authentication data by an administrator;
- 3122 b) management of the authentication data by the user associated with this data.

3123 **11.6.5 Management of FIA_UAU.3, FIA_UAU.4, FIA_UAU.7**

3124 The following actions **could** be considered for the management functions in FMT:

- 3125 a) There are no management activities foreseen.

3126 **11.6.6 Management of FIA_UAU.5**

3127 The following actions **could** be considered for the management functions in FMT:

- 3128 a) the management of authentication mechanisms;

3129 **11.6.7 Management of FIA_UAU.6**

3130 The following actions **could** be considered for the management functions in FMT:

- 3131 a) if an authorized administrator **could** request re-authentication, the management
3132 includes a re-authentication request.

3133 **11.6.8 Management of FIA_UAU.7**

3134 The following actions **could** be considered for the management functions in FMT:

- 3135 a) the management of the rules for authentication.

3136 **11.6.9 Audit of FIA_UAU.1**

3137 The following actions should be auditable if FAU_GEN Security audit data generation is included
3138 in the PP, PP-Module, functional package or ST:

- 3139 a) Minimal: Unsuccessful use of the authentication mechanism;
3140 b) Basic: All use of the authentication mechanism;
3141 c) Detailed: All TSF mediated actions performed before authentication of the user.

3142 **11.6.10 Audit of FIA_UAU.2**

3143 The following actions should be auditable if FAU_GEN Security audit data generation is included
3144 in the PP, PP-Module, functional package or ST:

- 3145 a) Minimal: Unsuccessful use of the authentication mechanism;
3146 b) Basic: All use of the authentication mechanism.

3147 **11.6.11 Audit of FIA_UAU.3**

3148 The following actions should be auditable if FAU_GEN Security audit data generation is included
3149 in the PP, PP-Module, functional package or ST:

- 3150 a) Minimal: Detection of fraudulent authentication data;
3151 b) Basic: All immediate measures taken and results of checks on the fraudulent data.

3152 **11.6.12 Audit of FIA_UAU.4**

3153 The following actions should be auditable if FAU_GEN Security audit data generation is included
3154 in the PP, PP-Module, functional package or ST:

- 3155 a) Minimal: Attempts to reuse authentication data.

3156 **11.6.13 Audit of FIA_UAU.5**

3157 The following actions should be auditable if FAU_GEN Security audit data generation is included
3158 in the PP, PP-Module, functional package or ST:

- 3159 a) Minimal: The final decision on authentication;
3160 b) Basic: The result of each activated mechanism together with the final decision.

3161 **11.6.14 Audit of FIA_UAU.6**

3162 The following actions should be auditable if FAU_GEN Security audit data generation is included
3163 in the PP, PP-Module, functional package or ST:

- 3164 a) Minimal: Failure of re-authentication;
- 3165 b) Basic: All re-authentication attempts.

3166 **11.6.15 Audit of FIA_UAU.7**

3167 The following actions should be auditable if FAU_GEN Security audit data generation is included
3168 in the PP, PP-Module, functional package or ST:

- 3169 a) Well-formedness of rules regarding the semantics of rule-set;
- 3170 b) Basic: verification of enforceability of rules ~~(and their writing)~~.

3171 **Editors' Note:**

3172 **A comment was received saying "b) should be changed to make it clearer."**

3173 **Comments have been requested on this for several drafts... If no comments are received in this**
3174 **CD3 comment round then the text "and their writing" will be removed**

3175 **11.6.16 FIA_UAU.1 Timing of authentication**

3176 **Component relationships**

- 3177 Hierarchical to: No other components.
- 3178 Dependencies: FIA_UID.1 Timing of identification

3179 **FIA_UAU.1.1**

3180 **The TSF shall allow [assignment: *list of TSF mediated actions*] on behalf of the user to be**
3181 **performed before the user is authenticated.**

3182 **FIA_UAU.1.2**

3183 **The TSF shall require each user to be successfully authenticated before allowing any**
3184 **other TSF-mediated actions on behalf of that user.**

3185 **11.6.17 FIA_UAU.2 User authentication before any action**

3186 **Component relationships**

- 3187 Hierarchical to: FIA_UAU.1 Timing of authentication
- 3188 Dependencies: FIA_UID.1 Timing of identification

3189 **FIA_UAU.2.1**

3190 **The TSF shall require each user to be successfully authenticated before allowing any other TSF-**
3191 **mediated actions on behalf of that user.**

3192 **11.6.18 FIA_UAU.3 Unforgeable authentication**

3193 **Component relationships**

- 3194 Hierarchical to: No other components.
- 3195 Dependencies: No dependencies.

3196 **FIA_UAU.3.1**

3197 **The TSF shall [selection: *detect, prevent*] use of authentication data that has been forged**
3198 **by any user of the TSF.**

3199 **FIA_UAU.3.2**

3200 **The TSF shall** [selection: *detect, prevent*] use of authentication data that has been copied
3201 **from any other user of the TSF.**

3202 **11.6.19 FIA_UAU.4 Single-use authentication mechanisms**3203 **Component relationships**

3204 Hierarchical to: No other components.

3205 Dependencies: No dependencies.

3206 **FIA_UAU.4.1**

3207 **The TSF shall** prevent reuse of authentication data related to [assignment: *identified*
3208 *authentication mechanism(s)*].

3209 **11.6.20 FIA_UAU.5 Multiple authentication mechanisms**3210 **Component relationships**

3211 Hierarchical to: No other components.

3212 Dependencies: No dependencies.

3213 **FIA_UAU.5.1**

3214 **The TSF shall** provide [assignment: *list of multiple authentication mechanisms*] to support
3215 **user authentication.**

3216 **FIA_UAU.5.2**

3217 **The TSF shall** authenticate any user's claimed identity according to the [assignment:
3218 *rules describing how the multiple authentication mechanisms provide authentication*].

3219 **11.6.21 FIA_UAU.6 Re-authenticating**3220 **Component relationships**

3221 Hierarchical to: No other components.

3222 Dependencies: No dependencies.

3223 **FIA_UAU.6.1**

3224 **The TSF shall** re-authenticate the user under the conditions [assignment: *list of*
3225 *conditions under which re-authentication is required*].

3226 **11.6.22 FIA_UAU.7 Protected authentication feedback**3227 **Component relationships**

3228 Hierarchical to: No other components.

3229 Dependencies: FIA_UAU.1 Timing of authentication

3230 **FIA_UAU.7.1**

3231 **The TSF shall** provide only [assignment: *list of feedback*] to the user while the
3232 **authentication is in progress.**

3233 **11.7 User identification (FIA_UID)**

3234 **11.7.1 Family behaviour**

3235 This family defines the conditions under which users shall be required to identify themselves
 3236 before performing any other actions that are to be mediated by the TSF and which require user
 3237 identification.

3238 **11.7.2 Components leveling and description**

3239 Figure 44 shows the component leveling for this family.



3240 **Figure 44 — FIA_UID: Component leveling**

3241 FIA_UID.1 Timing of identification, allows users to perform certain actions before being
 3242 identified by the TSF.

3243 FIA_UID.2 User identification before any action, requires that users identify themselves before
 3244 any action will be allowed by the TSF.

3245 **11.7.3 Management of FIA_UID.1**

3246 The following actions could be considered for the management functions in FMT:

- 3247 a) The management of the user identities;
- 3248 b) If an authorized administrator can change the actions allowed before identification,
 3249 the managing of the action lists.

3250 **11.7.4 Management of FIA_UID.2**

3251 The following actions could be considered for the management functions in FMT:

- 3252 a) The management of the user identities;

3253 **11.7.5 Audit of FIA_UID.1, FIA_UID.2**

3254 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3255 in the PP, PP-Module, functional package or ST:

- 3256 a) Minimal: Unsuccessful use of the user identification mechanism, including the user
 3257 identity provided;
- 3258 b) Basic: All use of the user identification mechanism, including the user identity
 3259 provided.

3260 **11.7.6 FIA_UID.1 Timing of identification**

3261 **Component relationships**

3262 Hierarchical to: No other components.

3263 Dependencies: No dependencies.

3264 **FIA_UID.1.1**

3265 **The TSF shall allow [assignment: list of TSF-mediated actions] on behalf of the user to be**
 3266 **performed before the user is identified.**

3267 **FIA_UID.1.2**

3268 **The TSF shall require each user to be successfully identified before allowing any TSF-**
 3269 **mediated actions on behalf of that user.**

3270 **11.7.7 FIA_UID.2 User identification before any action**

3271 Hierarchical to: FIA_UID.1 Timing of identification

3272 Dependencies: No dependencies.

3273 **FIA_UID.2.1**

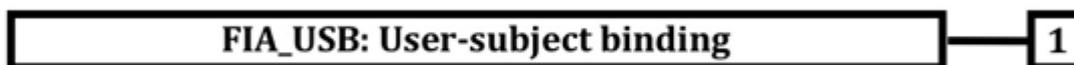
3274 The TSF shall require each user to be successfully identified before allowing any TSF-mediated
 3275 actions on behalf of that user.

3276 **11.8 User-subject binding (FIA_USB)**3277 **11.8.1 Family behaviour**

3278 An authenticated user, in order to use the TOE, typically activates a subject. The user's security
 3279 attributes are associated (totally or partially) with this subject. This family defines
 3280 requirements to create and maintain the association of the user's security attributes to a subject
 3281 acting on the user's behalf.

3282 **11.8.2 Components leveling and description**

3283 Figure 45 shows the component leveling for this family.



3284 **Figure 45 — FIA_USB: Component leveling**

3285 FIA_USB.1 User-subject binding, requires the specification of any rules governing the
 3286 association between user attributes and the subject attributes into which they are mapped.

3287 **11.8.3 Management of FIA_USB.1**

3288 The following actions could be considered for the management functions in FMT:

3289 a) An authorized administrator can define default subject security attributes;

3290 b) An authorized administrator can change subject security attributes.

3291 **11.8.4 Audit of FIA_USB.1**

3292 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3293 in the PP, PP-Module, functional package or ST:

3294 a) Minimal: Unsuccessful binding of user security attributes to a subject

3295 b) Basic: Success and failure of binding of user security attributes to a subject.

3296 **11.8.5 FIA_USB.1 User-subject binding**3297 **Component relationships**

3298 Hierarchical to: No other components.

3299 Dependencies: FIA_ATD.1 User attribute definition

3300 FIA_USB.1.1

3301 The TSF **shall** associate the following user security attributes with subjects acting on the
3302 behalf of that user: [assignment: *list of user security attributes*].

3303 FIA_USB.1.2

3304 The TSF **shall** enforce the following rules on the initial association of user security
3305 attributes with subjects acting on the behalf of users: [assignment: *rules for the initial*
3306 *association of attributes*].

3307 FIA_USB.1.3

3308 The TSF **shall** enforce the following rules governing changes to the user security
3309 attributes associated with subjects acting on the behalf of users: [assignment: *rules for*
3310 *the changing of attributes*].

3311

3312 **12 Class FMT: Security management**

3313 **12.1 Class description**

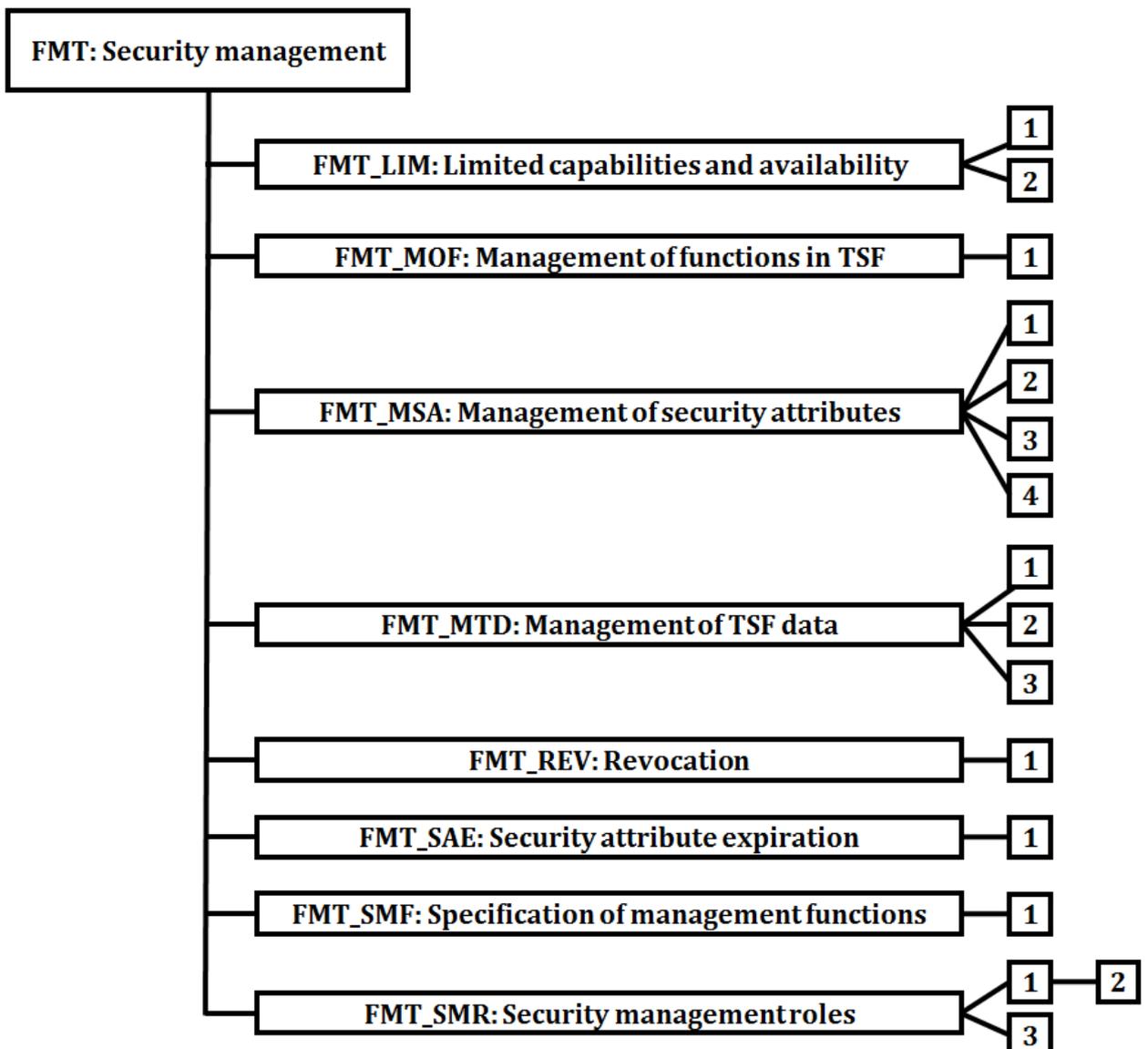
3314 This class is intended to specify the management of several aspects of the TSF: security
 3315 attributes, TSF data and functions. The different management roles and their interaction, such
 3316 as separation of capability, **can** be specified.

3317 This class has several objectives:

- 3318 a) Management of TSF data;
- 3319 b) Management of security attributes;
- 3320 c) Management of functions of the TSF;
- 3321 d) Definition of security roles.

3322 Figure 46 shows the decomposition of this class, it's families and components. Elements are not
 3323 shown in the figure.

3324 Annex H provides explanatory information for this class and **should** be consulted when using
 3325 the components identified in this class.



3326

3327 **Figure 46 — FMT: Security management class decomposition**

3328 **12.2 Limited capabilities and availability (FMT_LIM)**

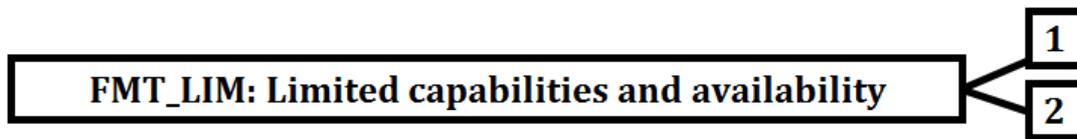
3329 **12.2.1 Family behaviour**

3330 This family defines requirements that limit the capabilities and availability of functions in a
3331 combined manner.

3332 Note FDP_ACF restricts the access to functions whereas the component Limited Capability of this family
3333 requires the functions themselves to be designed in a specific manner.

3334 **12.2.2 Components leveling and description**

3335 Figure 47 shows the component leveling for this family.



3336

3337 **Figure 47 — FMT_LIM: Component leveling**

3338 FMT_LIM.1 Limited capabilities requires that the TSF is built to provide only the capabilities
3339 (perform action, gather information) necessary for its genuine purpose.

3340 FMT_LIM.2 Limited availability requires that the TSF restrict the use of functions (refer to
3341 Limited capabilities (FMT_LIM.1)). This **can** be achieved, for instance, by removing or by
3342 disabling functions in a specific phase of the TOE's life-cycle.

3343 **12.2.3 Management of FMT_LIM.1, FMT_LIM.2**

3344 The following actions **could** be considered for the management functions in FMT:

- 3345 a) There are no management activities foreseen.

3346 **12.2.4 Audit of FMT_LIM.1**

3347 The following actions should be auditable if FAU_GEN Security audit data generation is included
3348 in the PP, PP-Module, functional package or ST:

- 3349 a) There are no auditable events foreseen.

3350 **12.2.5 FMT_LIM.1 Limited capabilities**

3351 **Component relationships**

3352 Hierarchical to: No other components.

3353 Dependencies: FMT_LIM.2 Limited availability

3354 **FMT_LIM.1.1**

3355 **The TSF shall limit its capabilities so that in conjunction with “Limited availability**
3356 **(FMT_LIM.2)” the following policy is enforced [assignment: Limited capability and**
3357 **availability policy].**

3358 **12.2.6 FMT_LIM.2 Limited availability**

3359 **Component relationships**

3360 Hierarchical to: No other components.

3361 Dependencies: FMT_LIM.1 Limited capabilities

3362 FMT_LIM.2.1

3363 The TSF **shall** be designed in a manner that limits its availability so that in conjunction
3364 with “Limited capabilities (FMT_LIM.1)” the following policy is enforced [assignment:
3365 *Limited capability and availability policy*].

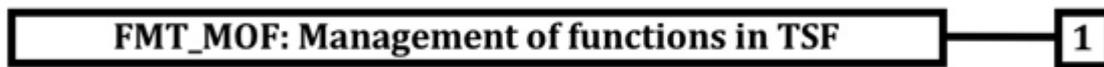
3366 12.3 Management of functions in TSF (FMT_MOF)

3367 12.3.1 Family behaviour

3368 This family allows authorized users to control over the management of functions in the TSF.

3369 12.3.2 Components leveling and description

3370 Figure 48 shows the component leveling for this family.



3371 **Figure 48 — FMT_MOF: Component leveling**

3372 FMT_MOF.1 Management of security functions behaviour allows the authorized users (roles) to
3373 manage the behaviour of functions in the TSF that use rules or have specified conditions that
3374 **may** be manageable.

3375 12.3.3 Management of FMT_MOF.1

3376 The following actions **could** be considered for the management functions in FMT:

3377 a) managing the group of roles that **can** interact with the functions in the TSF.

3378 12.3.4 Audit of FMT_MOF.1

3379 The following actions should be auditable if FAU_GEN Security audit data generation is included
3380 in the PP, PP-Module, functional package or ST:

3381 a) Basic: All modifications in the behaviour of the functions in the TSF.

3382 12.3.5 FMT_MOF.1 Management of security functions behaviour

3383 Component relationships

3384 Hierarchical to: No other components.

3385 Dependencies: FMT_SMR.1 Security roles

3386 FMT_SMF.1 Specification of Management Functions

3387 FMT_MOF.1.1

3388 The TSF **shall** restrict the ability to [selection: *determine the behaviour of, disable, enable,*
3389 *modify the behaviour of*] the functions [assignment: *list of functions*] to [assignment: *the*
3390 *authorized identified roles*].

3391 12.4 Management of security attributes (FMT_MSA)

3392 12.4.1 Family behaviour

3393 This family allows authorized users control over the management of security attributes. This
3394 management might include capabilities for viewing and modifying of security attributes.

3395 **12.4.2 Components leveling and description**

3396 Figure 49 shows the component leveling for this family.



3397 **Figure 49 — FMT_MSA: Component leveling**

3398 FMT_MSA.1 Management of security attributes allows authorized users (roles) to manage the
3399 specified security attributes.

3400 FMT_MSA.2 Secure security attributes ensures that values assigned to security attributes are
3401 valid with respect to the secure state.

3402 FMT_MSA.3 Static attribute ensures that the default values of security attributes are
3403 appropriately either permissive or restrictive in nature.

3404 FMT_MSA.4 Security attribute value inheritance allows the rules/policies to be specified that
3405 will dictate the value to be inherited by a security attribute.

3406 **12.4.3 Management of FMT_MSA.1**

3407 The following actions **could** be considered for the management functions in FMT:

- 3408 a) Managing the group of roles that **can** interact with the security attributes;
- 3409 b) Management of rules by which security attributes inherit specified values.

3410 **12.4.4 Management of FMT_MSA.2**

3411 The following actions **could** be considered for the management functions in FMT:

- 3412 a) Management of rules by which security attributes inherit specified values.

3413 **12.4.5 Management of FMT_MSA.3**

3414 The following actions **could** be considered for the management functions in FMT:

- 3415 a) Managing the group of roles that **can** specify initial values;
- 3416 b) Managing the permissive or restrictive setting of default values for a given access
3417 control SFP;
- 3418 c) Management of rules by which security attributes inherit specified values.

3419 **12.4.6 Management of FMT_MSA.4**

3420 The following actions **could** be considered for the management functions in FMT:

- 3421 a) Specification of the role permitted to establish or modify security attributes.

3422 **12.4.7 Audit of FMT_MSA.1**

3423 The following actions should be auditable if FAU_GEN Security audit data generation is included
3424 in the PP, PP-Module, functional package or ST:

- 3425 a) Basic: All modifications of the values of security attributes.

3426 **12.4.8 Audit of FMT_MSA.2**

3427 The following actions should be auditable if FAU_GEN Security audit data generation is included
3428 in the PP, PP-Module, functional package or ST:

3429 a) Minimal: All offered and rejected values for a security attribute.

3430 b) Detailed: All offered and accepted secure values for a security attribute.

3431 **12.4.9 Audit of FMT_MSA.3**

3432 The following actions should be auditable if FAU_GEN Security audit data generation is included
3433 in the PP, PP-Module, functional package or ST:

3434 a) Basic: Modifications of the default setting of permissive or restrictive rules.

3435 b) Basic: All modifications of the initial values of security attributes.

3436 **12.4.10 Audit of FMT_MSA.4**

3437 The following actions should be auditable if FAU_GEN Security audit data generation is included
3438 in the PP, PP-Module, functional package or ST:

3439 a) Basic: Modifications of security attributes, possibly with the old and/or values of
3440 security attributes that were modified.

3441 **12.4.11 FMT_MSA.1 Management of security attributes**3442 **Component relationships**

3443 Hierarchical to: No other components.

3444 Dependencies: [FDP_ACC.1 Subset access control, or
3445 FDP_IFC.1 Subset information flow control]

3446 FMT_SMR.1 Security roles

3447 FMT_SMF.1 Specification of Management Functions

3448 **FMT_MSA.1.1**

3449 **The TSF shall enforce the [assignment: *access control SFP(s), information flow control***
3450 ***SFP(s)] to restrict the ability to [selection: *change_default, query, modify, delete,****
3451 ***[assignment: other operations]] the security attributes [assignment: *list of security****
3452 ***attributes] to [assignment: *the authorized identified roles].****

3453 **12.4.12 FMT_MSA.2 Secure security attributes**3454 **Component relationships**

3455 Hierarchical to: No other components.

3456 Dependencies: [FDP_ACC.1 Subset access control, or
3457 FDP_IFC.1 Subset information flow control]

3458 FMT_MSA.1 Management of security attributes

3459 FMT_SMR.1 Security roles

3460 **FMT_MSA.2.1**

3461 **The TSF shall ensure that only secure values are accepted for [assignment: *list of security***
3462 ***attributes].***

3463 **12.4.13 FMT_MSA.3 Static attribute initialization**

3464 **Component relationships**

3465	Hierarchical to:	No other components.
3466	Dependencies:	FMT_MSA.1 Management of security attributes
3467		FMT_SMR.1 Security roles

3468 **FMT_MSA.3.1**

3469 **The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*]**
 3470 **to provide [selection, choose one of: *restrictive, permissive, [assignment: other property]*]**
 3471 **default values for security attributes that are used to enforce the SFP.**

3472 **FMT_MSA.3.2**

3473 **The TSF shall allow the [assignment: *the authorized identified roles*] to specify alternative**
 3474 **initial values to override the default values when an object or information is created.**

3475 **12.4.14 FMT_MSA.4 Security attribute value inheritance**

3476 **Component relationships**

3477	Hierarchical to:	No other components.
3478	Dependencies:	[FDP_ACC.1 Subset access control, or
3479		FDP_IFC.1 Subset information flow control]

3480 **FMT_MSA.4.1**

3481 **The TSF shall use the following rules to set the value of security attributes: [assignment:**
 3482 ***rules for setting the values of security attributes*].**

3483 **12.5 Management of TSF data (FMT_MTD)**

3484 **12.5.1 Family behaviour**

3485 This family allows authorized users (roles) control over the management of TSF data.

3486 **12.5.2 Components leveling and description**

3487 Figure 50 shows the component leveling for this family.



3488 **Figure 50 — FMT_MTD: Component leveling**

3489 FMT_MTD.1 Management of TSF data allows authorized users to manage TSF data.

3490 FMT_MTD.2 Management of limits on TSF data specifies the action to be taken if limits on TSF
 3491 data are reached or exceeded.

3492 FMT_MTD.3 Secure TSF data ensures that values assigned to TSF data are valid with respect to
 3493 the secure state.

3494 **12.5.3 Management of FMT_MTD.1**3495 The following actions **could** be considered for the management functions in FMT:3496 a) Managing the group of roles that **can** interact with the TSF data.3497 **12.5.4 Management of FMT_MTD.2**3498 The following actions **could** be considered for the management functions in FMT:3499 a) Managing the group of roles that **can** interact with the limits on the TSF data.3500 **12.5.5 Management of FMT_MTD.3**3501 The following actions **could** be considered for the management functions in FMT:

3502 a) There are no management activities foreseen.

3503 **12.5.6 Audit of FMT_MTD.1**3504 The following actions should be auditable if FAU_GEN Security audit data generation is included
3505 in the PP, PP-Module, functional package or ST:

3506 a) Basic: All modifications to the values of TSF data.

3507 **12.5.7 Audit of FMT_MTD.2**3508 The following actions should be auditable if FAU_GEN Security audit data generation is included
3509 in the PP, PP-Module, functional package or ST:

3510 a) Basic: All modifications to the limits on TSF data.

3511 b) Basic: All modifications in the actions to be taken in case of violation of the limits.

3512 **12.5.8 Audit of FMT_MTD.3**3513 The following actions should be auditable if FAU_GEN Security audit data generation is included
3514 in the PP, PP-Module, functional package or ST:

3515 a) Minimal: All rejected values of TSF data.

3516 **12.5.9 FMT_MTD.1 Management of TSF data**3517 **Component relationships**

3518 Hierarchical to: No other components.

3519 Dependencies: FMT_SMR.1 Security roles

3520 FMT_SMF.1 Specification of Management Functions

3521 **FMT_MTD.1.1**3522 **The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear,***
3523 ***assignment: other operations*]] the [assignment: *list of TSF data*] to [assignment: *the***
3524 ***authorized identified roles*].**3525 **12.5.10 FMT_MTD.2 Management of limits on TSF data**3526 **Component relationships**

3527 Hierarchical to: No other components.

3528 Dependencies: FMT_MTD.1 Management of TSF data

3529 FMT_SMR.1 Security roles

3530 **FMT_MTD.2.1**

3531 **The TSF shall restrict the specification of the limits for [assignment: *list of TSF data*] to**
 3532 **[assignment: *the authorized identified roles*].**

3533 **FMT_MTD.2.2**

3534 **The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated**
 3535 **limits: [assignment: *actions to be taken*].**

3536 **12.5.11 FMT_MTD.3 Secure TSF data**

3537 **Component relationships**

3538 Hierarchical to: No other components.

3539 Dependencies: FMT_MTD.1 Management of TSF data

3540 **FMT_MTD.3.1**

3541 **The TSF shall ensure that only secure values are accepted for [assignment: *list of TSF***
 3542 ***data*].**

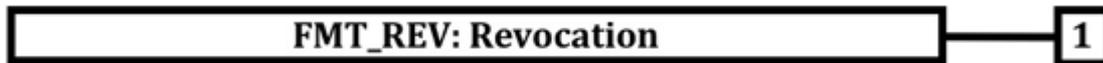
3543 **12.6 Revocation (FMT_REV)**

3544 **12.6.1 Family behaviour**

3545 This family addresses revocation of security attributes for a variety of entities within a TOE.

3546 **12.6.2 Components leveling and description**

3547 Figure 51 shows the component leveling for this family.



3548 **Figure 51 — FMT_REV: Component leveling**

3549 FMT_REV.1 Revocation provides for revocation of security attributes to be enforced at some
 3550 point in time.

3551 **12.6.3 Management of FMT_REV.1**

3552 The following actions could be considered for the management functions in FMT:

- 3553 a) Managing the group of roles that can invoke revocation of security attributes;
- 3554 b) Managing the lists of users, subjects, objects, and other resources for which
 3555 revocation is possible;
- 3556 c) Managing the revocation rules.

3557 **12.6.4 Audit of FMT_REV.1**

3558 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3559 in the PP, PP-Module, functional package or ST:

- 3560 a) Minimal: Unsuccessful revocation of security attributes;
- 3561 b) Basic: All attempts to revoke security attributes.

3562 **12.6.5 FMT_REV.1 Revocation**3563 **Component relationships**

3564 Hierarchical to: No other components.

3565 Dependencies: FMT_SMR.1 Security roles

3566 **FMT_REV.1.1**

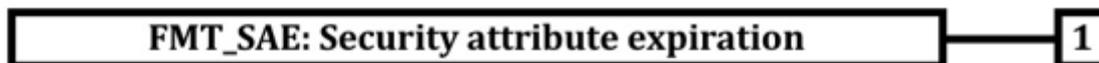
3567 The TSF **shall** restrict the ability to revoke [assignment: *list of security attributes*]
 3568 associated with the [selection: *users, subjects, objects, [assignment: other additional*
 3569 *resources*]] under the control of the TSF to [assignment: *the authorized identified roles*].

3570 **FMT_REV.1.2**3571 The TSF **shall** enforce the rules [assignment: *specification of revocation rules*].3572 **12.7 Security attribute expiration (FMT_SAE)**3573 **12.7.1 Family behaviour**

3574 This family addresses the capability to enforce time limits for the validity of security attributes.

3575 **12.7.2 Components leveling and description**

3576 Figure 52 shows the component leveling for this family.

3577 **Figure 52 — FMT_SAE: Component leveling**

3578 FMT_SAE.1 Time-limited authorization provides the capability for an authorized user to specify
 3579 an expiration time on specified security attributes.

3580 **12.7.3 Management of FMT_SAE.1**3581 The following actions **could** be considered for the management functions in FMT:

- 3582 a) Managing the list of security attributes for which expiration is to be supported;
- 3583 b) The actions to be taken if the expiration time has passed.

3584 **12.7.4 Audit of FMT_SAE.1**

3585 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3586 in the PP, PP-Module, functional package or ST:

- 3587 a) Basic: Specification of the expiration time for an attribute;
- 3588 b) Basic: Action taken due to attribute expiration.

3589 **12.7.5 FMT_SAE.1 Time-limited authorization**3590 **Component relationships**

3591 Hierarchical to: No other components.

3592 Dependencies: FMT_SMR.1 Security roles

3593 FPT_STM.1 Reliable time stamps

3594 **FMT_SAE.1.1**

3595 **The TSF shall restrict the capability to specify an expiration time for [assignment: *list of***
 3596 ***security attributes for which expiration is to be supported*] to [assignment: *the authorized***
 3597 ***identified roles*].**

3598 **FMT_SAE.1.2**

3599 **For each of these security attributes, the TSF shall be able to [assignment: *list of actions***
 3600 ***to be taken for each security attribute*] after the expiration time for the indicated security**
 3601 **attribute has passed.**

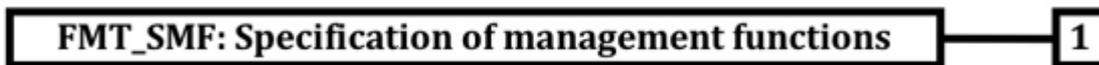
3602 **12.8 Specification of Management Functions (FMT_SMF)**

3603 **12.8.1 Family behaviour**

3604 This family allows the specification of the management functions to be provided by the TOE.
 3605 Management functions provide TSFI that allow administrators to define the parameters that
 3606 control the operation of security-related aspects of the TOE, such as data protection attributes,
 3607 TOE protection attributes, audit attributes, and identification and authentication attributes.
 3608 Management functions also include those functions performed by an operator to ensure
 3609 continued operation of the TOE, such as backup and recovery. This family works in conjunction
 3610 with the other components in the FMT: Security management class: the component in this
 3611 family calls out the management functions, and other families in FMT: Security management
 3612 restrict the ability to use these management functions.

3613 **12.8.2 Components leveling and description**

3614 Figure 53 shows the component leveling for this family.



3615 **Figure 53 — FMT_SMF: Component leveling**

3616 FMT_SMF.1 Specification of Management Functions requires that the TSF provide specific
 3617 management functions.

3618 **12.8.3 Management of FMT_SMF.1**

3619 The following actions could be considered for the management functions in FMT:

- 3620 a) There are no management activities foreseen.

3621 **12.8.4 Audit of FMT_SMF.1**

3622 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3623 in the PP, PP-Module, functional package or ST:

- 3624 a) Minimal: Use of the management functions.

3625 **12.8.5 FMT_SMF.1 Specification of Management Functions**

3626 **Component relationships**

3627 Hierarchical to: No other components.

3628 Dependencies: No dependencies.

3629 **FMT_SMF.1.1**

3630 **The TSF shall be capable of performing the following management functions:**
 3631 **[assignment: list of management functions to be provided by the TSF].**

3632 **12.9 Security management roles (FMT_SMR)**3633 **12.9.1 Family behaviour**

3634 This family is intended to control the assignment of different roles to users. The capabilities of
 3635 these roles with respect to security management are described in the other families in this class.

3636 **12.9.2 Components leveling and description**

3637 Figure 54 shows the component leveling for this family.



3638 **Figure 54 — FMT_SMR: Component leveling**

3639 FMT_SMR.1 Security roles specifies the roles with respect to security that the TSF recognizes.

3640 FMT_SMR.2 Restrictions on security roles specifies that in addition to the specification of the
 3641 roles, there are rules that control the relationship between the roles.

3642 FMT_SMR.3 Assuming roles, requires that an explicit request is given to the TSF to assume a
 3643 role.

3644 **12.9.3 Management of FMT_SMR.1**

3645 The following actions could be considered for the management functions in FMT:

3646 a) Managing the group of users that are part of a role.

3647 **12.9.4 Management of FMT_SMR.2**

3648 The following actions could be considered for the management functions in FMT:

3649 a) Managing the group of users that are part of a role;

3650 b) Managing the conditions that the roles must satisfy.

3651 **12.9.5 Management of FMT_SMR.3**

3652 There are no management activities foreseen.

3653 **12.9.6 Audit of FMT_SMR.1**

3654 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3655 in the PP, PP-Module, functional package or ST:

3656 a) Minimal: modifications to the group of users that are part of a role;

3657 b) Detailed: every use of the rights of a role.

3658 **12.9.7 Audit of FMT_SMR.2**

3659 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3660 in the PP, PP-Module, functional package or ST:

3661 a) Minimal: modifications to the group of users that are part of a role;

- 3662 b) Minimal: unsuccessful attempts to use a role due to the given conditions on the
3663 roles;
3664 c) Detailed: every use of the rights of a role.

3665 **12.9.8 Audit of FMT_SMR.3**

3666 The following actions should be auditable if FAU_GEN Security audit data generation is included
3667 in the PP, PP-Module, functional package or ST:

- 3668 a) Minimal: explicit request to assume a role.

3669 **12.9.9 FMT_SMR.1 Security roles**

3670 **Component relationships**

- 3671 Hierarchical to: No other components.
3672 Dependencies: FIA_UID.1 Timing of identification

3673 **FMT_SMR.1.1**

3674 **The TSF shall maintain the roles [assignment: *the authorized identified roles*].**

3675 **FMT_SMR.1.2**

3676 **The TSF shall be able to associate users with roles.**

3677 **12.9.10 FMT_SMR.2 Restrictions on security roles**

3678 **Component relationships**

- 3679 Hierarchical to: FMT_SMR.1 Security roles
3680 Dependencies: FIA_UID.1 Timing of identification

3681 **FMT_SMR.2.1**

3682 The TSF shall maintain the roles: [assignment: *authorized identified roles*].

3683 **FMT_SMR.2.2**

3684 The TSF shall be able to associate users with roles.

3685 **FMT_SMR.2.3**

3686 **The TSF shall ensure that the conditions [assignment: *conditions for the different roles*]
3687 are satisfied.**

3688 **12.9.11 FMT_SMR.3 Assuming roles**

- 3689 Hierarchical to: No other components.
3690 Dependencies: FMT_SMR.1 Security roles

3691 **FMT_SMR.3.1**

3692 **The TSF shall require an explicit request to assume the following roles: [assignment: *the*
3693 *roles*].**

3694

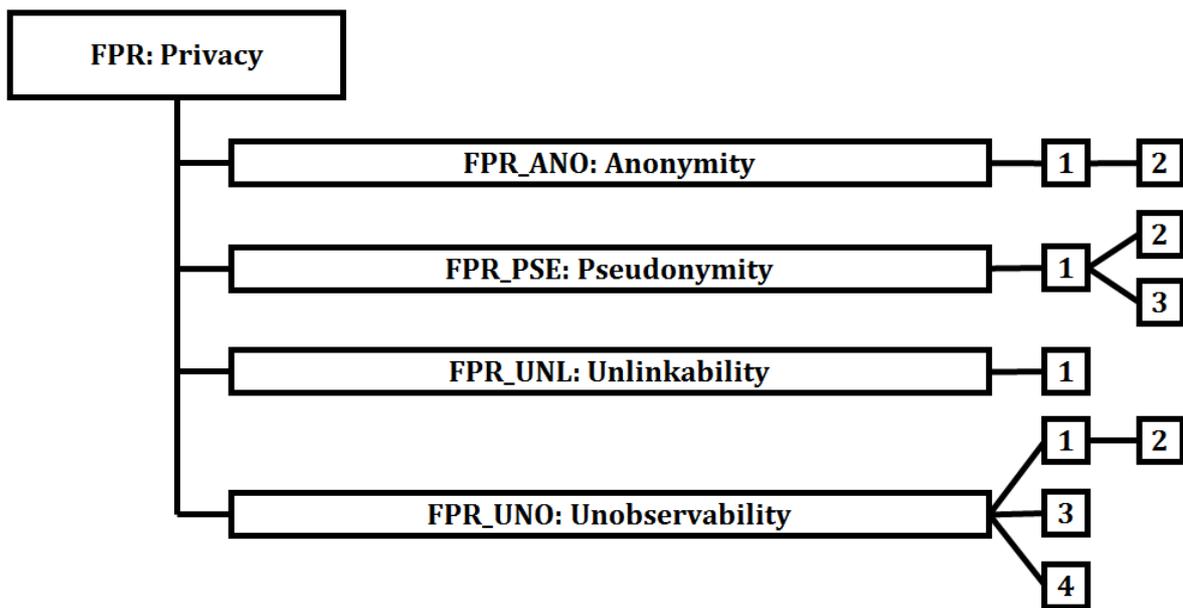
3695 **13 Class FPR: Privacy**

3696 **13.1 Class description**

3697 This class contains privacy requirements. These requirements provide a user protection against
3698 discovery and misuse of identity by other users.

3699 Figure 55 shows the decomposition of this class, it's families and components. Elements are not
3700 shown in the figure.

3701 Annex I provides explanatory information for this class and **should** be consulted when using the
3702 components identified in this class.



3703

3704

Figure 55 — FPR: Privacy class decomposition

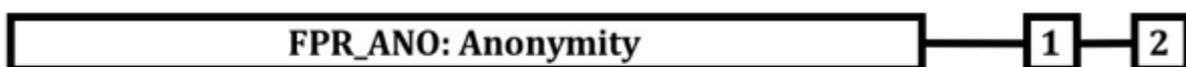
3705 **13.2 Anonymity (FPR_ANO)**

3706 **13.2.1 Family behaviour**

3707 This family ensures that a user **may** use a resource or service without disclosing the user's
3708 identity. The requirements for Anonymity provide protection of the user identity. Anonymity is
3709 not intended to protect the subject identity.

3710 **13.2.2 Components leveling and description**

3711 Figure 56 shows the component leveling for this family.



3712

Figure 56 — FPR_ANO: Component leveling

3713 FPR_ANO.1 Anonymity, requires that other users or subjects are unable to determine the
3714 identity of a user bound to a subject or operation.

3715 FPR_ANO.2 Anonymity without soliciting information enhances the requirements of FPR_ANO.1
3716 Anonymity by ensuring that the TSF does not ask for the user identity.

3717 **13.2.3 Management of FPR_ANO.1, FPR_ANO.2**

3718 The following actions **could** be considered for the management functions in FMT:

3719 a) There are no management activities foreseen.

3720 **13.2.4 Audit of FPR_ANO.1, FPR_ANO.2**

3721 The following actions should be auditable if FAU_GEN Security audit data generation is included
3722 in the PP, PP-Module, functional package or ST:

3723 a) Minimal: The invocation of the anonymity mechanism.

3724 **13.2.5 FPR_ANO.1 Anonymity**

3725 **Component relationships**

3726 Hierarchical to: No other components.

3727 Dependencies: No dependencies.

3728 **FPR_ANO.1.1**

3729 The TSF **shall** ensure that [assignment: *set of users and/or subjects*] are unable to
3730 determine the real user name bound to [assignment: *list of subjects and/or operations*
3731 *and/or objects*].

3732 **13.2.6 FPR_ANO.2 Anonymity without soliciting information**

3733 **Component relationships**

3734 Hierarchical to: FPR_ANO.1 Anonymity

3735 Dependencies: No dependencies.

3736 **FPR_ANO.2.1**

3737 The TSF **shall** ensure that [assignment: *set of users and/or subjects*] are unable to determine the
3738 real user name bound to [assignment: *list of subjects and/or operations and/or objects*].

3739 **FPR_ANO.2.2**

3740 The TSF **shall** provide [assignment: *list of services*] to [assignment: *list of subjects*]
3741 without soliciting any reference to the real user name.

3742 **13.3 Pseudonymity (FPR_PSE)**

3743 **13.3.1 Family behaviour**

3744 This family ensures that a user **may** use a resource or service without disclosing its user
3745 identity but **can** still be accountable for that use.

3746 **13.3.2 Components leveling and description**

3747 Figure 57 shows the component leveling for this family.



3748 **Figure 57 — FPR_PSE: Component leveling**

3749 FPR_PSE.1 Pseudonymity requires that a set of users and/or subjects are unable to determine
 3750 the identity of a user bound to a subject or operation, but that this user is still accountable for
 3751 its actions.

3752 FPR_PSE.2 Reversible pseudonymity, requires the TSF to provide a capability to determine the
 3753 original user identity based on a provided alias.

3754 FPR_PSE.3 Alias pseudonymity, requires the TSF to follow certain construction rules for the
 3755 alias to the user identity.

3756 **13.3.3 Management of FPR_PSE.1, FPR_PSE.2, FPR_PSE.3**

3757 The following actions **could** be considered for the management functions in FMT:

3758 a) There are no management activities foreseen.

3759 **13.3.4 Audit of FPR_PSE.1, FPR_PSE.2, FPR_PSE.3**

3760 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3761 in the PP, PP-Module, functional package or ST:

3762 a) Minimal: The subject/user that requested resolution of the user identity **should** be
 3763 audited.

3764 **13.3.5 FPR_PSE.1 Pseudonymity**

3765 **Component relationships**

3766 Hierarchical to: No other components.

3767 Dependencies: No dependencies.

3768 **FPR_PSE.1.1**

3769 **The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to**
 3770 **determine the real user name bound to [assignment: *list of subjects and/or operations***
 3771 ***and/or objects*].**

3772 **FPR_PSE.1.2**

3773 **The TSF shall be able to provide [assignment: *number of aliases*] aliases of the real user**
 3774 **name to [assignment: *list of subjects*].**

3775 **FPR_PSE.1.3**

3776 **The TSF shall [selection, choose one of: *determine an alias for a user, accept the alias from***
 3777 ***the user*] and verify that it conforms to the [assignment: *alias metric*].**

3778 **13.3.6 FPR_PSE.2 Reversible pseudonymity**

3779 **Component relationships**

3780 Hierarchical to: FPR_PSE.1 Pseudonymity

3781 Dependencies: FIA_UID.1 Timing of identification

3782 **FPR_PSE.2.1**

3783 The TSF **shall** ensure that [assignment: *set of users and/or subjects*] are unable to determine the
 3784 real user name bound to [assignment: *list of subjects and/or operations and/or objects*].

3785 **FPR_PSE.2.2**

3786 The TSF **shall** be able to provide [assignment: *number of aliases*] aliases of the real user name to
 3787 [assignment: *list of subjects*].

3788 **FPR_PSE.2.3**

3789 The TSF **shall** [selection, choose one of: *determine an alias for a user, accept the alias from the*
 3790 *user*] and verify that it conforms to the [assignment: *alias metric*].

3791 **FPR_PSE.2.4**

3792 **The TSF shall provide [selection: *an authorized user, [assignment: list of trusted subjects]]***
 3793 **a capability to determine the user identity based on the provided alias only under the**
 3794 **following [assignment: *list of conditions*].**

3795 **13.3.7 FPR_PSE.3 Alias pseudonymity**

3796 **Component relationships**

3797 Hierarchical to: FPR_PSE.1 Pseudonymity

3798 Dependencies: No dependencies.

3799 **FPR_PSE.3.1**

3800 The TSF **shall** ensure that [assignment: *set of users and/or subjects*] are unable to determine the
 3801 real user name bound to [assignment: *list of subjects and/or operations and/or objects*].

3802 **FPR_PSE.3.2**

3803 The TSF **shall** be able to provide [assignment: *number of aliases*] aliases of the real user name to
 3804 [assignment: *list of subjects*].

3805 **FPR_PSE.3.3**

3806 The TSF **shall** [selection, choose one of: *determine an alias for a user, accept the alias from the*
 3807 *user*] and verify that it conforms to the [assignment: *alias metric*].

3808 **FPR_PSE.3.4**

3809 **The TSF shall provide an alias to the real user name which shall be identical to an alias**
 3810 **provided previously under the following [assignment: *list of conditions*] otherwise the**
 3811 **alias provided shall be unrelated to previously provided aliases.**

3812 **13.4 Unlinkability (FPR_UNL)**

3813 **13.4.1 Family behaviour**

3814 This family ensures that selected entities **may** be linked together without external entities being
 3815 able to back trace these links.

3816 **13.4.2 Components leveling and description**

3817 Figure 58 shows the component leveling for this family.



3818

3819 **Figure 58 — FPR_UNL: Component leveling**

3820 **FPR_UNL.1 Unlinkability of operations** requires that users and/or subjects are unable to
 3821 determine whether the same user caused certain specific operations in the system, or whether
 3822 operations are related in some other manner. This component ensures that users cannot link
 3823 different operations in the system and thereby obtain information.

3824 13.4.3 Management of FPR_UNL.1

3825 The following actions **could** be considered for the management functions in FMT:

3826 a) The management of the unlinkability function.

3827 13.4.4 Audit of FPR_UNL.1

3828 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3829 in the PP, PP-Module, functional package or ST:

3830 a) Minimal: The invocation of the unlinkability mechanism.

3831 13.4.5 FPR_UNL.1 Unlinkability of operations

3832 Component relationships

3833 Hierarchical to: No other components.

3834 Dependencies: No dependencies.

3835 FPR_UNL.1.1

3836 The TSF **shall** ensure that [assignment: *set of entities and/or operations*] are unable to
 3837 determine whether [assignment: *list of entities and/or operations*] [selection: *were*
 3838 *caused by the same user, are related as follows [assignment: *list of relations*]].*

3839 NOTE This SFR does not only stipulate at the individual set of operations performed by one entity. This SFR
 3840 intends to look at a chain of interlinked operations by multiple entities. This chain can be subsumed as a transaction.

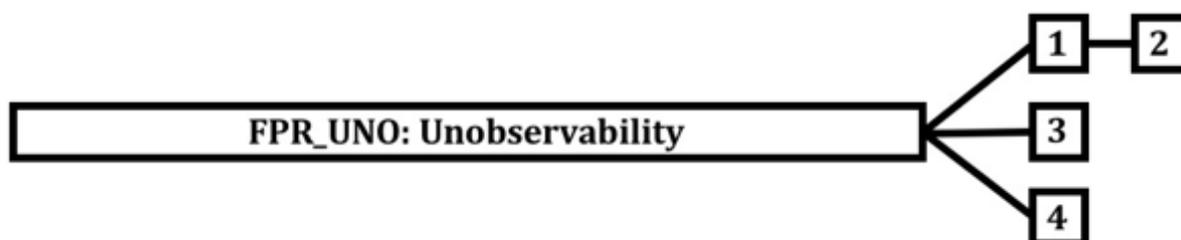
3841 13.5 Unobservability (FPR_UNO)

3842 13.5.1 Family behaviour

3843 This family ensures that a user **may** use a resource or service without others, especially third
 3844 parties, being able to observe that the resource or service is being used.

3845 13.5.2 Components leveling and description

3846 Figure 59 shows the component leveling for this family.



3847 **Figure 59 — FPR_UNO: Component leveling**

3848 FPR_UNO.1 Unobservability, requires that users and/or subjects cannot determine whether an
 3849 operation is being performed.

3850 FPR_UNO.2 Allocation of information impacting unobservability, requires that the TSF provide
 3851 specific mechanisms to avoid the concentration of privacy related information within the TOE.
 3852 Such concentrations might impact unobservability if a security compromise occurs.

3853 FPR_UNO.3 Unobservability without soliciting information, requires that the TSF does not try to
3854 obtain privacy related information that might be used to compromise unobservability.

3855 FPR_UNO.4 Authorized user observability, requires the TSF to provide one or more authorized
3856 users with a capability to observe the usage of resources and/or services.

3857 **13.5.3 Management of FPR_UNO.1, FPR_UNO.2**

3858 The following actions **could** be considered for the management functions in FMT:

3859 a) The management of the behaviour of the unobservability function.

3860 **13.5.4 Management of FPR_UNO.3**

3861 The following actions **could** be considered for the management functions in FMT:

3862 a) There are no management activities foreseen.

3863 **13.5.5 Management of FPR_UNO.4**

3864 The following actions **could** be considered for the management functions in FMT:

3865 a) The list of authorized users that are capable of determining the occurrence of
3866 operations.

3867 **13.5.6 Audit of FPR_UNO.1, FPR_UNO.2**

3868 The following actions should be auditable if FAU_GEN Security audit data generation is included
3869 in the PP, PP-Module, functional package or ST:

3870 a) Minimal: The invocation of the unobservability mechanism.

3871 **13.5.7 Audit of FPR_UNO.3**

3872 The following actions should be auditable if FAU_GEN Security audit data generation is included
3873 in the PP, PP-Module, functional package or ST:

3874 a) There are no auditable events foreseen.

3875 **13.5.8 Audit of FPR_UNO.4**

3876 The following actions should be auditable if FAU_GEN Security audit data generation is included
3877 in the PP, PP-Module, functional package or ST:

3878 a) Minimal: The observation of the use of a resource or service by a user or subject.

3879 **13.5.9 FPR_UNO.1 Unobservability**

3880 **Component relationships**

3881 Hierarchical to: No other components.

3882 Dependencies: No dependencies.

3883 **FPR_UNO.1.1**

3884 **The TSF shall ensure that [assignment: *list of users and/or subjects*] are unable to**
3885 **observe the operation [assignment: *list of operations*] on [assignment: *list of objects*] by**
3886 **[assignment: *list of protected users and/or subjects*].**

3887 **13.5.10 FPR_UNO.2 Allocation of information impacting unobservability**

3888 **Component relationships**

3889 Hierarchical to: FPR_UNO.1 Unobservability

3890	Dependencies:	No dependencies.
3891	FPR_UNO.2.1	
3892	The TSF shall ensure that [assignment: <i>list of users and/or subjects</i>] are unable to observe the	
3893	operation [assignment: <i>list of operations</i>] on [assignment: <i>list of objects</i>] by [assignment: <i>list of</i>	
3894	<i>protected users and/or subjects</i>].	
3895	FPR_UNO.2.2	
3896	The TSF shall allocate the [assignment: <i>unobservability related information</i>] among	
3897	different parts of the TOE such that the following conditions hold during the lifetime of	
3898	the information: [assignment: <i>list of conditions</i>].	
3899	13.5.11 FPR_UNO.3 Unobservability without soliciting information	
3900	Component relationships	
3901	Hierarchical to:	No other components.
3902	Dependencies:	FPR_UNO.1 Unobservability
3903	FPR_UNO.3.1	
3904	The TSF shall provide [assignment: <i>list of services</i>] to [assignment: <i>list of subjects</i>]	
3905	without soliciting any reference to [assignment: <i>privacy related information</i>].	
3906	13.5.12 FPR_UNO.4 Authorized user observability	
3907	Component relationships	
3908	Hierarchical to:	No other components.
3909	Dependencies:	No dependencies.
3910	FPR_UNO.4.1	
3911	The TSF shall provide [assignment: <i>set of authorized users</i>] with the capability to observe	
3912	the usage of [assignment: <i>list of resources and/or services</i>].	
3913		

3914 **14 Class FPT: Protection of the TSF**

3915 **14.1 Class description**

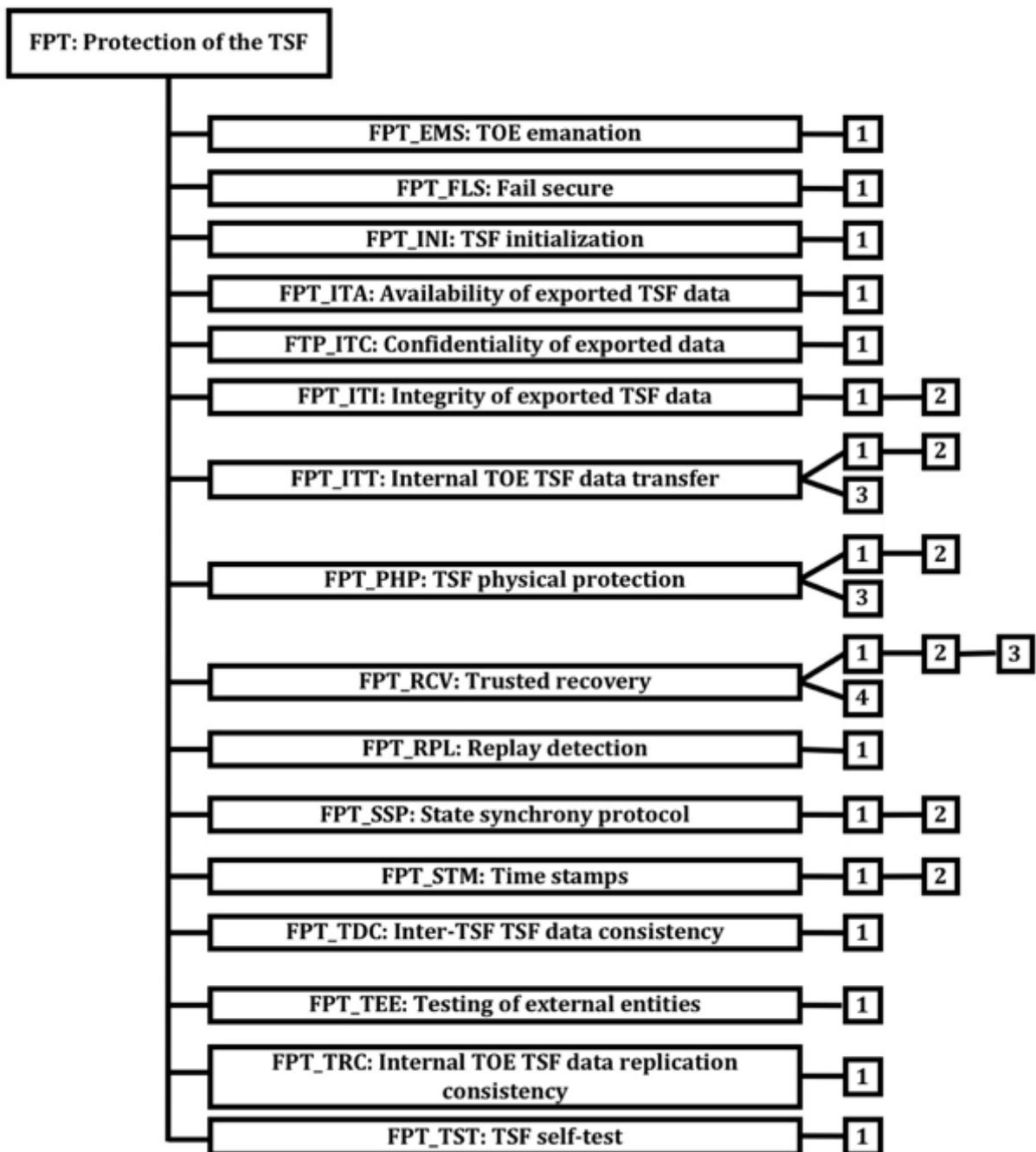
3916 This class contains families of functional requirements that relate to the integrity and
3917 management of the mechanisms that constitute the TSF and to the integrity of TSF data. In some
3918 sense, families in this class **may** appear to duplicate components in the FDP: User data
3919 protection class; they **may** even be implemented using the same mechanisms. However, FDP:
3920 User data protection focuses on user data protection, while FPT: Protection of the TSF focuses
3921 on TSF data protection. In fact, Components from the FPT: Protection of the TSF class are
3922 necessary to provide requirements that the SFPs in the TOE cannot be tampered with or
3923 bypassed.

3924 From the point of view of this class, regarding to the TSF there are three significant elements:

- 3925 a) The TSF's implementation, which executes and implements the mechanisms that
3926 enforce the SFRs.
- 3927 b) The TSF's data, which are the administrative databases that guide the enforcement
3928 of the SFRs.
- 3929 c) The external entities that the TSF **may** interact with in order to enforce the SFRs.

3930 Figure 60 shows the decomposition of this class, it's families and components. Elements are not
3931 shown in the figure.

3932 Annex J provides explanatory information for this class and **should** be consulted when using the
 3933 components identified in this class.



3934 **Figure 60 — FPT: Protection of the TSF class decomposition**

3935 14.2 TOE emanation (FPT_EMS)

3936 14.2.1 Family behaviour

3937 The family FPT_EMS (TOE Emanation) of the Class FPT (Protection of the TSF) is defined here
 3938 to describe the IT security functional requirements of the TOE related to leakage of information
 3939 based on emanation.

3940 The TOE **shall** prevent attacks against the TOE and secret data processed by the TOE where the
 3941 attack is based on external observable phenomena of the TOE during its operation. Hereby,

3942 different types of emissions and interfaces of the TOE as well as different types of TSF data and
 3943 user data may be addressed.

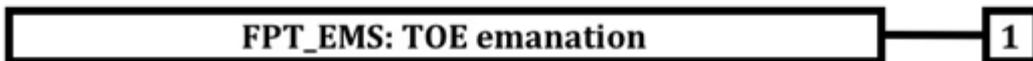
3944 EXAMPLE

3945 Examples of such attacks against the TOE and its processed secret data are simple power analysis (SPA), differential
 3946 power analysis (DPA), simple electromagnetic analysis (SEMA), differential electromagnetic analysis (DEMA), timing
 3947 attacks, padding oracle attacks, cache miss attacks, etc.

3948 This family describes the functional requirements for the limitation of intelligible emanations
 3949 which are not directly addressed by any other component of ISO/IEC 15408-2.

3950 **14.2.2 Components leveling and description**

3951 Figure 61 shows the component leveling for this family.



3952 **Figure 61 — FPT_EMS: Component leveling**

3953 This family consists of only one component, FPT_EMS.1 Emanation of TSF and User data, which
 3954 defines requirements for the TOE to mitigate intelligible emanations.

3955 **14.2.3 Management of FPT_EMS.1**

3956 The following actions could be considered for the management functions in FMT:

- 3957 a) There are no management activities foreseen.

3958 **14.2.4 Audit of FPT_EMS.1**

3959 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3960 in the PP, PP-Module, functional package or ST:

- 3961 a) There are no auditable events foreseen.

3962 **14.2.5 FPT_EMS.1 Emanation of TSF and User data**

3963 **Component relationships**

3964 Hierarchical to: No other components.

3965 Dependencies: No dependencies.

3966 **FPT_EMS.1.1**

3967 The TSF shall ensure that the TOE does not emit emissions over its attack surface in such
 3968 amount that these emissions enable access to TSF data and user data as specified in the
 3969 following table:

3970 **FPT_EMS.1.1 Table**

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

3971

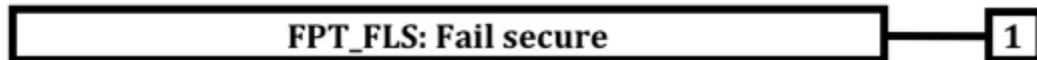
3972 14.3 Fail secure (FPT_FLS)

3973 14.3.1 Family behaviour

3974 The requirements of this family ensure that the TOE will always enforce its SFRs in the event of
3975 identified categories of failures in the TSF.

3976 14.3.2 Components leveling and description

3977 Figure 62 shows the component leveling for this family.



3978 **Figure 62 — FPT_FLS: Component leveling**

3979 This family consists of only one component, FPT_FLS.1 Failure with preservation of secure
3980 state, which requires that the TSF preserve a secure state in the face of the identified failures.

3981 14.3.3 Management of FPT_FLS.1

3982 The following actions **could** be considered for the management functions in FMT:

3983 a) There are no management activities foreseen.

3984 14.3.4 Audit of FPT_FLS.1

3985 The following actions should be auditable if FAU_GEN Security audit data generation is included
3986 in the PP, PP-Module, functional package or /ST:

3987 a) Basic: Failure of the TSF.

3988 14.3.5 FPT_FLS.1 Failure with preservation of secure state

3989 Component relationships

3990 Hierarchical to: No other components.

3991 Dependencies: No dependencies.

3992 FPT_FLS.1.1

3993 **The TSF shall preserve a secure state when the following types of failures occur:**
3994 **[assignment: list of types of failures in the TSF].**

3995 14.4 TSF initialization (FPT_INI)

3996 14.4.1 Family behaviour

3997 This family describes the functional requirements for the initialization of the TSF by a dedicated
3998 function of the TOE that ensures the initialization in a correct and secure operational state.

3999 14.4.2 Components leveling and description

4000 Figure 63 shows the component leveling for this family.



4001 **Figure 63 — FPT_INI: Component leveling**

4002 This family consists of only one component, Component FPT_INI.1. This component requires the
 4003 TOE to provide a TSF initialization function that brings the TSF into a secure operational state
 4004 at power-on.

4005 **14.4.3 Management of FPT_INI.1**

4006 The following actions **could** be considered for the management functions in FMT:

- 4007 a) There are no management activities foreseen.

4008 **14.4.4 Audit of FPT_INI.1**

4009 The following actions should be auditable if FAU_GEN Security audit data generation is included
 4010 in the PP, PP-Module, functional package or /ST:

- 4011 a) There are no auditable events foreseen.

4012 **14.4.5 FPT_INI.1 TSF initialization**

4013 **Component relationships**

4014 Hierarchical to: No other components.

4015 Dependencies: No dependencies.

4016 **FPT_INI.1.1**

4017 **The TOE shall provide an initialization function which is self-protected for integrity and**
 4018 **authenticity.**

4019 **FPT_INI.1.2**

4020 **The TOE initialization function shall ensure that certain properties hold on certain**
 4021 **elements immediately before establishing the TSF in a secure initial state, as specified**
 4022 **below:**

4023 **FPT_INI.1.2 Table**

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

4024 **FPT_INI.1.3**

4025 **The TOE initialization function shall detect and respond to errors and failures during**
 4026 **initialization such that the TOE [selection: *is halted, successfully completes initialization***
 4027 **with [selection: *reduced functionality, signaling error state, [assignment: list of actions]*].**

4028 **FPT_INI.1.4**

4029 **The TOE initialization function shall only interact with the TSF in [assignment: *defined***
 4030 ***methods*] during initialization.**

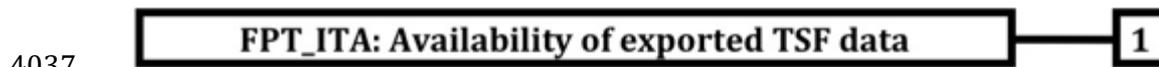
4031 **14.5 Availability of exported TSF data (FPT_ITA)**

4032 **14.5.1 Family behaviour**

4033 This family defines the rules for the prevention of loss of availability of TSF data moving
 4034 between the TSF and another trusted IT product.

4035 14.5.2 Components leveling and description

4036 Figure 64 shows the component leveling for this family.



4038 **Figure 64 — FPT_ITA: Component leveling**

4039 This family consists of only one component, FPT_ITA.1 Inter-TSF availability within a defined
4040 availability metric. This component requires that the TSF ensure, to an identified degree of
4041 probability, the availability of TSF data provided to another trusted IT product.

4042 14.5.3 Management of FPT_ITA.1

4043 The following actions **could** be considered for the management functions in FMT:

- 4044 a) management of the list of types of TSF data that must be available to another
4045 trusted IT product.

4046 14.5.4 Audit of FPT_ITA.1

4047 The following actions should be auditable if FAU_GEN Security audit data generation is included
4048 in the PP, PP-Module, functional package or ST:

- 4049 a) Minimal: the absence of TSF data when required by a TOE.

4050 14.5.5 FPT_ITA.1 Inter-TSF availability within a defined availability metric

4051 Component relationships

4052 Hierarchical to: No other components.

4053 Dependencies: No dependencies.

4054 FPT_ITA.1.1

4055 **The TSF shall ensure the availability of [assignment: list of types of TSF data] provided to**
4056 **another trusted IT product within [assignment: a defined availability metric] given the**
4057 **following conditions [assignment: conditions to ensure availability].**

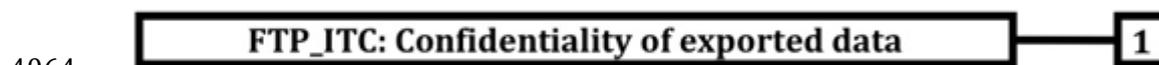
4058 14.6 Confidentiality of exported TSF data (FPT_ITC)

4059 14.6.1 Family behaviour

4060 This family defines the rules for the protection from unauthorized disclosure of TSF data during
4061 transmission between the TSF and another trusted IT product.

4062 14.6.2 Components leveling and description

4063 Figure 65 shows the component leveling for this family.



4065 **Figure 65 — FPT_ITC: Component leveling**

4066 This family consists of only one component, FPT_ITC.1 Inter-TSF confidentiality during
4067 transmission, which requires that the TSF ensure that data transmitted between the TSF and
4068 another trusted IT product is protected from disclosure while in transit.

4069 **14.6.3 Management of FPT_ITC.1**

4070 The following actions **could** be considered for the management functions in FMT:

- 4071 a) There are no management activities foreseen.

4072 **14.6.4 Audit of FPT_ITC.1**

4073 The following actions should be auditable if FAU_GEN Security audit data generation is included
4074 in the PP, PP-Module, functional package or ST:

- 4075 a) There are no auditable events foreseen.

4076 **14.6.5 FPT_ITC.1 Inter-TSF confidentiality during transmission**

4077 **Component relationships**

4078 Hierarchical to: No other components.

4079 Dependencies: No dependencies.

4080 **FPT_ITC.1.1**

4081 **The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product**
4082 **from unauthorized disclosure during transmission.**

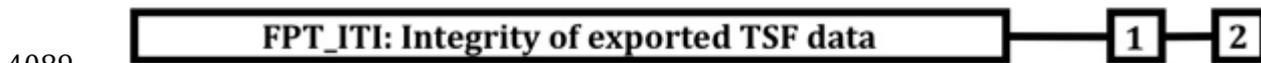
4083 **14.7 Integrity of exported TSF data (FPT_ITI)**

4084 **14.7.1 Family behaviour**

4085 This family defines the rules for the protection, from unauthorized modification, of TSF data
4086 during transmission between the TSF and another trusted IT product.

4087 **14.7.2 Components leveling and description**

4088 Figure 66 shows the component leveling for this family.



4090 **Figure 66 — FPT_ITI: Component leveling**

4091 FPT_ITI.1 Inter-TSF detection of modification, provides the ability to detect modification of TSF
4092 data during transmission between the TSF and another trusted IT product, under the
4093 assumption that another trusted IT product is cognizant of the mechanism used.

4094 FPT_ITI.2 Inter-TSF detection and correction of modification, provides the ability for another
4095 trusted IT product not only to detect modification, but to correct modified TSF data under the
4096 assumption that another trusted IT product is cognizant of the mechanism used.

4097 **14.7.3 Management of FPT_ITI.1**

4098 The following actions **could** be considered for the management functions in FMT:

- 4099 a) There are no management activities foreseen.

4100 **14.7.4 Management of FPT_ITI.2**

4101 The following actions **could** be considered for the management functions in FMT:

- 4102 a) Management of the types of TSF data that the TSF tries to correct if modified in
4103 transit;

4104 b) Management of the types of action that the TSF takes if TSF data is modified in
4105 transit.

4106 **14.7.5 Audit of FPT_ITI.1**

4107 The following actions should be auditable if FAU_GEN Security audit data generation is included
4108 in the PP, PP-Module, functional package or ST:

4109 a) Minimal: the detection of modification of transmitted TSF data.

4110 b) Basic: the action taken upon detection of modification of transmitted TSF data.

4111 **14.7.6 Audit of FPT_ITI.2**

4112 The following actions should be auditable if FAU_GEN Security audit data generation is included
4113 in the PP, PP-Module, functional package or ST:

4114 a) Minimal: the detection of modification of transmitted TSF data.

4115 b) Basic: the action taken upon detection of modification of transmitted TSF data.

4116 c) Basic: the use of the correction mechanism.

4117 **14.7.7 FPT_ITI.1 Inter-TSF detection of modification**

4118 **Component relationships**

4119 Hierarchical to: No other components.

4120 Dependencies: No dependencies.

4121 **FPT_ITI.1.1**

4122 **The TSF shall provide the capability to detect modification of all TSF data during**
4123 **transmission between the TSF and another trusted IT product within the following**
4124 **metric: [assignment: *a defined modification metric*].**

4125 **FPT_ITI.1.2**

4126 **The TSF shall provide the capability to verify the integrity of all TSF data transmitted**
4127 **between the TSF and another trusted IT product and perform [assignment: *action to be***
4128 ***taken*] if modifications are detected.**

4129 **14.7.8 FPT_ITI.2 Inter-TSF detection and correction of modification**

4130 **Component relationships**

4131 Hierarchical to: FPT_ITI.1 Inter-TSF detection of modification

4132 Dependencies: No dependencies.

4133 **FPT_ITI.2.1**

4134 The TSF shall provide the capability to detect modification of all TSF data during transmission
4135 between the TSF and another trusted IT product within the following metric: [assignment: *a*
4136 *defined modification metric*].

4137 **FPT_ITI.2.2**

4138 The TSF shall provide the capability to verify the integrity of all TSF data transmitted between
4139 the TSF and another trusted IT product and perform [assignment: *action to be taken*] if
4140 modifications are detected.

4141 **FPT_ITI.2.3**

4142 **The TSF shall provide the capability to correct [assignment: *type of modification*] of all**
 4143 **TSF data transmitted between the TSF and another trusted IT product.**

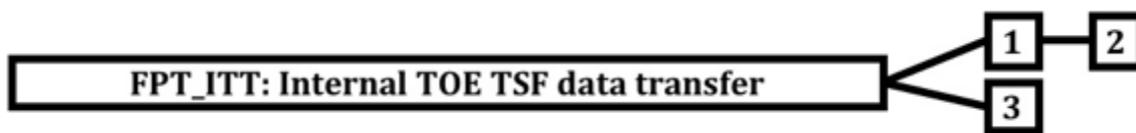
4144 **14.8 Internal TOE TSF data transfer (FPT_ITT)**

4145 **14.8.1 Family behaviour**

4146 This family provides requirements that address protection of TSF data when it is transferred
 4147 between separate parts of a TOE across an internal channel.

4148 **14.8.2 Components leveling and description**

4149 Figure 67 shows the component leveling for this family.



4150

4151 **Figure 67 — FPT_ITT: Component leveling**

4152 FPT_ITT.1 Basic internal TSF data transfer protection, requires that TSF data be protected when
 4153 transmitted between separate parts of the TOE.

4154 FPT_ITT.2 TSF data transfer separation, requires that the TSF separate user data from TSF data
 4155 during transmission.

4156 FPT_ITT.3 TSF data integrity monitoring, requires that the TSF data transmitted between
 4157 separate parts of the TOE is monitored for identified integrity errors.

4158 **14.8.3 Management of FPT_ITT.1**

4159 The following actions could be considered for the management functions in FMT:

- 4160 a) management of the types of modification against which the TSF should protect;
- 4161 b) management of the mechanism used to provide the protection of the data in transit
- 4162 between different parts of the TSF.

4163 **14.8.4 Management of FPT_ITT.2**

4164 The following actions could be considered for the management functions in FMT:

- 4165 a) management of the types of modification against which the TSF should protect;
- 4166 b) management of the mechanism used to provide the protection of the data in transit
- 4167 between different parts of the TSF;
- 4168 c) management of the separation mechanism.

4169 **14.8.5 Management of FPT_ITT.3**

4170 The following actions could be considered for the management functions in FMT:

- 4171 a) management of the types of modification against which the TSF should protect;
- 4172 b) management of the mechanism used to provide the protection of the data in transit
- 4173 between different parts of the TSF;
- 4174 c) management of the types of modification of TSF data the TSF should try to detect;
- 4175 d) management of the actions that will be taken.

4176 **14.8.6 Audit of FPT_ITT.1, FPT_ITT.2**

4177 The following actions should be auditable if FAU_GEN Security audit data generation is included
4178 in the PP, PP-Module, functional package or ST:

4179 a) There are no auditable events foreseen.

4180 **14.8.7 Audit of FPT_ITT.3**

4181 The following actions should be auditable if FAU_GEN Security audit data generation is included
4182 in the PP, PP-Module, functional package or ST:

4183 a) Minimal: the detection of modification of TSF data;

4184 b) Basic: the action taken following detection of an integrity error.

4185 **14.8.8 FPT_ITT.1 Basic internal TSF data transfer protection**4186 **Component relationships**

4187 Hierarchical to: No other components.

4188 Dependencies: No dependencies.

4189 **FPT_ITT.1.1**

4190 **The TSF shall protect TSF data from [selection: *disclosure, modification*] when it is**
4191 **transmitted between separate parts of the TOE.**

4192 **14.8.9 FPT_ITT.2 TSF data transfer separation**4193 **Component relationships**

4194 Hierarchical to: FPT_ITT.1 Basic internal TSF data transfer
4195 protection

4196 Dependencies: No dependencies.

4197 **FPT_ITT.2.1**

4198 The TSF shall protect TSF data from [selection: *disclosure, modification*] when it is transmitted
4199 between separate parts of the TOE.

4200 **FPT_ITT.2.2**

4201 **The TSF shall separate user data from TSF data when such data is transmitted between**
4202 **separate parts of the TOE.**

4203 **14.8.10 FPT_ITT.3 TSF data integrity monitoring**4204 **Component relationships**

4205 Hierarchical to: No other components.

4206 Dependencies: FPT_ITT.1 Basic internal TSF data transfer
4207 protection

4208 **FPT_ITT.3.1**

4209 **The TSF shall be able to detect [selection: *modification of data, substitution of data, re-***
4210 **ordering of data, deletion of data, [assignment: *other integrity errors*]] for TSF data**
4211 **transmitted between separate parts of the TOE.**

4212 **FPT_ITT.3.2**

4213 **Upon detection of a data integrity error, the TSF shall take the following actions:**
 4214 **[assignment: specify the action to be taken].**

4215 **14.9 TSF physical protection (FPT_PHP)**

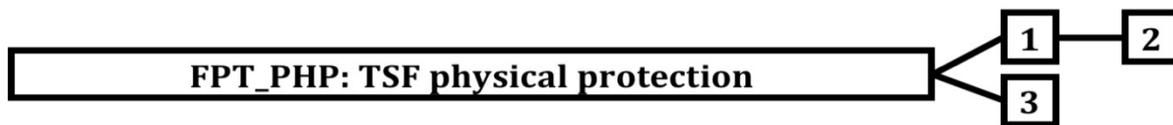
4216 **14.9.1 Family behaviour**

4217 TSF physical protection components refer to restrictions on unauthorized physical access to the
 4218 TSF, and to the deterrence of, and resistance to, unauthorized physical modification, or
 4219 substitution of the TSF.

4220 The requirements of components in this family ensure that the TSF is protected from physical
 4221 tampering and interference. Satisfying the requirements of these components results in the TSF
 4222 being packaged and used in such a manner that physical tampering is detectable, or resistance
 4223 to physical tampering is enforced. Without these components, the protection functions of a TSF
 4224 lose their effectiveness in environments where physical damage cannot be prevented. This
 4225 family also provides requirements regarding how the TSF shall respond to physical tampering
 4226 attempts.

4227 **14.9.2 Components leveling and description**

4228 Figure 68 shows the component leveling for this family.



4229

4230 **Figure 68 — FPT_PHP: Component leveling**

4231 FPT_PHP.1 Passive detection of physical attack, provides for features that indicate when a TSF
 4232 device or TSF element is subject to tampering. However, notification of tampering is not
 4233 automatic; an authorized user must invoke a security administrative function or perform
 4234 manual inspection to determining if tampering has occurred.

4235 FPT_PHP.2 Notification of physical attack, provides for automatic notification of tampering for
 4236 an identified subset of physical penetrations.

4237 FPT_PHP.3 Resistance to physical attack, provides for features that prevent or resist physical
 4238 tampering with TSF devices and TSF elements.

4239 **14.9.3 Management of FPT_PHP.1**

4240 The following actions could be considered for the management functions in FMT:

- 4241 a) Management of the user or role that determines whether physical tampering has
- 4242 occurred.

4243 **14.9.4 Management of FPT_PHP.2**

4244 The following actions could be considered for the management functions in FMT:

- 4245 a) Management of the user or role that gets informed about intrusions;
- 4246 b) Management of the list of devices that should inform the indicated user or role
- 4247 about the intrusion.

4248 **14.9.5 Management of FPT_PHP.3**

4249 The following actions could be considered for the management functions in FMT:

4250 a) Management of the automatic responses to physical tampering.

4251 **14.9.6 Audit of FPT_PHP.1**

4252 The following actions should be auditable if FAU_GEN Security audit data generation is included
4253 in the PP, PP-Module, functional package or ST:

4254 a) Minimal: if detection by IT means, detection of intrusion.

4255 **14.9.7 Audit of FPT_PHP.2**

4256 The following actions should be auditable if FAU_GEN Security audit data generation is included
4257 in the PP, PP-Module, functional package or ST:

4258 a) Minimal: detection of intrusion.

4259 **14.9.8 Audit of FPT_PHP.3**

4260 The following actions should be auditable if FAU_GEN Security audit data generation is included
4261 in the PP, PP-Module, functional package or ST:

4262 a) There are no auditable events foreseen.

4263 **14.9.9 FPT_PHP.1 Passive detection of physical attack**

4264 **Component relationships**

4265 Hierarchical to: No other components.

4266 Dependencies: No dependencies.

4267 **FPT_PHP.1.1**

4268 **The TSF shall provide unambiguous detection of physical tampering that might**
4269 **compromise the TSF.**

4270 **FPT_PHP.1.2**

4271 **The TSF shall provide the capability to determine whether physical tampering with the**
4272 **TSF's devices or TSF's elements has occurred.**

4273 **14.9.10 FPT_PHP.2 Notification of physical attack**

4274 **Component relationships**

4275 Hierarchical to: FPT_PHP.1 Passive detection of physical attack

4276 Dependencies: FMT_LIM.1 Limited capabilities

4277 **FPT_PHP.2.1**

4278 The TSF shall provide unambiguous detection of physical tampering that might compromise the
4279 TSF.

4280 **FPT_PHP.2.2**

4281 The TSF shall provide the capability to determine whether physical tampering with the TSF's
4282 devices or TSF's elements has occurred.

4283 **FPT_PHP.2.3**

4284 **For [assignment: list of TSF devices/elements for which active detection is required], the**
4285 **TSF shall monitor the devices and elements and notify [assignment: a designated user or**
4286 **role] when physical tampering with the TSF's devices or TSF's elements has occurred.**

4287 **14.9.11 FPT_PHP.3 Resistance to physical attack**

4288 **Component relationships**

4289 Hierarchical to: No other components.

4290 Dependencies: No dependencies.

4291 **FPT_PHP.3.1**

4292 **The TSF shall resist [assignment: *physical tampering scenarios*] to the [assignment: *list of***
 4293 ***TSF devices/elements*] by responding automatically such that the SFRs are always**
 4294 **enforced.**

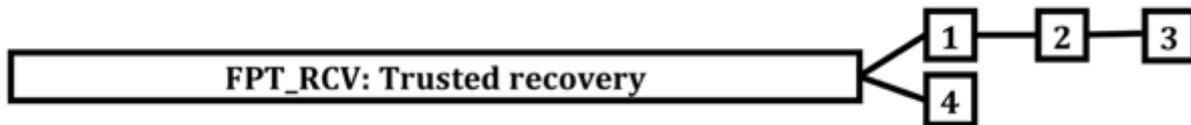
4295 **14.10 Trusted recovery (FPT_RCV)**

4296 **14.10.1 Family behaviour**

4297 The requirements of this family ensure that the TSF can determine that the TOE is started up
 4298 without protection compromise and can recover without protection compromise after
 4299 discontinuity of operations. This family is important because the start-up state of the TSF
 4300 determines the protection of subsequent states.

4301 **14.10.2 Components leveling and description**

4302 Figure 69 shows the component leveling for this family.



4303

4304 **Figure 69 — FPT_RCV: Component leveling**

4305 FPT_RCV.1 Manual recovery, allows a TOE to only provide mechanisms that involve human
 4306 intervention to return to a secure state.

4307 FPT_RCV.2 Automated recovery, provides, for at least one type of service discontinuity,
 4308 recovery to a secure state without human intervention; recovery for other discontinuities that
 4309 can require human intervention.

4310 FPT_RCV.3 Automated recovery without undue loss, also provides for automated recovery, but
 4311 strengthens the requirements by disallowing undue loss of protected objects.

4312 FPT_RCV.4 Function recovery, provides for recovery at the level of particular functions,
 4313 ensuring either successful completion or rollback of TSF data to a secure state.

4314 **14.10.3 Management of FPT_RCV.1**

4315 The following actions could be considered for the management functions in FMT:

- 4316 a) Management of who can access the restore capability within the maintenance
- 4317 mode.

4318 **14.10.4 Management of FPT_RCV.2, FPT_RCV.3**

4319 The following actions could be considered for the management functions in FMT:

- 4320 a) Management of who can access the restore capability within the maintenance
- 4321 mode;
- 4322 b) Management of the list of failures/service discontinuities that will be handled
- 4323 through the automatic procedures.

4324 **14.10.5 Management of FPT_RCV.4**4325 The following actions **could** be considered for the management functions in FMT:

4326 a) There are no management activities foreseen.

4327 **14.10.6 Audit of FPT_RCV.1, FPT_RCV.2, FPT_RCV.3**4328 The following actions should be auditable if FAU_GEN Security audit data generation is included
4329 in the PP, PP-Module, functional package or ST:

4330 a) Minimal: the fact that a failure or service discontinuity occurred;

4331 b) Minimal: resumption of the regular operation;

4332 c) Basic: type of failure or service discontinuity.

4333 **14.10.7 Audit of FPT_RCV.4**4334 The following actions should be auditable if FAU_GEN Security audit data generation is included
4335 in the PP, PP-Module, functional package or ST:4336 a) Minimal: if possible, the impossibility to return to a secure state after a failure of
4337 the TSF;

4338 b) Basic: if possible, the detection of a failure of a function.

4339 **14.10.8 FPT_RCV.1 Manual recovery**4340 **Component relationships**

4341 Hierarchical to: No other components.

4342 Dependencies: AGD_OPE.1 Operational user guidance

4343 **FPT_RCV.1.1**4344 **After [assignment: *list of failures/service discontinuities*] the TSF **shall** enter a**
4345 **maintenance mode where the ability to return to a secure state is provided.**4346 **14.10.9 FPT_RCV.2 Automated recovery**4347 **Component relationships**

4348 Hierarchical to: FPT_RCV.1 Manual recovery

4349 Dependencies: AGD_OPE.1 Operational user guidance

4350 **FPT_RCV.2.1**4351 **When automated recovery from [assignment: *list of failures/service discontinuities*] is not**
4352 **possible, the TSF **shall** enter a maintenance mode where the ability to return to a secure state is**
4353 **provided.**4354 **FPT_RCV.2.2**4355 **For [assignment: *list of failures/service discontinuities*], the TSF **shall** ensure the return of**
4356 **the TOE to a secure state using automated procedures.**4357 **14.10.10 FPT_RCV.3 Automated recovery without undue loss**4358 **Component relationships**

4359 Hierarchical to: FPT_RCV.2 Automated recovery

4360 Dependencies: AGD_OPE.1 Operational user guidance

4361 **FPT_RCV.3.1**

4362 When automated recovery from [assignment: *list of failures/service discontinuities*] is not
 4363 possible, the TSF **shall** enter a maintenance mode where the ability to return to a secure state is
 4364 provided.

4365 **FPT_RCV.3.2**

4366 For [assignment: *list of failures/service discontinuities*], the TSF **shall** ensure the return of the
 4367 TOE to a secure state using automated procedures.

4368 **FPT_RCV.3.3**

4369 **The functions provided by the TSF to recover from failure or service discontinuity shall**
 4370 **ensure that the secure initial state is restored without exceeding [assignment:**
 4371 ***quantification*] for loss of TSF data or objects under the control of the TSF.**

4372 **FPT_RCV.3.4**

4373 **The TSF shall provide the capability to determine the objects that were or were not**
 4374 **capable of being recovered.**

4375 **14.10.11 FPT_RCV.4 Function recovery**

4376 **Component relationships**

4377 Hierarchical to: No other components.

4378 Dependencies: No dependencies.

4379 **FPT_RCV.4.1**

4380 **The TSF shall ensure that [assignment: *list of functions and failure scenarios*] have the**
 4381 **property that the function either completes successfully, or for the indicated failure**
 4382 **scenarios, recovers to a consistent and secure state.**

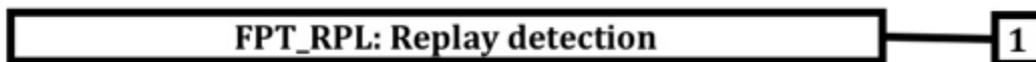
4383 **14.11 Replay detection (FPT_RPL)**

4384 **14.11.1 Family behaviour**

4385 This family addresses detection of replay for various types of entities and subsequent actions to
 4386 correct. In the case where replay **may** be detected, this effectively prevents it.

4387 **14.11.2 Components leveling and description**

4388 Figure 70 shows the component leveling for this family.



4389

4390 **Figure 70 — FPT_RPL: Component leveling**

4391 The family consists of only one component, FPT_RPL.1 Replay detection, which requires that
 4392 the TSF **shall** be able to detect the replay of identified entities.

4393 **14.11.3 Management of FPT_RPL.1**

4394 The following actions **could** be considered for the management functions in FMT:

- 4395 a) Management of the list of identified entities for which replay is detected;
- 4396 b) Management of the list of actions that need to be taken in case of replay.

4397 **14.11.4 Audit of FPT_RPL.1**

4398 The following actions should be auditable if FAU_GEN Security audit data generation is included
4399 in the PP, PP-Module, functional package or ST:

4400 a) Basic: Detected replay attacks.

4401 b) Detailed: Action to be taken based on the specific actions.

4402 **14.11.5 FPT_RPL.1 Replay detection**4403 **Component relationships**

4404 Hierarchical to: No other components.

4405 Dependencies: No dependencies.

4406 **FPT_RPL.1.1**

4407 The TSF **shall** detect replay for the following entities: [assignment: *list of identified*
4408 *entities*].

4409 **FPT_RPL.1.2**

4410 The TSF **shall** perform [assignment: *list of specific actions*] when replay is detected.

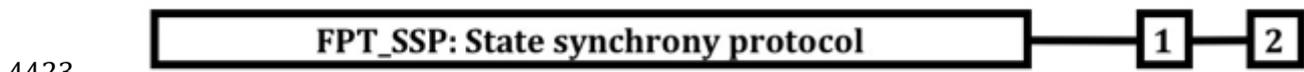
4411 **14.12 State synchrony protocol (FPT_SSP)**4412 **14.12.1 Family behaviour**

4413 Distributed TOEs **can** give rise to greater complexity than monolithic TOEs through the
4414 potential for differences in state between parts of the TOE, and through delays in
4415 communication. In most cases synchronization of state between distributed functions involves
4416 an exchange protocol, not a simple action. When malice exists in the distributed environment of
4417 these protocols, more complex defensive protocols are required.

4418 State synchrony protocol (FPT_SSP) establishes the requirement for certain critical functions of
4419 the TSF to use this trusted protocol. State synchrony protocol (FPT_SSP) ensures that two
4420 distributed parts of the TOE have synchronized their states after a security-relevant action.

4421 **14.12.2 Components leveling and description**

4422 Figure 71 shows the component leveling for this family.



4424 **Figure 71 — FPT_SSP: Component leveling**

4425 FPT_SSP.1 Simple trusted acknowledgement, requires only a simple acknowledgment by the
4426 data recipient.

4427 FPT_SSP.2 Mutual trusted acknowledgement, requires mutual acknowledgment of the data
4428 exchange.

4429 **14.12.3 Management of FPT_SSP.1, FPT_SSP.2**

4430 The following actions **could** be considered for the management functions in FMT:

4431 a) There are no management activities foreseen.

4432 **14.12.4 Audit of FPT_SSP.1, FPT_SSP.2**

4433 The following actions should be auditable if FAU_GEN Security audit data generation is included
4434 in the PP, PP-Module, functional package or ST:

- 4435 a) Minimal: failure to receive an acknowledgement when expected.

4436 **14.12.5 FPT_SSP.1 Simple trusted acknowledgement**

4437 **Component relationships**

4438 Hierarchical to: No other components.

4439 Dependencies: FPT_ITT.1 Basic internal TSF data transfer
4440 protection

4441 **FPT_SSP.1.1**

4442 **The TSF shall acknowledge, when requested by another part of the TSF, the receipt of an**
4443 **unmodified TSF data transmission.**

4444 **14.12.6 FPT_SSP.2 Mutual trusted acknowledgement**

4445 **Component relationships**

4446 Hierarchical to: FPT_SSP.1 Simple trusted acknowledgement

4447 Dependencies: FPT_ITT.1 Basic internal TSF data transfer
4448 protection

4449 **FPT_SSP.2.1**

4450 The TSF shall acknowledge, when requested by another part of the TSF, the receipt of an
4451 unmodified TSF data transmission.

4452 **FPT_SSP.2.2**

4453 **The TSF shall ensure that the relevant parts of the TSF know the correct status of**
4454 **transmitted data among its different parts, using acknowledgements.**

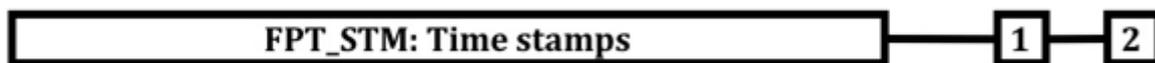
4455 **14.13 Time stamps (FPT_STM)**

4456 **14.13.1 Family behaviour**

4457 This family addresses requirements for a reliable time stamp function within a TOE.

4458 **14.13.2 Components leveling and description**

4459 Figure 72 shows the component leveling for this family.



4460

4461 **Figure 72 — FPR_STM: Component leveling**

4462 FPT_STM.1 Reliable time stamps, requires that the TSF provide reliable time stamps for TSF
4463 functions.

4464 FPT_STM.2 Time source, requires the description of the time source used in timestamps

4465 **14.13.3 Management of FPT_STM.1**

4466 The following actions could be considered for the management functions in FMT:

4467 a) Management of the time.

4468 **14.13.4 Management of FPT_STM.2**

4469 The following actions **could** be considered for the management functions in FMT:

4470 a) Setting of time by user authorized according to security policy.

4471 **14.13.5 Audit of FPT_STM.1**

4472 The following actions should be auditable if FAU_GEN Security audit data generation is included
4473 in the PP, PP-Module, functional package or ST:

4474 a) Minimal: changes to the time.

4475 b) Detailed: providing a timestamp.

4476 **14.13.6 Audit of FPT_STM.2**

4477 The following actions should be auditable if FAU_GEN Security audit data generation is included
4478 in the PP, PP-Module, functional package or ST:

4479 a) Minimal: discontinuous changes to the time;

4480 b) Detailed: changes to the time source.

4481 **14.13.7 FPT_STM.1 Reliable time stamps**

4482 **Component relationships**

4483 Hierarchical to: No other components.

4484 Dependencies: No dependencies.

4485 **FPT_STM.1.1**

4486 **The TSF **shall** be able to provide reliable time stamps.**

4487 **14.13.8 FPT_STM.2 Time source**

4488 **Component relationships**

4489 Hierarchical to: No other components.

4490 Dependencies: FPT_STM.1 Reliable time stamps

4491 FMT_SMR.1 Security roles

4492 **FPT_STM.2.1**

4493 **The TSF **shall** allow the [assignment: *user authorized by security policy*] to [assignment:
4494 *set the time, configure another time source*].**

4495 **14.14 Inter-TSF TSF data consistency (FPT_TDC)**

4496 **14.14.1 Family behaviour**

4497 In a distributed environment, a TOE **may** need to exchange TSF data with another trusted IT
4498 product. This family defines the requirements for sharing and consistent interpretation of these
4499 attributes between the TSF of the TOE and a different trusted IT product.

4500 **14.14.2 Components leveling and description**

4501 Figure 73 shows the component leveling for this family.

FPT_TDC: Inter-TSF TSF data consistency

1

4502

4503

Figure 73 — FPT_TDC: Component leveling

4504 FPT_TDC.1 Inter-TSF basic TSF data consistency, requires that the TSF provide the capability to
 4505 ensure consistency of attributes between TSFs.

4506 **14.14.3 Management of FPT_TDC.1**

4507 The following actions **could** be considered for the management functions in FMT:

- 4508 a) There are no management activities foreseen.

4509 **14.14.4 Audit of FPT_TDC.1**

4510 The following actions should be auditable if FAU_GEN Security audit data generation is included
 4511 in the PP, PP-Module, functional package or ST:

- 4512 a) Minimal: Successful use of TSF data consistency mechanisms.
 4513 b) Basic: Use of the TSF data consistency mechanisms.
 4514 c) Basic: Identification of which TSF data have been interpreted.
 4515 d) Basic: Detection of modified TSF data.

4516 **14.14.5 FPT_TDC.1 Inter-TSF basic TSF data consistency**

4517 **Component relationships**

- | | |
|-----------------------|----------------------|
| 4518 Hierarchical to: | No other components. |
| 4519 Dependencies: | No dependencies. |

4520 **FPT_TDC.1.1**

4521 **The TSF **shall** provide the capability to consistently interpret [assignment: *list of TSF data***
 4522 ***types*] when shared between the TSF and another trusted IT product.**

4523 **FPT_TDC.1.2**

4524 **The TSF **shall** use [assignment: *list of interpretation rules to be applied by the TSF*] when**
 4525 **interpreting the TSF data from another trusted IT product.**

4526 **14.15 Testing of external entities (FPT_TEE)**

4527 **14.15.1 Family behaviour**

4528 This family defines requirements for the TSF to perform tests on one or more external entities.
 4529 This component is not intended to be applied to human users.

4530 External entities **can** include applications running on the TOE, hardware or software running
 4531 “underneath” the TOE (platforms, operating systems etc.) or applications/boxes connected to
 4532 the TOE (intrusion detection systems, firewalls, login servers, time servers etc.).

4533 **14.15.2 Components leveling and description**

4534 Figure 74 shows the component leveling for this family.

FPT_TEE: Testing of external entities

1

4535

4536

Figure 74 — FPT_TEE: Component leveling

4537 FPT_TEE.1 Testing of external entities, provides for testing of the external entities by the TSF.

4538 **14.15.3 Management of FPT_TEE.1**

4539 The following actions **could** be considered for the management functions in FMT:

- 4540 a) Management of the conditions under which the testing of external entities occurs,
4541 such as during initial start-up, regular interval, or under specified conditions;
4542 b) Management of the time interval if appropriate.

4543 **14.15.4 Audit of FPT_TEE.1**

4544 The following actions should be auditable if FAU_GEN Security audit data generation is included
4545 in the PP, PP-Module, functional package or ST:

- 4546 a) Basic: Execution of the tests of the external entities and the results of the tests.

4547 **14.15.5 FPT_TEE.1 Testing of external entities**4548 **Component relationships**

4549 Hierarchical to: No other components.

4550 Dependencies: No dependencies.

4551 **FPT_TEE.1.1**

4552 **The TSF shall run a suite of tests [selection: *during initial start-up, periodically during***
4553 ***normal operation, at the request of an authorized user, [assignment: other conditions]] to***
4554 **check the fulfillment of [assignment: *list of properties of the external entities*].**

4555 **FPT_TEE.1.2**

4556 **If the test fails, the TSF shall [assignment: *action(s)*].**

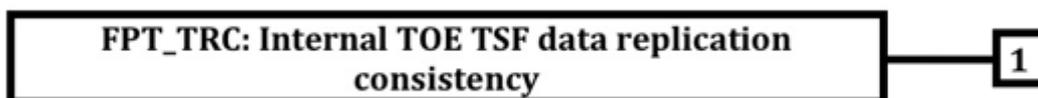
4557 **14.16 Internal TOE TSF data replication consistency (FPT_TRC)**4558 **14.16.1 Family behaviour**

4559 The requirements of this family are needed to ensure the consistency of TSF data when such
4560 data is replicated internal to the TOE. Such data **may** become inconsistent if the internal channel
4561 between parts of the TOE becomes inoperative. If the TOE is internally structured as a network
4562 and parts of the TOE network connections are broken, this **may** occur when parts become
4563 disabled.

4564 **14.16.2 Components leveling and description**

4565 Figure 75 shows the component leveling for this family.

4566



4567

Figure 75 — FPT_TRC: Component leveling

4568 This family consists of only one component, FPT_TRC.1 Internal TSF consistency, which
4569 requires that the TSF ensure the consistency of TSF data that is replicated in multiple locations.

4570 **14.16.3 Management of FPT_TRC.1**

4571 The following actions **could** be considered for the management functions in FMT:

- 4572 a) There are no management activities foreseen.

4573 **14.16.4 Audit of FPT_TRC.1**

4574 The following actions should be auditable if FAU_GEN Security audit data generation is included
4575 in the PP, PP-Module, functional package or ST:

- 4576 a) Minimal: restoring consistency upon reconnection;
4577 b) Basic: Detected inconsistency between TSF data.

4578 **14.16.5 FPT_TRC.1 Internal TSF consistency**

4579 **Component relationships**

4580	Hierarchical to:	No other components.
4581	Dependencies:	FPT_ITT.1 Basic internal TSF data transfer
4582		protection

4583 **FPT_TRC.1.1**

4584 **The TSF shall ensure that TSF data is consistent when replicated between parts of the**
4585 **TOE.**

4586 **FPT_TRC.1.2**

4587 **When parts of the TOE containing replicated TSF data are disconnected, the TSF shall**
4588 **ensure the consistency of the replicated TSF data upon reconnection before processing**
4589 **any requests for [assignment: *list of functions dependent on TSF data replication***
4590 ***consistency*].**

4591 **14.17 TSF self-test (FPT_TST)**

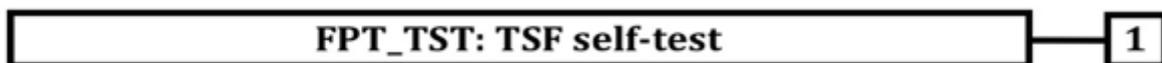
4592 **14.17.1 Family behaviour**

4593 The family defines the requirements for the self-testing of the TSF with respect to some
4594 expected correct operation. Examples are interfaces to enforcement functions, and sample
4595 arithmetical operations on critical parts of the TOE. These tests **can** be carried out at start-up,
4596 periodically, at the request of the authorized user, or when other conditions are met. The
4597 actions to be taken by the TOE as the result of self-testing are defined in other families.

4598 The requirements of this family are also needed to detect the corruption of TSF data and TSF
4599 itself (i.e. TSF executable code or TSF hardware component) by various failures that do not
4600 necessarily stop the TOE's operation (which would be handled by other families). These checks
4601 must be performed because these failures **cannot** necessarily be prevented. Such failures **can**
4602 occur either because of unforeseen failure modes or associated oversights in the design of
4603 hardware, firmware, or software, or because of malicious corruption of the TSF due to
4604 inadequate logical and/or physical protection.

4605 **14.17.2 Components leveling and description**

4606 Figure 76 shows the component leveling for this family.



4607

4608 **Figure 76 — FPT_TST: Component leveling**

4609 FPT_TST.1 TSF self-testing, provides the ability to test the TSF's correct operation. These tests
 4610 **can** be performed at start-up, periodically, at the request of the authorized user, or when other
 4611 conditions are met. It also provides the ability to verify the integrity of TSF data and TSF itself.

4612 **14.17.3 Management of FPT_TST.1**

4613 The following actions **could** be considered for the management functions in FMT:

- 4614 a) Management of the conditions under which TSF self-testing occurs, such as during
- 4615 initial start-up, regular interval, or under specified conditions;
- 4616 b) Management of the time interval if appropriate.

4617 **14.17.4 Audit of FPT_TST.1**

4618 The following actions should be auditable if FAU_GEN Security audit data generation is included
 4619 in the PP, PP-Module, functional package or ST:

- 4620 a) Minimal: Indication that the TSF self-tests were completed and any failures of the
- 4621 tests.
- 4622 b) Basic: Execution of the TSF self-tests and the results of the tests.

4623 **14.17.5 FPT_TST.1 TSF self-testing**

4624 **Component relationships**

4625 Hierarchical to: No other components.

4626 Dependencies: No dependencies.

4627 **FPT_TST.1.1**

4628 **The TSF shall run a suite of the following self-tests [selection: *during initial start-up,***
 4629 ***periodically during normal operation, at the request of the authorized user, at the***
 4630 ***conditions [assignment: conditions under which self-test should occur]] to demonstrate the***
 4631 **correct operation of [selection: [assignment: parts of TSF], the TSF]: [assignment: list of**
 4632 ***self-tests run by the TSF].***

4633 **FPT_TST.1.2**

4634 **The TSF shall provide authorized users with the capability to verify the integrity of**
 4635 **[selection: [assignment: parts of TSF data], TSF data].**

4636 **FPT_TST.1.3**

4637 **The TSF shall provide authorized users with the capability to verify the integrity of**
 4638 **[selection: [assignment: parts of TSF], TSF].**

4639

4640 **15 Class FRU: Resource utilization**

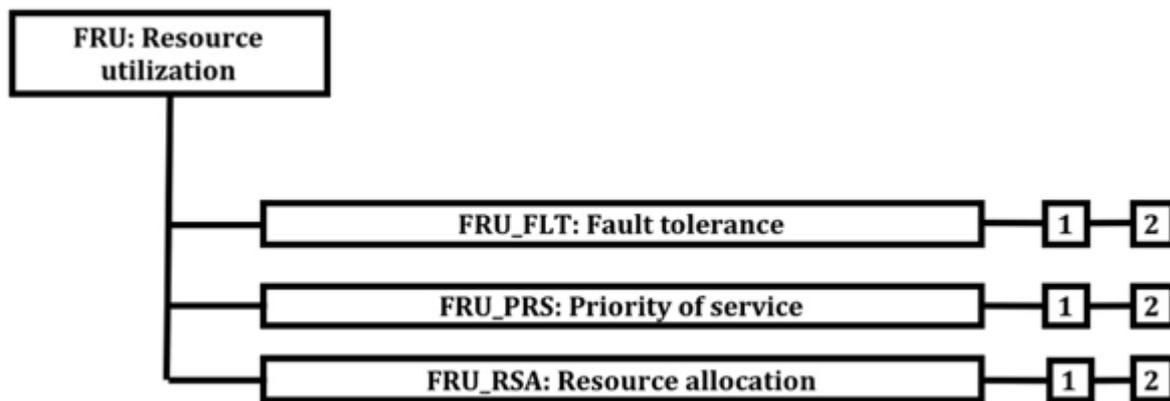
4641 **15.1 Class description**

4642 This class provides three families that support the availability of required resources such as
 4643 processing capability and/or storage capacity. The family Fault Tolerance provides protection
 4644 against unavailability of capabilities caused by failure of the TOE. The family Priority of Service
 4645 ensures that the resources will be allocated to the more important or time-critical tasks and
 4646 cannot be monopolized by lower priority tasks. The family Resource Allocation provides limits
 4647 on the use of available resources, therefore preventing users from monopolizing the resources.

4648 Figure 77 shows the decomposition of this class, it's families and components. Elements are not
 4649 shown in the figure.

4650 Annex K provides explanatory information for this class and **should** be consulted when using
 4651 the components identified in this class.

4652



4653 **Figure 77 — FRU: Resource utilization class decomposition**

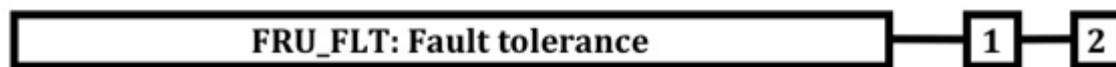
4654 **15.2 Fault tolerance (FRU_FLT)**

4655 **15.2.1 Family behaviour**

4656 The requirements of this family ensure that the TOE will maintain correct operation even in the
 4657 event of failures.

4658 **15.2.2 Components leveling and description**

4659 Figure 78 shows the component leveling for this family.



4660

4661 **Figure 78 — FRU_FLT: Component leveling**

4662 FRU_FLT.1 Degraded fault tolerance, requires the TOE to continue correct operation of
 4663 identified capabilities in the event of identified failures.

4664 FRU_FLT.2 Limited fault tolerance, requires the TOE to continue correct operation of all
 4665 capabilities in the event of identified failures.

4666 **15.2.3 Management of FRU_FLT.1, FRU_FLT.2**

4667 The following actions **could** be considered for the management functions in FMT:

4668 a) There are no management activities foreseen.

4669 **15.2.4 Audit of FRU_FLT.1**

4670 The following actions should be auditable if FAU_GEN Security audit data generation is included
4671 in the PP, PP-Module, functional package or ST:

4672 a) Minimal: Any failure detected by the TSF.

4673 b) Basic: All TOE capabilities being discontinued due to a failure.

4674 **15.2.5 Audit of FRU_FLT.2**

4675 The following actions should be auditable if FAU_GEN Security audit data generation is included
4676 in the PP, PP-Module, functional package or ST:

4677 a) Minimal: Any failure detected by the TSF.

4678 **15.2.6 FRU_FLT.1 Degraded fault tolerance**

4679 **Component relationships**

4680 Hierarchical to: No other components.

4681 Dependencies: FPT_FLS.1 Failure with preservation of secure state

4682 **FRU_FLT.1.1**

4683 **The TSF shall ensure the operation of [assignment: list of TOE capabilities] when the**
4684 **following failures occur: [assignment: list of type of failures].**

4685 **15.2.7 FRU_FLT.2 Limited fault tolerance**

4686 **Component relationships**

4687 Hierarchical to: FRU_FLT.1 Degraded fault tolerance

4688 Dependencies: FPT_FLS.1 Failure with preservation of secure state

4689 **FRU_FLT.2.1**

4690 The TSF shall ensure the operation of **all the TOE's capabilities** when the following failures
4691 occur: [assignment: list of type of failures].

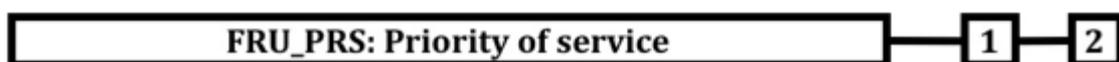
4692 **15.3 Priority of service (FRU_PRS)**

4693 **15.3.1 Family behaviour**

4694 The requirements of this family allow the TSF to control the use of resources under the control
4695 of the TSF by users and subjects such that high priority activities under the control of the TSF
4696 will always be accomplished without undue interference or delay caused by low priority
4697 activities.

4698 **15.3.2 Components leveling and description**

4699 Figure 79 shows the component leveling for this family.



4700

4701

Figure 79 — FRU_PRS: Component leveling

4702 FRU_PRS.1 Limited priority of service, provides priorities for a subject's use of a subset of the
4703 resources under the control of the TSF.

4704 FRU_PRS.2 Full priority of service, provides priorities for a subject's use of all of the resources
4705 under the control of the TSF.

4706 **15.3.3 Management of FRU_PRS.1, FRU_PRS.2**

4707 The following actions **could** be considered for the management functions in FMT:

4708 a) Assignment of priorities to each subject in the TSF.

4709 **15.3.4 Audit of FRU_PRS.1, FRU_PRS.2**

4710 The following actions should be auditable if FAU_GEN Security audit data generation is included
4711 in the PP, PP-Module, functional package or ST:

4712 a) Minimal: Rejection of operation based on the use of priority within an allocation.

4713 b) Basic: All attempted uses of the allocation function which involves the priority of
4714 the service functions.

4715 **15.3.5 FRU_PRS.1 Limited priority of service**

4716 Hierarchical to: No other components.

4717 Dependencies: No dependencies.

4718 **FRU_PRS.1.1**

4719 The TSF **shall** assign a priority to each subject in the TSF.

4720 **FRU_PRS.1.2**

4721 The TSF **shall** ensure that each access to [assignment: *controlled resources*] **shall be**
4722 **mediated on the basis of the subjects assigned priority.**

4723 **15.3.6 FRU_PRS.2 Full priority of service**

4724 **Component relationships**

4725 Hierarchical to: FRU_PRS.1 Limited priority of service

4726 Dependencies: No dependencies.

4727 **FRU_PRS.2.1**

4728 The TSF **shall** assign a priority to each subject in the TSF.

4729 **FRU_PRS.2.2**

4730 The TSF **shall** ensure that each access to **all shareable resources shall** be mediated on the
4731 basis of the subjects assigned priority.

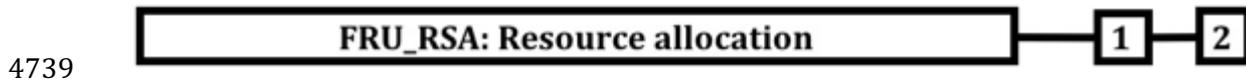
4732 **15.4 Resource allocation (FRU_RSA)**

4733 **15.4.1 Family behaviour**

4734 The requirements of this family allow the TSF to control the use of resources by users and
4735 subjects such that denial of service will not occur because of unauthorized monopolization of
4736 resources.

4737 **15.4.2 Components leveling and description**

4738 Figure 80 shows the component leveling for this family.

4740 **Figure 80 — FRU_RSA: Component leveling**4741 FRU_RSA.1 Maximum quotas, provides requirements for quota mechanisms that ensure that
4742 users and subjects will not monopolize a controlled resource.4743 FRU_RSA.2 Minimum and maximum quotas, provides requirements for quota mechanisms that
4744 ensure that users and subjects will always have at least a minimum of a specified resource and
4745 that they will not be able to monopolize a controlled resource.4746 **15.4.3 Management of FRU_RSA.1**4747 The following actions **could** be considered for the management functions in FMT:

- 4748 a) Specifying maximum limits for a resource for groups and/or individual users
-
- 4749 and/or subjects by an administrator.

4750 **15.4.4 Management of FRU_RSA.2**4751 The following actions **could** be considered for the management functions in FMT:

- 4752 a) Specifying minimum and maximum limits for a resource for groups and/or
-
- 4753 individual users and/or subjects by an administrator.

4754 **15.4.5 Audit of FRU_RSA.1, FRU_RSA.2**4755 The following actions should be auditable if FAU_GEN Security audit data generation is included
4756 in the PP, PP-Module, functional package or ST:

- 4757 a) Minimal: Rejection of allocation operation due to resource limits.
-
- 4758 b) Basic: All attempted uses of the resource allocation functions for resources that are
-
- 4759 under control of the TSF.

4760 **15.4.6 FRU_RSA.1 Maximum quotas**4761 **Component relationships**

4762 Hierarchical to: No other components.

4763 Dependencies: No dependencies.

4764 **FRU_RSA.1.1**4765 **The TSF shall enforce maximum quotas of the following resources: [assignment:**
4766 **controlled resources] that [selection: individual user, defined group of users, subjects] can**
4767 **use [selection: simultaneously, over a specified period of time].**4768 **15.4.7 FRU_RSA.2 Minimum and maximum quotas**4769 **Component relationships**

4770 Hierarchical to: FRU_RSA.1 Maximum quotas

4771 Dependencies: No dependencies.

4772 **FRU_RSA.2.1**

4773 The TSF **shall** enforce maximum quotas of the following resources [assignment: *controlled*
4774 *resources*] that [selection: *individual user, defined group of users, subjects*] **can** use [selection:
4775 *simultaneously, over a specified period of time*].

4776 **FRU_RSA.2.2**

4777 **The TSF shall ensure the provision of minimum quantity of each [assignment: *controlled***
4778 ***resource*] that is available for [selection: *an individual user, defined group of users,***
4779 ***subjects*] to use [selection: *simultaneously, over a specified period of time*].**

4780

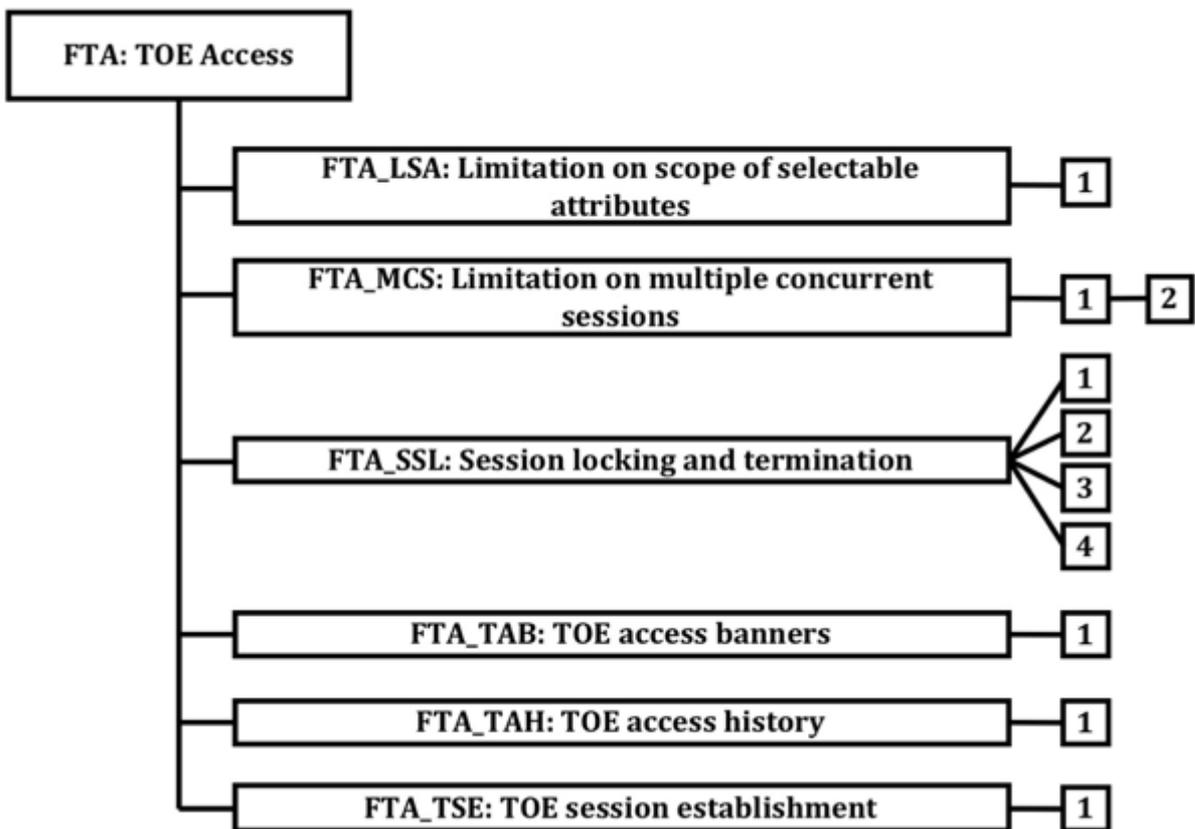
4781 **16 Class FTA: TOE access**

4782 **16.1 Class description**

4783 This family specifies functional requirements for controlling the establishment of a user's
4784 session.

4785 Figure 81 shows the decomposition of this class, it's families and components. Elements are not
4786 shown in the figure.

4787 Annex L provides explanatory information for this class and **should** be consulted when using
4788 the components identified in this class.



4789 **Figure 81 — FTA: TOE access class decomposition**

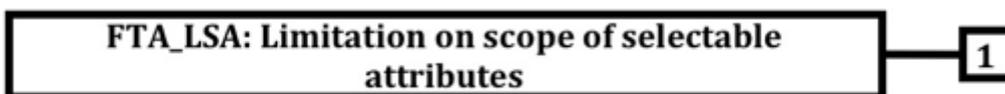
4790 **16.2 Limitation on scope of selectable attributes (FTA_LSA)**

4791 **16.2.1 Family behaviour**

4792 This family defines requirements to limit the scope of session security attributes that a user **can**
4793 select for a session.

4794 **16.2.2 Components leveling and description**

4795 Figure 82 shows the component leveling for this family.



4796

4797 **Figure 82 — FTA_LSA: Component leveling**

4798 FTA_LSA.1 Limitation on scope of selectable attributes, provides the requirement for a TOE to
 4799 limit the scope of the session security attributes during session establishment.

4800 **16.2.3 Management of FTA_LSA.1**

4801 The following actions **could** be considered for the management functions in FMT:

- 4802 a) Management of the scope of the session security attributes by an administrator.

4803 **16.2.4 Audit of FTA_LSA.1**

4804 The following actions should be auditable if FAU_GEN Security audit data generation is included
 4805 in the PP, PP-Module, functional package or ST:

- 4806 a) Minimal: All failed attempts at selecting session security attributes.
- 4807 b) Basic: All attempts at selecting session security attributes.
- 4808 c) Detailed: Capture of the values of each of the session security attributes.

4809 **16.2.5 FTA_LSA.1 Limitation on scope of selectable attributes**

4810 **Component relationships**

- 4811 Hierarchical to: No other components.
- 4812 Dependencies: No dependencies.

4813 **FTA_LSA.1.1**

4814 The TSF **shall** restrict the scope of the session security attributes [assignment: *session*
 4815 *security attributes*], based on [assignment: *attributes*].

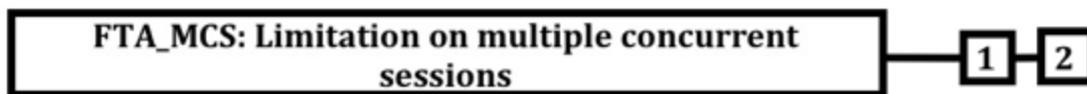
4816 **16.3 Limitation on multiple concurrent sessions (FTA_MCS)**

4817 **16.3.1 Family behaviour**

4818 This family defines requirements to place limits on the number of concurrent sessions that
 4819 belong to the same user.

4820 **16.3.2 Components leveling and description**

4821 Figure 83 shows the component leveling for this family.



4822

4823 **Figure 83 — FTA_MCS: Component leveling**

4824 FTA_MCS.1 Basic limitation on multiple concurrent sessions, provides limitations that apply to
 4825 all users of the TSF.

4826 FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions extends FTA_MCS.1
 4827 Basic limitation on multiple concurrent sessions by requiring the ability to specify limitations
 4828 on the number of concurrent sessions based on the related security attributes.

4829 **16.3.3 Management of FTA_MCS.1**

4830 The following actions **could** be considered for the management functions in FMT:

- 4831 a) Management of the maximum allowed number of concurrent user sessions by an
 4832 administrator.

4833 **16.3.4 Management of FTA_MCS.2**4834 The following actions **could** be considered for the management functions in FMT:

- 4835 a) Management of the rules that govern the maximum allowed number of concurrent
-
- 4836 user sessions by an administrator.

4837 **16.3.5 Audit of FTA_MCS.1, FTA_MCS.2**4838 The following actions should be auditable if FAU_GEN Security audit data generation is included
4839 in the PP, PP-Module, functional package or ST:

- 4840 a) Minimal: Rejection of a new session based on the limitation of multiple concurrent
-
- 4841 sessions.
-
- 4842 b) Detailed: Capture of the number of currently concurrent user sessions and the user
-
- 4843 security attribute(s).

4844 **16.3.6 FTA_MCS.1 Basic limitation on multiple concurrent sessions**4845 **Component relationships**

4846 Hierarchical to: No other components.

4847 Dependencies: FIA_UID.1 Timing of identification

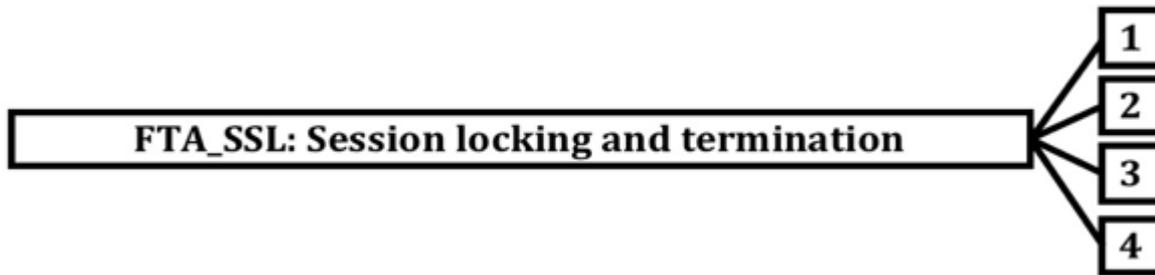
4848 **FTA_MCS.1.1**4849 The TSF **shall** restrict the maximum number of concurrent sessions that belong to the
4850 same user.4851 **FTA_MCS.1.2**4852 The TSF **shall** enforce, by default, a limit of [assignment: *default number*] sessions per
4853 user.4854 **16.3.7 FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions**4855 **Component relationships**4856 Hierarchical to: FTA_MCS.1 Basic limitation on multiple concurrent
4857 sessions

4858 Dependencies: FIA_UID.1 Timing of identification

4859 **FTA_MCS.2.1**4860 The TSF **shall** restrict the maximum number of concurrent sessions that belong to the same user
4861 according to the rules [assignment: *rules for the number of maximum concurrent*
4862 *sessions*].4863 **FTA_MCS.2.2**4864 The TSF **shall** enforce, by default, a limit of [assignment: *default number*] sessions per user.4865 **16.4 Session locking and termination (FTA_SSL)**4866 **16.4.1 Family behaviour**4867 This family defines requirements for the TSF to provide the capability for TSF-initiated and
4868 user-initiated locking, unlocking, and termination of interactive sessions.

4869 **16.4.2 Components leveling and description**

4870 Figure 84 shows the component leveling for this family.



4871

4872 **Figure 84 — FTA_SSL: Component leveling**

4873 FTA_SSL.1 TSF-initiated session locking includes system-initiated locking of an interactive
4874 session after a specified period of user inactivity.

4875 FTA_SSL.2 User-initiated locking, provides capabilities for the user to lock and unlock the user's
4876 own interactive sessions.

4877 FTA_SSL.3 TSF-initiated termination, provides requirements for the TSF to terminate the
4878 session after a specified period of user inactivity.

4879 FTA_SSL.4 User-initiated termination, provides capabilities for the user to terminate the user's
4880 own interactive sessions.

4881 **16.4.3 Management of FTA_SSL.1**

4882 The following actions **could** be considered for the management functions in FMT:

- 4883 a) Specification of the time of user inactivity after which lock-out occurs for an
4884 individual user;
- 4885 b) Specification of the default time of user inactivity after which lock-out occurs;
- 4886 c) Management of the events that occur prior to unlocking the session.

4887 **16.4.4 Management of FTA_SSL.2**

4888 The following actions **could** be considered for the management functions in FMT:

- 4889 a) Management of the events that occur prior to unlocking the session.

4890 **16.4.5 Management of FTA_SSL.3**

4891 The following actions **could** be considered for the management functions in FMT:

- 4892 a) Specification of the time of user inactivity after which termination of the interactive
4893 session occurs for an individual user;
- 4894 b) Specification of the default time of user inactivity after which termination of the
4895 interactive session occurs.

4896 **16.4.6 Management of FTA_SSL.4**

4897 The following actions **could** be considered for the management functions in FMT:

- 4898 a) There are no management activities foreseen.

4899 **16.4.7 Audit of FTA_SSL.1, FTA_SSL.2**

4900 The following actions should be auditable if FAU_GEN Security audit data generation is included
4901 in the PP, PP-Module, functional package or ST:

- 4902 a) Minimal: Locking of an interactive session by the session locking mechanism.
 4903 b) Minimal: Successful unlocking of an interactive session.
 4904 c) Basic: Any attempts at unlocking an interactive session.

4905 16.4.8 Audit of FTA_SSL.3

4906 The following actions should be auditable if FAU_GEN Security audit data generation is included
 4907 in the PP, PP-Module, functional package or ST:

- 4908 a) Minimal: Termination of an interactive session by the session locking mechanism.

4909 16.4.9 Audit of FTA_SSL.4

4910 The following actions should be auditable if FAU_GEN Security audit data generation is included
 4911 in the PP, PP-Module, functional package or ST:

- 4912 a) Minimal: Termination of an interactive session by the user.

4913 16.4.10 FTA_SSL.1 TSF-initiated session locking

4914 Component relationships

4915 Hierarchical to: No other components.

4916 Dependencies: FIA_UAU.1 Timing of authentication

4917 FTA_SSL.1.1

4918 The TSF **shall** lock an interactive session after [assignment: *time interval of user*
 4919 *inactivity*] by:

- 4920 a) clearing or overwriting display devices, making the current contents
 4921 unreadable;
 4922 b) disabling any activity of the user's data access/display devices other than
 4923 unlocking the session.

4924 FTA_SSL.1.2

4925 The TSF **shall** require the following events to occur prior to unlocking the session:
 4926 [assignment: *events to occur*].

4927 16.4.11 FTA_SSL.2 User-initiated locking

4928 Component relationships

4929 Hierarchical to: No other components.

4930 Dependencies: FIA_UAU.1 Timing of authentication

4931 FTA_SSL.2.1

4932 The TSF **shall** allow user-initiated locking of the user's own interactive session, by:

- 4933 a) clearing or overwriting display devices, making the current contents
 4934 unreadable;
 4935 b) disabling any activity of the user's data access/display devices other than
 4936 unlocking the session.

4937 FTA_SSL.2.2

4938 The TSF **shall** require the following events to occur prior to unlocking the session:
 4939 [assignment: *events to occur*].

4940 **16.4.12 FTA_SSL.3 TSF-initiated termination**

4941 **Component relationships**

4942 Hierarchical to: No other components.

4943 Dependencies: FMT_SMR.1 Security roles

4944 **FTA_SSL.3.1**

4945 **The TSF shall terminate an interactive session after a [assignment: time interval of user**
 4946 **inactivity].**

4947 **16.4.13 FTA_SSL.4 User-initiated termination**

4948 **Component relationships**

4949 Hierarchical to: No other components.

4950 Dependencies: No dependencies.

4951 **FTA_SSL.4.1**

4952 **The TSF shall allow user-initiated termination of the user's own interactive session.**

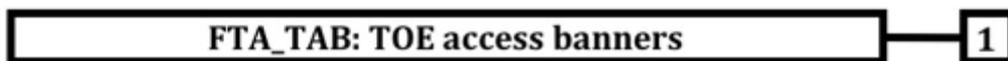
4953 **16.5 TOE access banners (FTA_TAB)**

4954 **16.5.1 Family behaviour**

4955 This family defines requirements to display a configurable advisory warning message to users
 4956 regarding the appropriate use of the TOE.

4957 **16.5.2 Components leveling and description**

4958 Figure 85 shows the component leveling for this family.



4959

4960 **Figure 85 — FTA_TAB: Component leveling**

4961 FTA_TAB.1 Default TOE access banners, provides the requirement for a TOE Access Banner.
 4962 This banner is displayed prior to the establishment dialogue for a session.

4963 **16.5.3 Management of FTA_TAB.1**

4964 The following actions could be considered for the management functions in FMT:

- 4965 a) Maintenance of the banner by the authorized administrator.

4966 **16.5.4 Audit of FTA_TAB.1**

4967 The following actions should be auditable if FAU_GEN Security audit data generation is included
 4968 in the PP, PP-Module, functional package or ST:

- 4969 a) There are no auditable events foreseen.

4970 **16.5.5 FTA_TAB.1 Default TOE access banners**

4971 **Component relationships**

4972 Hierarchical to: No other components.

4973 Dependencies: No dependencies.

4974 FTA_TAB.1.1

4975 Before establishing a user session, the [selection: *TSF, TOE platform*] shall display an
4976 [assignment: *description of the message*] message.

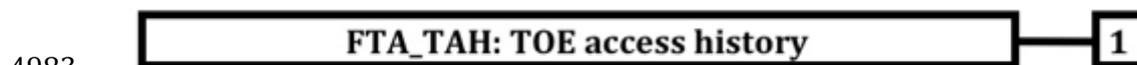
4977 16.6 TOE access history (FTA_TAH)

4978 16.6.1 Family behaviour

4979 This family defines requirements for the TSF to display to a user, upon successful session
4980 establishment, a history of successful and unsuccessful attempts to access the user's account.

4981 16.6.2 Components leveling and description

4982 Figure 86 shows the component leveling for this family.



4984 **Figure 86 — FTA_TAH: Component leveling**

4985 FTA_TAH.1 TOE access history, provides the requirement for a TOE to display information
4986 related to previous attempts to establish a session.

4987 16.6.3 Management of FTA_TAH.1

4988 The following actions could be considered for the management functions in FMT:

4989 a) There are no management activities foreseen.

4990 16.6.4 Audit of FTA_TAH.1

4991 The following actions should be auditable if FAU_GEN Security audit data generation is included
4992 in the PP, PP-Module, functional package or ST:

4993 a) There are no auditable events foreseen.

4994 16.6.5 FTA_TAH.1 TOE access history

4995 Component relationships

4996 Hierarchical to: No other components.

4997 Dependencies: No dependencies.

4998 FTA_TAH.1.1

4999 Upon successful session establishment, the TSF shall display the [selection: *date, time,*
5000 *method, location*] of the last successful session establishment to the user.

5001 FTA_TAH.1.2

5002 Upon successful session establishment, the TSF shall display the [selection: *date, time,*
5003 *method, location*] of the last unsuccessful attempt to session establishment and the
5004 number of unsuccessful attempts since the last successful session establishment.

5005 FTA_TAH.1.3

5006 The TSF shall not erase the access history information from the user interface without
5007 giving the user an opportunity to review the information.

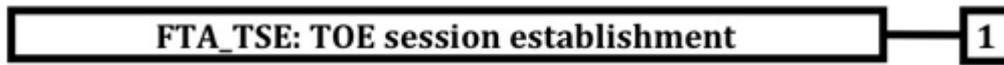
5008 **16.7 TOE session establishment (FTA_TSE)**

5009 **16.7.1 Family behaviour**

5010 This family defines requirements to deny a user permission to establish a session with the TOE.

5011 **16.7.2 Components leveling and description**

5012 Figure 87 shows the component leveling for this family.



5013

5014 **Figure 87 — FTA_TSE: Component leveling**

5015 FTA_TSE.1 TOE session establishment, provides requirements for denying users access to the
5016 TOE based on attributes.

5017 **16.7.3 Management of FTA_TSE.1**

5018 The following actions **could** be considered for the management functions in FMT:

- 5019 a) Management of the session establishment conditions by the authorized
5020 administrator.

5021 **16.7.4 Audit of FTA_TSE.1**

5022 The following actions should be auditable if FAU_GEN Security audit data generation is included
5023 in the PP, PP-Module, functional package or ST:

- 5024 a) Minimal: Denial of a session establishment due to the session establishment
5025 mechanism.
- 5026 b) Basic: All attempts at establishment of a user session.
- 5027 c) Detailed: Capture of the value of the selected access parameters.

5028 **16.7.5 FTA_TSE.1 TOE session establishment**

5029 **Component relationships**

5030 Hierarchical to: No other components.

5031 Dependencies: No dependencies.

5032 **FTA_TSE.1.1**

5033 **The TSF **shall** be able to deny session establishment based on [assignment: *attributes*].**

5034

5035 **17 Class FTP: Trusted path/channels**5036 **17.1 Class description**

5037 Families in this class provide requirements for a trusted communication path between users
5038 and the TSF, and for a trusted communication channel between the TSF and other trusted IT
5039 products. Trusted paths and channels have the following general characteristics:

- 5040 — The communications path is constructed using internal and external communications
5041 channels (as appropriate for the component) that isolate an identified subset of TSF data
5042 and commands from the remainder of the TSF and user data.
- 5043 — Use of the communications path **can** be initiated by the user and/or the TSF (as appropriate
5044 for the component).
- 5045 — The communications path is capable of providing assurance that the user is communicating
5046 with the correct TSF, and that the TSF is communicating with the correct user (as
5047 appropriate for the component).

5048 In this paradigm, a trusted channel is a communication channel that **can** be initiated by either
5049 side of the channel and provides non-repudiation characteristics with respect to the identity of
5050 the sides of the channel.

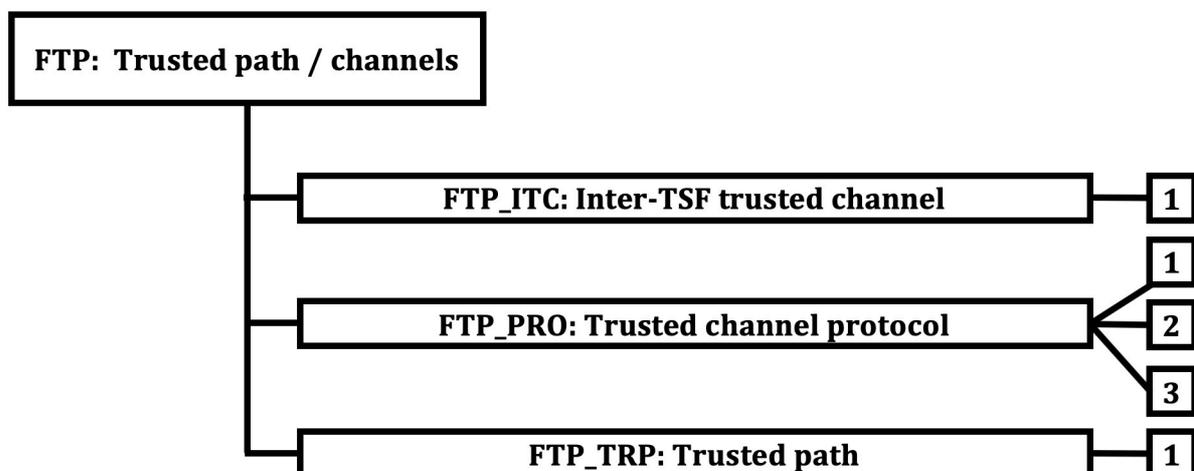
5051 A trusted path provides a means for users to perform functions through an assured direct
5052 interaction with the TSF. Trusted path is usually desired for user actions such as initial
5053 identification and/or authentication but **can** also be desired at other times during a user's
5054 session. Trusted path exchanges **can** be initiated by a user or the TSF. User responses via the
5055 trusted path are guaranteed to be protected from modification by or disclosure to untrusted
5056 applications.

5057 Families describing the use of commonly used communication protocols used in the provision
5058 of trusted channels and paths are also given.

5059 Figure 88 shows the decomposition of this class, it's families and components. Elements are not
5060 shown in the figure.

5061 Annex M provides explanatory information for this class and **should** be consulted when using
5062 the components identified in this class.

5063

5064 **Figure 88 — FTP: Trusted path/channels class decomposition**

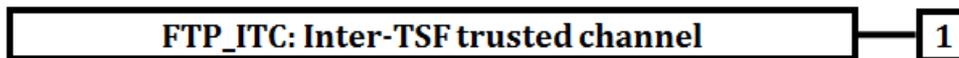
5065 **17.2 Inter-TSF trusted channel (FTP_ITC)**

5066 **17.2.1 Family behaviour**

5067 This family defines requirements for the creation of a trusted channel between the TSF and
 5068 other trusted IT products for the performance of security critical operations. The components
 5069 of this family **may** be included whenever there are requirements for the secure communication
 5070 of user or TSF data between the TOE and other trusted IT products.

5071 **17.2.2 Components leveling and description**

5072 Figure 89 shows the component leveling for this family.



5073 **Figure 89 — FTP_ITC: Component leveling**

5074 FTP_ITC.1 Inter-TSF trusted channel, requires that the TSF provide a trusted communication
 5075 channel between itself and another trusted IT product.

5076 **17.2.3 Management of FTP_ITC.1**

5077 The following actions **could** be considered for the management functions in FMT:

- 5078 a) Configuring the actions that require trusted channel, if supported.

5079 **17.2.4 Audit of FTP_ITC.1**

5080 The following actions should be auditable if FAU_GEN Security audit data generation is included
 5081 in the PP, PP-Module, functional package or ST:

- 5082 a) Minimal: Failure of the trusted channel functions.
- 5083 b) Minimal: Identification of the initiator and target of failed trusted channel
 5084 functions.
- 5085 c) Basic: All attempted uses of the trusted channel functions.
- 5086 d) Basic: Identification of the initiator and target of all trusted channel functions.

5087 **17.2.5 FTP_ITC.1 Inter-TSF trusted channel**

5088 **Component relationships**

5089 Hierarchical to: No other components.

5090 Dependencies: No dependencies.

5091 **FTP_ITC.1.1**

5092 **The TSF shall provide a communication channel between itself and another trusted IT**
 5093 **product that is logically distinct from other communication channels and provides**
 5094 **assured identification of its end points and protection of the channel data from**
 5095 **modification or disclosure.**

5096 **FTP_ITC.1.2**

5097 **The TSF shall permit [selection: *the TSF, another trusted IT product*] to initiate**
 5098 **communication via the trusted channel.**

5099 **FTP_ITC.1.3**

5100 **The TSF shall initiate communication via the trusted channel for [assignment: list of**
 5101 **functions for which a trusted channel is required].**

5102 **17.3 Trusted channel protocol (FTP_PRO)**5103 **17.3.1 Family behavior**

5104 This family defines requirements for establishing a trusted channel and using the trusted
 5105 channel to transfer the TSF data or user data securely.

5106 **17.3.2 Components leveling and description**

5107 Figure 90 shows the component leveling for this family.



Figure 90 — FTP_PRO: Component leveling

5108 FTP_PRO.1 Trusted channel protocol requires that communication be established in accordance
 5109 with a defined protocol.

5110 FTP_PRO.2 Trusted channel establishment requires that keys be securely established between
 5111 the peers.

5112 FTP_PRO.3 Trusted channel data protection requires that data in transit be protected.

5113 **17.3.3 Management of FTP_PRO.1**

5114 The following actions **could** be considered for the management functions in FMT:

- 5115 a) Configuring the protocols needed for the trusted channel,
- 5116 b) Configuring the credentials for using the trusted channel,
- 5117 c) Configuring the conditions for initializing and terminating the trusted channel.

5118 **17.3.4 Management of FTP_PRO.2**

5119 The following actions **could** be considered for the management functions in FMT:

- 5120 a) Configuring the parameters for shared secrets,
- 5121 b) Configuring the parameters for cryptographic key derivation.

5122 **17.3.5 Management of FTP_PRO.3**

5123 The following actions **could** be considered for the management functions in FMT:

- 5124 a) Configuring the encryption and integrity mechanisms used by the trusted channel.

5125

5126 **17.3.6 Audit of FTP_PRO.1**

5127 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 5128 in the PP, PP-Module, functional package or ST:

- 5129 a) Minimal: Failure of the trusted channel establishment
- 5130 b) Minimal: Identification of the initiator and target of failed trusted channel
- 5131 establishment
- 5132 c) Basic: All attempted uses of the trusted channel
- 5133 d) Basic: Identification of the initiator and target of all trusted channel attempts.
- 5134 Other events should be considered according to the specific protocols used.

5135 **17.3.7 Audit of FTP_PRO.2**

5136 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
5137 in the PP/ST:

- 5138 a) Minimal: Authentication failures during channel establishment
- 5139 b) Basic: All authentication attempts.

5140 **17.3.8 Audit of FTP_PRO.3**

5141 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
5142 in the PP/ST:

- 5143 a) Minimal: Failures when attempting to verify channel properties in FTP_PRO.3.2.

5144 **17.3.9 FTP_PRO.1 Trusted channel protocol**

5145 **Component relationships**

- 5146 Hierarchical to: No other components.
- 5147 Dependencies: FTP_PRO.2 Trusted channel establishment
- 5148 FTP_PRO.3 Trusted channel data protection.

5149 **FTP_PRO.1.1**

5150 **The TSF shall implement [assignment: *trusted channel protocol*] acting as [assignment:**
5151 ***defined protocol role(s)*] in accordance with: [assignment: *list of standards*].**

5152 **FTP_PRO.1.2**

5153 **The TSF shall enforce usage of the trusted channel for [assignment: *purpose(s) of the***
5154 ***trusted channel*] in accordance with: [assignment: *list of standards*].**

5155 **FTP_PRO.1.3**

5156 **The TSF shall permit [selection: *itself, its peer*] to initiate communication via the trusted**
5157 **channel.**

5158 **FTP_PRO.1.4**

5159 **The TSF shall enforce the following rules for the trusted channel: [assignment: *rules***
5160 ***governing operation and use of the trusted channel and/or its protocol*].**

5161 **FTP_PRO.1.5**

5162 **The TSF shall enforce the following static protocol options: [assignment: *list of options***
5163 ***and references to standards in which each is defined*].**

5164 **FTP_PRO.1.6**

5165 **The TSF shall negotiate one of the following protocol configurations with its peer:**
5166 **[assignment: *list of configurations and reference to standards in which each is defined*].**

5167 **17.3.10 FTP_PRO.2 Trusted channel establishment**5168 **Component relationships**

5169	Hierarchical to:	No other components.
5170	Dependencies:	FTP_PRO.1 Trusted channel protocol
5171		[FCS_CKM.1 Cryptographic key generation, or
5172		FCS_CKM.2 Cryptographic key distribution]
5173		FCS_CKM.5 Cryptographic key derivation
5174		FCS_COP.1 Cryptographic operation.

5175 **FTP_PRO.2.1**

5176 **The TSF shall establish a shared secret with its peer using one of the following**
 5177 **mechanisms: [assignment: *list of key establishment mechanisms*].**

5178 **FTP_PRO.2.2**

5179 **The TSF shall authenticate [selection: *its peer, itself to its peer*] using one of the following**
 5180 **mechanisms: [assignment: *list of authentication mechanisms*] and according to the**
 5181 **following rules: [assignment: *list of rules for carrying out the authentication*].**

5182 **FTP_PRO.2.3**

5183 **The TSF shall use [assignment: *key derivation function*] to derive the following**
 5184 **cryptographic keys from a shared secret: [assignment: *list of cryptographic keys*].**

5185 **17.3.11 FTP_PRO.3 Trusted channel data protection**5186 **Component relationships**

5187	Hierarchical to:	No other components.
5188	Dependencies:	FTP_PRO.1 Trusted channel protocol
5189		FTP_PRO.2 Trusted channel establishment
5190		FCS_COP.1 Cryptographic operation.

5191 **FTP_PRO.3.1**

5192 **The TSF shall protect data in transit from unauthorised disclosure using one of the**
 5193 **following mechanisms: [assignment: *list of encryption mechanisms*].**

5194 **FTP_PRO.3.2**

5195 **The TSF shall protect data in transit from [selection: *modification, deletion, insertion,***
 5196 ***replay, [assignment: *other*]] using one of the following mechanisms: [assignment: *list of****
 5197 ***integrity protection mechanisms*].**

5198 **17.4 Trusted path (FTP_TRP)**5199 **17.4.1 Family behaviour**

5200 This family defines the requirements to establish and maintain trusted communication to or
 5201 from users and the TSF. A trusted path **can** be required for any security-relevant interaction.
 5202 Trusted path exchanges **can** be initiated by a user during an interaction with the TSF, or the TSF
 5203 **can** establish communication with the user via a trusted path.

5204 **17.4.2 Components leveling and description**

5205 Figure 91 shows the component leveling for this family.



5206

5207 **Figure 91 — FTP_TRP: Component leveling**

5208 FTP_TRP.1 Trusted path, requires that a trusted path between the TSF and a user be provided
 5209 for a set of events defined by a PP, PP-Module, functional package or ST author. The user and/or
 5210 the TSF **can** have the ability to initiate the trusted path.

5211 **17.4.3 Management of FTP_TRP.1**

5212 The following actions **could** be considered for the management functions in FMT:

- 5213 a) Configuring the actions that require trusted path, if supported.

5214 **17.4.4 Audit of FTP_TRP.1**

5215 The following actions should be auditable if FAU_GEN Security audit data generation is included
 5216 in the PP, PP-Module, functional package or ST:

- 5217 a) Minimal: Failures of the trusted path functions.
 5218 b) Minimal: Identification of the user associated with all trusted path failures, if
 5219 available.
 5220 c) Basic: All attempted uses of the trusted path functions.
 5221 d) Basic: Identification of the user associated with all trusted path invocations, if
 5222 available.

5223 **17.4.5 FTP_TRP.1 Trusted path**

5224 **Component relationships**

- 5225 Hierarchical to: No other components.
 5226 Dependencies: No dependencies.

5227 **FTP_TRP.1.1**

5228 **The TSF **shall** provide a communication path between itself and [selection: *remote, local*]
 5229 users that is logically distinct from other communication paths and provides assured
 5230 identification of its end points and protection of the communicated data from [selection:
 5231 *modification, disclosure, [assignment: other types of integrity or confidentiality violation]].***

5232 **FTP_TRP.1.2**

5233 **The TSF **shall** permit [selection: *the TSF, local users, remote users*] to initiate
 5234 communication via the trusted path.**

5235 **FTP_TRP.1.3**

5236 **The TSF **shall** require the use of the trusted path for [selection: *initial user*
 5237 *authentication, [assignment: other services for which trusted path is required]].***

5238
5239
5240

Annex A (informative)

Security functional requirements structure of the application notes

5241 **A.1 General information**

5242 This annex contains additional guidance for the families and components defined in this
5243 document, which **may** be required by users, developers, or evaluators to use the components.
5244 To facilitate finding the appropriate information, the presentation of the classes, families and
5245 components in this annex is similar to the presentation within the main clauses of this
5246 document.

5247 **A.2 Structure of the notes**

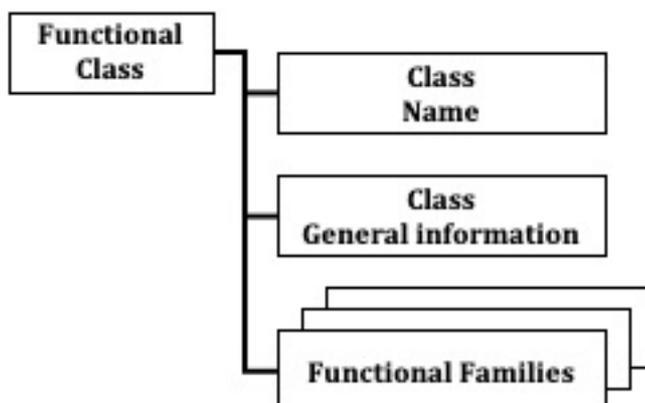
5248 **A.2.1 General**

5249 The content and presentation of the notes related to functional requirements in this document
5250 will be defined below.

5251 **A.2.2 Class structure**

5252 **A.2.2.1 General**

5253 Figure A.1 illustrates the functional class structure in this annex.



NOTE
A functional class could contain
multiple Functional Families

5254
5255

Figure A.1 — Functional class structure

5256 **A.2.2.2 Class name**

5257 This is the unique name of the class defined within the normative elements of this document.

5258 **A.2.2.3 Class introduction**

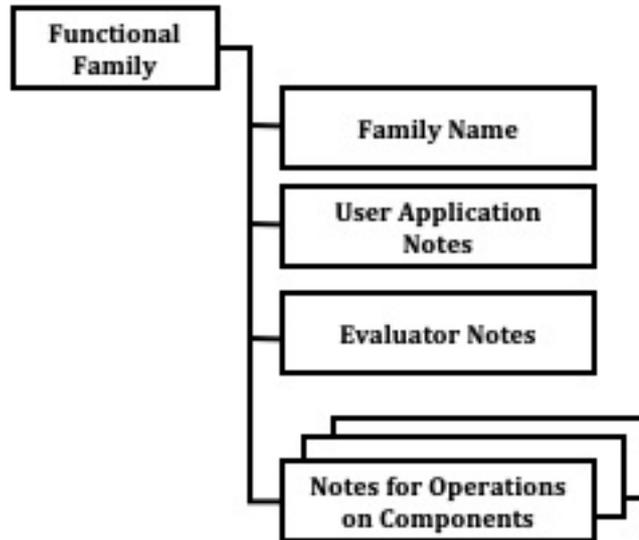
5259 The class introduction in this annex provides information about the use of the families and
5260 components of the class. This information is completed with the informative diagram that
5261 describes the organization of each class with the families in each class and the hierarchical
5262 relationship between components in each family.

5263 **A.2.3 Family structure**

5264 **A.2.3.1 General**

5265 Figure A.2 illustrates the functional family structure for application notes in diagrammatic form.

5266



5267 **Figure A.2 — Functional family structure for application notes**

5268 **A.2.3.2 Family name**

5269 This is the unique name of the family defined within the normative elements of this document.

5270 **A.2.3.3 User application notes**

5271 The user notes contain additional information that is of interest to potential users of the family,
 5272 that is PP, PP-Module, ST and functional package authors, and developers of TOEs incorporating
 5273 the functional components. The presentation is informative and might cover warnings about
 5274 limitations of use and areas where specific attention might be required when using the
 5275 components.

5276 **NOTE** In the annexes the term PP, PP-Module, functional package or ST author includes authors of documents
 5277 used to formulate a PP or ST, this includes PP-Modules and functional packages.

5278 **A.2.3.4 Evaluator notes**

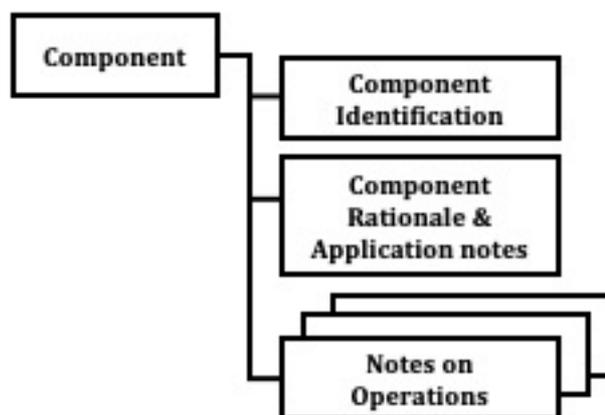
5279 The evaluator notes contain any information that is of interest to developers and evaluators of
 5280 TOEs that claim compliance with a component of the family. The presentation is informative
 5281 and **can** cover a variety of areas where specific attention might be needed when evaluating the
 5282 TOE. This **can** include clarifications of meaning and specification of the way to interpret
 5283 requirements, as well as caveats and warnings of specific interest to evaluators.

5284 These User Notes and Evaluator Notes subclauses are not mandatory and appear only if
 5285 appropriate.

5286 **A.2.4 Component structure**

5287 **A.2.4.1 General**

5288 Figure A.3 illustrates the functional component structure for the application notes.



5289

5290

Figure A.3 — Functional component structure

5291 **A.2.4.2 Component identification**

5292 This is the unique name of the component defined within the normative elements of this
5293 document.

5294 **A.2.4.3 Component rationale and application notes**

5295 Any specific information related to the component is found in the component rationale and
5296 application notes subclause.

5297 — The *rationale* contains the specifics of the rationale that refine the general statements on
5298 rationale for the specific level and is only be used if level specific amplification is required.

5299 — The *application notes* contain additional refinement in terms of narrative qualification as it
5300 pertains to a specific component. This refinement **may** pertain to user notes, and/or
5301 evaluator notes as described in A.2.3. This refinement **may** be used to explain the nature of
5302 the dependencies.

5303 The component rationale and application notes subclause is not mandatory and appears only if
5304 appropriate.

5305 **A.2.4.4 Notes on operations**

5306 This portion of each component contains advice relating to the permitted operations of the
5307 component.

5308 The permitted operations subclause is not mandatory and appears only if appropriate.

5309
5310
5311

Annex B
(informative)
Dependency tables for security functional components

5312 **B.1 Dependency tables**

5313 The tables from B.1 to B.11 show the hierarchical, direct, indirect, and optional dependencies
5314 among functional components.

5315 Each of the components that is a dependency of some functional component is allocated a
5316 column. Each functional component is allocated a row. The value in the table cell indicates
5317 whether the column label component is a hierarchical requirement (indicated by an "H").
5318 directly required (indicated by a cross "X"), indirectly required (indicated by a dash "-"), or
5319 optionally required (indicated by a "O") by the row label component. Sets of optional
5320 requirements are indicated by using a subscript group, e.g. O¹ and O².

5321 NOTE Depending upon the optional requirements chosen, some indirect dependencies are not
5322 applicable.

5323 If no character is presented, the component is not dependent upon another component.

5324 EXAMPLE

5325 An example of a component with optional dependencies is FDP_ETC.1 Export of user data without security attributes,
5326 which requires either FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control to be present.
5327 So, if FDP_ACC.1 Subset access control is present, FDP_IFC.1 Subset information flow control is not necessary and vice
5328 versa.

5329

5330

Table B.1 — Dependency table for Class FAU: Security audit

	FAU_GEN.1	FAU_SAA.1	FAU_SAA.3	FAU_SAR.1	FAU_STG.1	FAU_STG.2	FAU_STG.4	FIA_UID.1	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FPT_STM.1	FTP_ITC.1
FAU_ARP.1	-	X										-	
FAU_GEN.1												X	
FAU_GEN.2	X							X				-	
FAU_SAA.1	X											-	
FAU_SAA.2								X					
FAU_SAA.3													
FAU_SAA.4			H										
FAU_SAR.1	X											-	
FAU_SAR.2	-			X								-	
FAU_SAR.3	-			X								-	
FAU_SEL.1	X							-	X	-	-	-	
FAU_STG.1	X											-	X
FAU_STG.2	X											-	
FAU_STG.3	X					H						-	
FAU_STG.4	-					X						-	
FAU_STG.5	X					X	H					-	

5331

5332

5333

Table B.2 — Dependency table for Class FCO: Communication

	FIA_UID.1	FCO_NRR.1	FCO_NRO.1
FCO_NRO.1	X		
FCO_NRO.2	X		H
FCO_NRR.1	X		
FCO_NRR.2	X	H	

5334

5335 **Table B.3 — Dependency table for Class FCS: Cryptographic support**

	FCS_CKM.1	FCS_CKM.2	FCS_CKM.3	FCS_CKM.5	FCS_CKM.6	FCS_COP.1	FCS_RBG.1	FCS_RBG.2	FCS_RBG.3	FCS_RBG.4	FCS_RBG.5	FCS_RBG.6	FCS_RNG.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_ITC.1	FDP_ITC.2	FIA_UID.1	FMT_MSA.1	FMT_MSA.3	FMT_SMF.1	FMT_SMR.1	FPT_FLS.1	FPT_TST.1	FPT_TDC.1	FPT_ITC.1	FPT_TRP.1	
FCS_CKM.1	-	O ¹	X	O ¹	X	O ¹	O ²	-	-	O ²	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
FCS_CKM.2	O ¹	-	X	O ¹	-	-	-	-	-	-	-	-	-	-	-	-	-	O ¹	O ¹	-	-	-	-	-	-	-	-	-	-	
FCS_CKM.3	O ¹	-	-	O ¹	-	-	-	-	-	-	-	-	-	-	-	-	-	O ¹	O ¹	-	-	-	-	-	-	-	-	-	-	
FCS_CKM.5	-	O ¹	-	-	X	O ¹	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
FCS_CKM.6	O ¹	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	O ¹	O ¹	-	-	-	-	-	-	-	-	-	-	
FCS_COP.1	O ²	-	X	O ²	-	-	-	-	-	-	-	-	-	-	-	-	-	O ¹	O ¹	-	-	-	-	-	-	-	-	-	-	
FCS_RBG.1							-	O ¹	O ¹																X	X				
FCS_RBG.2							X	-	-																					
FCS_RBG.3							X	-	-																					
FCS_RBG.4							X	-	X																					
FCS_RBG.5							X	O ¹	O ¹																					
FCS_RBG.6							X	O ¹	O ¹																					
FCS_RNG.1																														

5336

Table B.4 — Dependency table for Class FDP: User data protection

	FDP_TRP.1	FTP_ITC.1	FPT_TDC.1	FPT_TST.1	FPT_FLS.1	FMT_SMR.1	FMT_SMF.1	FMT_MSA.3	FMT_MSA.1	FIA_UID.1	FDP_UTT.2	FDP_UTT.1	FDP_SDI.1	FDP_ROL.1	FDP_RIP.1	FDP_ITT.3	FDP_ITT.2	FDP_ITT.1	FDP_ITC.2	FDP_ITC.1	FDP_IFT.4	FDP_IFT.3	FDP_IFT.1	FDP_IFC.1	FDP_ACF.1	FDP_ACC.1	FCS_RNG.1	FCS_RBG.3	FCS_RBG.2	FCS_RBG.1	FCS_COP.1	FCS_CKM.6	FCS_CKM.5	FCS_CKM.3	FCS_CKM.1					
FDP_ACC.1																																								
FDP_ACC.2																																								
FDP_ACF.1																																								
FDP_DAU.1																																								
FDP_DAU.2																																								
FDP_ETC.1																																								
FDP_ETC.2																																								
FDP_IFC.1																																								
FDP_IFC.2																																								
FDP_IFT.1																																								
FDP_IFT.2																																								
FDP_IFT.3																																								
FDP_IFT.4																																								
FDP_IFT.5																																								
FDP_IFT.6																																								
FDP_IRC.1																																								

FTP_TRP.1	-
FTP_ITC.1	O ²
FPT_TDC.1	
FPT_TST.1	
FPT_FLS.1	
FMT_SMR.1	-
FMT_SMF.1	-
FMT_MSA.3	-
FMT_MSA.1	-
FIA_UID.1	-
FDP_UTT.2	H
FDP_UTT.1	O ²
FDP_SDL.1	
FDP_ROL.1	
FDP_RIP.1	
FDP_ITT.3	
FDP_ITT.2	
FDP_ITT.1	
FDP_ITC.2	
FDP_ITC.1	
FDP_IFF.4	
FDP_IFF.3	
FDP_IFF.1	-
FDP_IFC.1	O ¹
FDP_ACF.1	-
FDP_ACC.1	O ¹
FCS_RNG.1	
FCS_RBG.3	
FCS_RBG.2	
FCS_RBG.1	
FCS_COP.1	
FCS_CKM.6	
FCS_CKM.5	
FCS_CKM.3	
FCS_CKM.1	
FDP_UTT.3	

5339

5340

Table B.5 — Dependency table for Class FIA: Identification and authentication

	FIA_ATD.1	FIA_UAU.1	FIA_UID.1
FIA_AFL.1		X	-
FIA_API.1			
FIA_ATD.1			
FIA_SOS.1			
FIA_SOS.2			
FIA_UAU.1			X
FIA_UAU.2		H	X
FIA_UAU.3			
FIA_UAU.4			
FIA_UAU.5			
FIA_UAU.6			
FIA_UAU.7		X	-
FIA_UID.1			
FIA_UID.2			H
FIA_USB.1	X		

5341

5342

5343

Table B.6 — Dependency table for Class FMT: Security management

	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FIA_UID.1	FMT_LIM.1	FMT_LIM.2	FMT_MSA.1	FMT_MSA.3	FMT_MTD.1	FMT_SMF.1	FMT_SMR.1	FPT_STM.1
FMT_LIM.1						-	X						
FMT_LIM.2						X	-						
FMT_MOF.1					-						X	X	
FMT_MSA.1	O ¹	-	O ¹	-	-			-	-		X	X	
FMT_MSA.2	O ¹	-	O ¹	-	-			X	-		-	X	
FMT_MSA.3	-	-	-	-	-			X	-		-	X	
FMT_MSA.4	O ¹	-	O ¹	-	-			-	-		-	-	
FMT_MTD.1					-						X	X	
FMT_MTD.2					-					X	-	X	
FMT_MTD.3					-					X	-	-	
FMT_REV.1					-							X	
FMT_SAE.1					-							X	X
FMT_SMF.1													
FMT_SMR.1					X								
FMT_SMR.2					X							H	
FMT_SMR.3					-							X	

5344

5345

5346

Table B.7 — Dependency table for Class FPR: Privacy

	FIA_UID.1	FPR_ANO.1	FPR_PSE.1	FPR_UNO.1
FPR_ANO.1				
FPR_ANO.2		H		
FPR_PSE.1				
FPR_PSE.2	X		H	
FPR_PSE.3			H	
FPR_UNL.1				
FPR_UNO.1				
FPR_UNO.2				H
FPR_UNO.3				X
FPR_UNO.4				

5347

5348

5349

Table B.8 — Dependency table for Class FPT: Protection of the TSF

	AGD_OPE.1	ADV_FSP.1	FIA_UID.1	FMT_LIM.1	FMT_LIM.2	FMT_SMF.1	FMT_SMR.1	FPT_ITI.1	FPT_ITT.1	FPT_PHP.1	FPT_RCV.1	FPT_RCV.2	FPT_SSP.1	FPT_STM.1
FPT_EMS.1														
FPT_FLS.1														
FPT_INI.1														
FPT_ITA.1														
FPT_ITC.1														
FPT_ITI.1														
FPT_ITI.2								H						
FPT_ITT.1														
FPT_ITT.2									H					
FPT_ITT.3									X					
FPT_PHP.1														
FPT_PHP.2				X	-					H				
FPT_PHP.3														
FPT_RCV.1	X	-												
FPT_RCV.2	X	-									H			
FPT_RCV.3	X	-										H		
FPT_RCV.4														
FPT_RPL.1														
FPT_SSP.1									X					
FPT_SSP.2									X				H	
FPT_STM.1														
FPT_STM.2							X							X
FPT_TDC.1														
FPT_TEE.1														
FPT_TRC.1									X					
FPT_TST.1														

5350

5351

NOTE The AGD and ADV classes and their dependencies are described in ISO/IEC 15408-3

5352 **Table B.9 — Dependency table for Class FRU: Resource utilization**

	FPT_FLS.1	FRU_FLT.1	FRU_PRS.1	FRU_RSA.1
FRU_FLT.1	X			
FRU_FLT.2	X	H		
FRU_PRS.1				
FRU_PRS.2			H	
FRU_RSA.1				
FRU_RSA.2				H

5353
5354
5355 **Table B.10 — Dependency table for Class FTA: TOE access**

	FIA_UAU.1	FIA_UID.1	FMT_SMR.1	FTA_MCS.1
FTA_LSA.1				
FTA_MCS.1		X		
FTA_MCS.2		X		H
FTA_SSL.1	X	-		
FTA_SSL.2	X	-		
FTA_SSL.3			X	
FTA_SSL.4				
FTA_TAB.1				
FTA_TAH.1				
FTA_TSE.1				

5356

Table B.11 — Dependency table for Class FTP: Trusted Path/channels

	FTP_PRO.3	FTP_PRO.2	FTP_PRO.1	FTP_TRP.1	FTP_ITC.1	FPT_TDC.1	FPT_TST.1	FPT_FLS.1	FMT_SMR.1	FMT_SMF.1	FMT_MSA.3	FMT_MSA.1	FIA_UID.1	FDP_ITC.2	FDP_ITC.1	FDP_IFF.1	FDP_IFC.1	FDP_ACF.1	FDP_ACC.1	FCS_RNG.1	FCS_RBG.3	FCS_RBG.2	FCS_RBG.1	FCS_COP.1	FCS_CKM.6	FCS_CKM.5	FCS_CKM.3	FCS_CKM.2	FCS_CKM.1
FTP_ITC.1																													
FTP_PRO.1	X	X																											
FTP_PRO.2			X																					X		X		O ¹	O ¹
FTP_PRO.3		X	X																					X					
FTP_TRP.1																													

Annex C (normative)

Class FAU: Security audit - application notes

5360
5361
5362
5363

5364 C.1 General information

5365 C.1.1 General information about audit requirements

5366 ISO/IEC 15408 audit families allow PP, PP-Module, functional package or /ST authors the ability
5367 to define requirements for monitoring user activities and, in some cases, detecting real,
5368 possible, or imminent violations of the enforcement of the SFRs. The TOE's security audit
5369 functions are defined to help monitor security-relevant events, and act as a deterrent against
5370 security violations. The requirements of the audit families refer to functions that include audit
5371 data protection, record format, and event selection, as well as analysis tools, violation alarms,
5372 and real-time analysis. The audit records **may** be presented in human-readable format either
5373 directly or indirectly or both.

5374 EXAMPLE 1

5375 An example of direct presentation is storing the audit records in human-readable format

5376 An example of indirect presentation is by using audit reduction tools.

5377 While developing the security audit requirements, the PP, PP-Module, functional package or ST
5378 author **should** take note of the inter-relationships among the audit families and components.
5379 The potential exists to specify a set of audit requirements that comply with the
5380 family/component dependencies lists, while at the same time resulting in a deficient audit
5381 function.

5382 EXAMPLE 2

5383 An audit function that requires all security relevant events to be audited but without the selectivity to control them
5384 on any reasonable basis such as individual user or object.

5385 C.1.2 Audit requirements in a distributed environment

5386 The implementation of audit requirements for networks and other large systems **can** differ
5387 significantly from those needed for stand-alone systems. Larger, more complex, and active
5388 systems require more thought concerning which audit data to collect and how this **can** be
5389 managed, due to lowered feasibility of interpreting (or even storing) what gets collected. The
5390 traditional notion of a time-ordered list, set of records or "trail" of audited events is not always
5391 applicable in a global asynchronous network with many arbitrary events occurring at once.

5392 Also, different hosts and servers on a distributed TOE **can** have differing naming policies and
5393 values. Further, the use of symbolic names for audit review requires a net-wide convention to
5394 avoid redundancies and "name clashes."

5395 A multi-object audit repository, portions of which are accessible by a potentially wide variety of
5396 authorized users, are usually required if audit repositories are to serve a useful function in
5397 distributed systems.

5398 Finally, misuse of authority by authorized users **can** be addressed by systematically avoiding
5399 local storage of audit data pertaining to administrator actions.

5400 C.2 Security audit automatic response (FAU_ARP)

5401 C.2.1 User application notes

5402 The Security audit automatic response family describes requirements for the handling of audit
 5403 events. The requirement **could** include requirements for alarms or TSF action (automatic
 5404 response).

5405 EXAMPLE

5406 the TSF **could** include the generation of real time alarms, termination of the offending process, disabling of a service,
 5407 or disconnection or invalidation of a user account.

5408 An audit event is defined to be an “potential security violation” if so indicated by the Security
 5409 audit analysis (FAU_SAA) components.

5410 **C.2.2 FAU_ARP.1 Security alarms**

5411 **C.2.2.1 Component rationale and application notes**

5412 One or more actions **should** be taken for follow up action in the event of an alarm.

5413 These actions **could** include informing the authorized user of the alarm, presenting the
 5414 authorized user with a set of possible containment actions, or options for the authorized user to
 5415 take corrective actions.

5416 The timing of the actions **should** be carefully considered by the PP, PP-Module, functional
 5417 package or /ST author.

5418 **C.2.2.2 Operations**

5419 In FAU_ARP.1.1, the PP, PP-Module, functional package or ST author specifies the actions to be
 5420 taken in case of a potential security violation.

5421 EXAMPLE

5422 An example of such a list is: “inform the authorized user, disable the subject that created the potential security
 5423 violation.”

5424 The list **may** also specify that the action to be taken **can** be specified by an authorized user.

5425 **C.3 Security audit data generation (FAU_GEN)**

5426 **C.3.1 User application notes**

5427 The Security audit data generation family includes requirements to specify the audit events that
 5428 **shall** be generated by the TSF for security-relevant events.

5429 This family is presented in a manner that avoids a dependency on all components requiring
 5430 audit support. Each component has an audit subclause developed in which the events to be
 5431 audited for that functional area are listed. When the PP, PP-Module, functional package or /ST is
 5432 written, the items in the audit area are used to complete the variable in these components.
 5433 Thus, the specification of what **could** be audited for a functional area is localized in that
 5434 functional area.

5435 The list of auditable events is entirely dependent on the other functional families within the PP,
 5436 PP-Module, functional package or ST./ST. Each family definition **should** therefore include a list
 5437 of its family-specific auditable events. Each auditable event in the list of auditable events
 5438 specified in the functional family **should** correspond to one of the levels of audit event
 5439 generation specified in this family (i.e. minimal, basic, detailed). This provides the PP, PP-
 5440 Module, functional package or ST author with information necessary to ensure that all
 5441 appropriate auditable events are specified in the PP, PP-Module, functional package or ST./ST.
 5442 The following example shows how auditable events are to be specified in appropriate functional
 5443 families:

5444 EXAMPLE 1

5445 “The following actions **should** be auditable if Security audit data generation (FAU_GEN) is included in the PP, PP-
 5446 Module, functional package or ST:

5447 a) Minimal: Successful use of the user security attribute administration functions.

- 5448 b) Basic: All attempted uses of the user security attribute administration functions.
5449 c) Basic: Identification of which user security attributes have been modified.
5450 d) Detailed: With the exception of specific sensitive attribute data items, the new values of the attributes
5451 should be captured.”

5452 NOTE Sensitive attribute data items include passwords and cryptographic keys.

5453 For each functional component that is chosen, the auditable events that are indicated in that
5454 component, at and below the level indicated in Security audit data generation (FAU_GEN)
5455 **should** be auditable. So, in the previous example “Basic” would be selected in Security audit data
5456 generation (FAU_GEN), the auditable events mentioned in a), b) and c) **should** be auditable.

5457 Observe that the categorization of auditable events (minimal, basic, detailed) is hierarchical in
5458 that order.

5459 This means that

- 5460 • When Minimal Audit Generation is desired, all auditable events identified as being
5461 Minimal **should** be included in the PP, PP-Module, functional package or ST through the
5462 use of the appropriate assignment operation.
- 5463 • When Basic Audit Generation is desired, all auditable events identified as being either
5464 Minimal or Basic, **should** also be included in the PP, PP-Module, functional package or ST
5465 through the use of the appropriate assignment operation, except when the higher-level
5466 event simply provides more detail than the lower level event.
- 5467 • When Detailed Audit Generation is desired, all identified auditable events (Minimal,
5468 Basic, and Detailed) **should** be included in the PP, PP-Module, functional package or ST.

5469 A PP, PP-Module, functional package or ST author **may** decide to include other auditable events
5470 beyond those required for a given audit level.

5471 EXAMPLE 2

5472 For example, the PP, PP-Module, functional package or ST **may** claim only minimal audit capabilities while including
5473 most of the basic capabilities because the few excluded capabilities conflict with other PP, PP-Module, functional
5474 package or ST constraints (perhaps because they require the collection of unavailable data).

5475 The functionality that creates the auditable event **should** be specified in the PP or ST as a
5476 functional requirement.

5477 EXAMPLE 3

5478 The following are examples of the types of the events that **can** be defined as auditable within each PP, PP-Module,
5479 functional package or ST functional component:

- 5480 a) Introduction of objects within the control of the TSF into a subject's address space;
- 5481 b) Deletion of objects;
- 5482 c) Distribution or revocation of access rights or capabilities;
- 5483 d) Changes to subject or object security attributes;
- 5484 e) Policy checks performed by the TSF as a result of a request by a subject;
- 5485 f) The use of access rights to bypass a policy check;
- 5486 g) Use of Identification and Authentication functions;
- 5487 h) Actions taken by an operator, and/or authorized user (such as. suppression of a TSF protection mechanism
5488 as human-readable labels);
- 5489 i) Import/export of data from/to removable media (such as printed output, tapes, USB sticks).

5490 C.3.1.1 Evaluator notes

5491 FAU_GEN.1.1 has a dependency on FPT_STM.1 Reliable time stamps. If correctness of time is not
5492 an issue for this TOE, elimination of this dependency **could** be justified by the PP, PP-Module,
5493 functional package or ST author.

5494

5495 C.3.2 FAU_GEN.1 Audit data generation

5496 C.3.2.1 Component rationale and application notes

5497 This component defines requirements to identify the auditable events for which audit records
5498 **should** be generated, and the information to be provided in the audit records.

5499 FAU_GEN.1 Audit data generation by itself might be used when the SFRs do not require that
5500 individual user identities be associated with audit events. This **could** be appropriate when the
5501 PP, PP-Module, functional package or ST also contains privacy requirements. If the user identity
5502 must be incorporated FAU_GEN.2 User identity association **could** be used in addition to
5503 FAU_GEN.1.

5504 If the subject is a user, the user identity **may** be recorded as the subject identity. The identity of
5505 the user **may** not yet have been verified if User authentication (FIA_UAU) has not been applied.
5506 Therefore, in the instance of an invalid login the claimed user identity **should** be recorded. It
5507 **should** also be considered whether to indicate when a recorded identity has not been
5508 authenticated.

5509 C.3.2.2 Operations

5510 In FAU_GEN.1.1, the PP, PP-Module, functional package or ST author **should** select the level of
5511 auditable events called out in the audit subclause of other functional components included in
5512 the PP, PP-Module, functional package or ST. This level is one of the following: “minimum”,
5513 “basic”, “detailed” or “not specified”.

5514 In FAU_GEN.1.1, the PP, PP-Module, functional package or ST author **should** assign a list of other
5515 specifically defined auditable events to be included in the list of auditable events. The
5516 assignment **may** comprise none, or events that **could** be auditable events of a functional
5517 requirement that are of a higher audit level than requested in b), as well as the events
5518 generated through the use of a specified Application Programming Interface (API).

5519 In FAU_GEN.1.2, the PP, PP-Module, functional package or ST author **should** assign, for each of
5520 the auditable events included in the PP, PP-Module, functional package or ST, either a list of
5521 other audit relevant information to be included in audit events records or none.

5522 C.3.3 FAU_GEN.2 User identity association

5523 C.3.3.1 Component rationale and application notes

5524 This component addresses the requirement of accountability of auditable events at the level of
5525 individual user identity. This component **should** be used in addition to FAU_GEN.1 Audit data
5526 generation.

5527 There is a potential conflict between the audit and privacy requirements. For audit purposes, it
5528 **may** be desirable to know who performed an action. A user **may** want to keep his/her actions to
5529 himself/herself and not be identified by other persons such as a site with job offers. Or it might
5530 be required in the Organizational Security Policy that the identity of the users must be
5531 protected. In those cases, the objectives for audit and privacy **could** contradict each other.
5532 Therefore, if this requirement is selected and privacy is important, inclusion of the component
5533 user pseudonymity might be considered. Requirements on determining the real user name
5534 based on its pseudonym are specified in the privacy class.

5535 If the identity of the user has not yet been verified through authentication, in the instance of an
5536 invalid login the claimed user identity **should** be recorded. It **should** be considered to indicate
5537 when a recorded identity has not been authenticated.

5538 C.4 Security audit analysis (FAU_SAA)

5539 C.4.1 User application notes

5540 This family defines requirements for automated means that analyze system activity and audit
 5541 data looking for possible or real security violations. This analysis **may** work in support of
 5542 intrusion detection, or automatic response to a potential security violation.

5543 The action to be performed by the TSF on detection of a potential violation is defined in Security
 5544 audit automatic response (FAU_ARP) components.

5545 For real-time analysis, audit data **could** be transformed into a useful format for automated
 5546 treatment, but into a different useful format for delivery to authorized users for review.

5547 **C.4.2 FAU_SAA.1 Potential violation analysis**

5548 **C.4.2.1 Component rationale and application notes**

5549 This component is used to specify the set of auditable events whose occurrence or accumulated
 5550 occurrence held to indicate a potential violation of the enforcement of the SFRs, and any rules to
 5551 be used to perform the violation analysis.

5552 **C.4.2.2 Operations**

5553 In FAU_SAA.1.2, the PP, PP-Module, functional package or ST author **should** identify the subset
 5554 of defined auditable events whose occurrence or accumulated occurrence need to be detected
 5555 as an indication of a potential violation of the enforcement of the SFRs.

5556 In FAU_SAA.1.2, the PP, PP-Module, functional package or ST author **should** specify any other
 5557 rules that the TSF **should** use in its analysis of the audit trail. Those rules **could** include specific
 5558 requirements to express the needs for the events to occur in a certain period of time. If there
 5559 are no additional rules that the TSF **should** use in the analysis of the audit trail, this assignment
 5560 **can** be completed with “none”.

5561 EXAMPLE

5562 Period of time: period of the day, duration

5563 **C.4.3 FAU_SAA.2 Profile based anomaly detection**

5564 **C.4.3.1 Component rationale and application notes**

5565 A *profile* is a structure that characterizes the behaviour of users and/or subjects; it represents
 5566 how the users/subjects interact with the TSF in a variety of ways. Patterns of usage are
 5567 established with respect to the various types of activity the users/subjects engage in. The ways
 5568 in which the various types of activity are recorded in the profile are referred to as *profile*
 5569 *metrics*.

5570 EXAMPLE

5571 Patterns of usage: patterns in exceptions raised, patterns in resource utilization (when, which, how), patterns in
 5572 actions performed.

5573 Profile metrics: resource measures, event counters, timers

5574 Each profile represents the expected patterns of usage performed by members of the *profile*
 5575 *target group*. This pattern **may** be based on past use (historical patterns) or on normal use for
 5576 users of similar target groups (expected behaviour). A profile target group refers to one or more
 5577 users who interact with the TSF. The activity of each member of the profile group is used by the
 5578 analysis tool in establishing the usage patterns represented in the profile. The following are
 5579 some examples of profile target groups:

- 5580 a) **Single user account:** one profile per user;
- 5581 b) **Group ID or Group Account:** one profile for all users who possess the same group
 5582 ID or operate using the same group account;
- 5583 c) **Operating Role:** one profile for all users sharing a given operating role;
- 5584 d) **System:** one profile for all users of a system.

5585 Each member of a profile target group is assigned an individual *suspicion rating* that represents
 5586 how closely that member's new activity corresponds to the established patterns of usage
 5587 represented in the group profile.

5588 The sophistication of the anomaly detection tool will largely be determined by the number of
 5589 target profile groups required by the PP, PP-Module, functional package or ST and the
 5590 complexity of the required profile metrics.

5591 The PP, PP-Module, functional package or ST author **should** enumerate specifically what activity
 5592 **should** be monitored and/or analysed by the TSF. The PP, PP-Module, functional package or ST
 5593 author **should** also identify specifically what information pertaining to the activity is necessary
 5594 to construct the usage profiles.

5595 FAU_SAA.2 Profile based anomaly detection requires that the TSF maintain profiles of system
 5596 usage. The word maintain implies that the anomaly detector is actively updating the usage
 5597 profile based on new activity performed by the profile target members. It is important here that
 5598 the metrics for representing user activity are defined by the PP, PP-Module, functional package
 5599 or ST author.

5600 EXAMPLE 2

5601 For example, there **may** be a thousand different actions an individual **may** be capable of performing, but the anomaly
 5602 detector **may** choose to monitor a subset of that activity.

5603 Anomalous activity gets integrated into the profile just like non-anomalous activity (assuming
 5604 the tool is monitoring those actions). Things that **may** have appeared anomalous four months
 5605 ago, might over time become the norm (and vice-versa) as the user's work duties change. The
 5606 TSF wouldn't be able to capture this notion if it filtered out anomalous activity from the profile
 5607 updating algorithms.

5608 Administrative notification **should** be provided such that the authorized user understands the
 5609 significance of the suspicion rating.

5610 The PP, PP-Module, functional package or ST author **should** define how to interpret suspicion
 5611 ratings and the conditions under which anomalous activity is indicated to the Security audit
 5612 automatic response (FAU_ARP) mechanism.

5613 **C.4.3.2 Operations**

5614 In FAU_SAA.2.1, the PP, PP-Module, functional package or ST author **should** specify the profile
 5615 target group. A single PP, PP-Module, functional package or ST **may** include multiple profile
 5616 target groups.

5617 In FAU_SAA.2.3, the PP, PP-Module, functional package or ST author **should** specify conditions
 5618 under which anomalous activity is reported by the TSF. Conditions **may** include the suspicion
 5619 rating reaching a certain value, or be based on the type of anomalous activity observed.

5620 **C.4.4 FAU_SAA.3 Simple attack heuristics**

5621 **C.4.4.1 Component rationale and application notes**

5622 In practice, it is at best rare when an analysis tool **can** detect with certainty when a security
 5623 violation is imminent. However, there do exist some system events that are so significant that
 5624 they are always worthy of independent review.

5625 EXAMPLE 1

5626 Example of such events include the deletion of a key TSF security data file (such as the password file) or activity such
 5627 as a remote user attempting to gain administrative privilege.

5628 These events are referred to as signature events in that their occurrence in isolation from the
 5629 rest of the system activity are indicative of intrusive activity.

5630 The complexity of a given tool will depend greatly on the assignments defined by the PP, PP-
 5631 Module, functional package or ST author in identifying the base set of *signature events*.

5632 The PP, PP-Module, functional package or ST author **should** enumerate specifically what events
 5633 **should** be monitored by the TSF in order to perform the analysis. The PP, PP-Module, functional
 5634 package or ST author **should** identify specifically what information pertaining to the event is
 5635 necessary to determine if the event maps to a signature event.

5636 Administrative notification **should** be provided such that the authorized user understands the
 5637 significance of the event and the appropriate possible responses.

5638 An effort was made in the specification of these requirements to avoid a dependency on audit
 5639 data as the sole input for monitoring system activity. This was done in recognition of the
 5640 existence of previously developed intrusion detection tools that do not perform their analyses
 5641 of system activity solely through the use of audit data.

5642 EXAMPLE 2

5643 Examples of other input data include network datagrams, resource/accounting data, or combinations of various
 5644 system data.

5645 The elements of FAU_SAA.3 Simple attack heuristics do not require that the TSF implementing
 5646 the immediate attack heuristics be the same TSF whose activity is being monitored. Thus, one
 5647 **can** develop an intrusion detection component that operates independently of the system
 5648 whose system activity is being analyzed.

5649 C.4.4.2 Operations

5650 In FAU_SAA.3.1, the PP, PP-Module, functional package or ST author **should** identify a base
 5651 subset of system events whose occurrence, in isolation from all other system activity, **may**
 5652 indicate a violation of the enforcement of the SFRs. These include events that by themselves
 5653 indicate a clear violation to the enforcement of the SFRs, or whose occurrence is so significant
 5654 that they warrant actions.

5655 In FAU_SAA.3.2, the PP, PP-Module, functional package or ST author **should** specify the
 5656 information used to determine system activity. This information is the input data used by the
 5657 analysis tool to determine the system activity that has occurred on the TOE. This data **may**
 5658 include audit data, combinations of audit data with other system data, or **may** consist of data
 5659 other than the audit data. The PP, PP-Module, functional package or ST author **should** define
 5660 precisely what system events and event attributes are being monitored within the input data.

5661 C.4.5 FAU_SAA.4 Complex attack heuristics

5662 C.4.5.1 Component rationale and application notes

5663 In practice, it is at best rare when an analysis tool **can** detect with certainty when a security
 5664 violation is imminent. However, there do exist some system events that are so significant they
 5665 are always worthy of independent review.

5666 EXAMPLE 1

5667 Example of such events include the deletion of a key TSF security data file (such as the password file) or activity such
 5668 as a remote user attempting to gain administrative privilege.

5669 These events are referred to as signature events in that their occurrence in isolation from the
 5670 rest of the system activity are indicative of intrusive activity. Event sequences are an ordered
 5671 set of signature events that might indicate intrusive activity.

5672 The complexity of a given tool will depend greatly on the assignments defined by the PP, PP-
 5673 Module, functional package or ST author in identifying the base set of signature events and
 5674 event sequences.

5675 The PP, PP-Module, functional package or ST author **should** enumerate specifically what events
 5676 **should** be monitored by the TSF in order to perform the analysis. The PP, PP-Module, functional
 5677 package or ST author **should** identify specifically what information pertaining to the event is
 5678 necessary to determine if the event maps to a signature event.

5679 Administrative notification **should** be provided such that the authorized user understands the
5680 significance of the event and the appropriate possible responses.

5681 An effort was made in the specification of these requirements to avoid a dependency on audit
5682 data as the sole input for monitoring system activity. This was done in recognition of the
5683 existence of previously developed intrusion detection tools that do not perform their analyses
5684 of system activity solely through the use of audit data.

5685 EXAMPLE 2

5686 Examples of other input data include network datagrams, resource/accounting data, or combinations of various
5687 system data.

5688 Levelling, therefore, requires the PP, PP-Module, functional package or ST author to specify the
5689 type of input data used to monitor system activity.

5690 The elements of FAU_SAA.4 Complex attack heuristics do not require that the TSF implementing
5691 the complex attack heuristics be the same TSF whose activity is being monitored. Thus, one **can**
5692 develop an intrusion detection component that operates independently of the system whose
5693 system activity is being analyzed.

5694 **C.4.5.2 Operations**

5695 In FAU_SAA.4.1, the PP, PP-Module, functional package or ST author **should** identify a base set of
5696 lists of sequences of system events whose occurrence are representative of known penetration
5697 scenarios. These event sequences represent known penetration scenarios. Each event
5698 represented in the sequence **should** map to a monitored system event, such that as the system
5699 events are performed, they are bound (mapped) to the known penetration event sequences.

5700 In FAU_SAA.4.1, the PP, PP-Module, functional package or ST author **should** identify a base
5701 subset of system events whose occurrence, in isolation from all other system activity, **may**
5702 indicate a violation of the enforcement of the SFRs. These include events that by themselves
5703 indicate a clear violation to the SFRs, or whose occurrence is so significant they warrant action.

5704 In FAU_SAA.4.2, the PP, PP-Module, functional package or ST author **should** specify the
5705 information used to determine system activity. This information is the input data used by the
5706 analysis tool to determine the system activity that has occurred on the TOE. This data **may**
5707 include audit data, combinations of audit data with other system data, or **may** consist of data
5708 other than the audit data. The PP, PP-Module, functional package or ST author **should** define
5709 precisely what system events and event attributes are being monitored within the input data.

5710 **C.5 Security audit review (FAU_SAR)**

5711 **C.5.1 User application notes**

5712 The Security audit review family defines requirements related to review of the audit
5713 information.

5714 These functions **should** allow pre-storage or post-storage audit selection.

5715 EXAMPLE

5716 An example of requirement related to review of the audit information is the ability to selectively review:

- 5717 — the actions of one or more users (such as. identification, authentication, TOE entry, and access control
5718 actions);
- 5719 — the actions performed on a specific object or TOE resource;
- 5720 — all of a specified set of audited exceptions; or
- 5721 — actions associated with a specific SFR attribute

5722 The distinction between audit reviews is based on functionality. Audit review (only)
5723 encompasses the ability to view audit data. Selectable review is more sophisticated and
5724 requires the ability to select subsets of audit data based on a single criterion or multiple criteria
5725 with logical (i.e. and/or) relations and order the audit data before it is reviewed.

5726 **C.5.2 FAU_SAR.1 Audit review**

5727 **C.5.2.1 Component rationale and application notes**

5728 This component provides authorized users the capability to obtain and interpret the
5729 information. In the case of human users this information needs to be in a human
5730 understandable presentation. In the case of external IT entities, the information needs to be
5731 unambiguously represented in an electronic fashion.

5732 This component is also used to specify that users and/or authorized users **can** read the audit
5733 records. These audit records will be provided in a manner appropriate to the user. There are
5734 different types of users (human users, machine users) that might have different needs.

5735 The content of the audit records that **can** be viewed **can** be specified.

5736 **C.5.2.2 Operations**

5737 In FAU_SAR.1.1, the PP, PP-Module, functional package or ST author **should** specify the
5738 authorized users that **can** use this capability. If appropriate the PP, PP-Module, functional
5739 package or ST author **may** include security roles (see FMT_SMR.1 Security roles).

5740 In FAU_SAR.1.1, the PP, PP-Module, functional package or ST author **should** specify the type of
5741 information the specified user is permitted to obtain from the audit records.

5742 EXAMPLE

5743 Examples are “all”, “subject identity”, “all information belonging to audit records referencing this user”.

5744 When employing the SFR, FAU_SAR.1, it is not necessary to repeat, in full detail, the list of audit
5745 information first specified in FAU_GEN.1. Use of terms such as “all” or “all audit information”
5746 assist in eliminating ambiguity and the further need for comparative analysis between the two
5747 security requirements.

5748 **C.5.3 FAU_SAR.2 Restricted audit review**

5749 **C.5.3.1 Component rationale and application notes**

5750 This component specifies that any users not identified in FAU_SAR.1 Audit review will not be
5751 able to read the audit records.

5752 **C.5.4 FAU_SAR.3 Selectable audit review**

5753 **C.5.4.1 Component rationale and application notes**

5754 This component is used to specify that it **should** be possible to perform selection of the audit
5755 data to be reviewed. If based on multiple criteria, those criteria **should** be related together with
5756 logical (i.e. “and” or “or”) relations, and the tools **should** provide the ability to manipulate audit
5757 data

5758 EXAMPLE

5759 Means of manipulating audit data include sorting and filtering.

5760 **C.5.4.2 Operations**

5761 In FAU_SAR.3.1, the PP, PP-Module, functional package or ST author **should** specify whether
5762 capabilities to select and/or order audit data is required from the TSF.

5763 In FAU_SAR.3.1, the PP, PP-Module, functional package or ST author **should** assign the criteria,
5764 possibly with logical relations, to be used to select the audit data for review. The logical
5765 relations are intended to specify whether the operation **can** be on an individual attribute or a
5766 collection of attributes.

5767 EXAMPLE

5768 An example of this assignment could be: “application, user account and/or location”.

5769 In this case, the operation **could** be specified using any combination of the three attributes:
5770 application, user account and location.

5771 **C.6 Security audit event selection (FAU_SEL)**

5772 **C.6.1 User application notes**

5773 The Security audit event selection family provides requirements related to the capabilities of
5774 identifying which of the possible auditable events are to be audited. The auditable events are
5775 defined in the Security audit data generation (FAU_GEN) family, but those events **should** be
5776 defined as being selectable in this component to be audited.

5777 This family ensures that it is possible to keep the audit trail from becoming so large that it
5778 becomes useless, by defining the appropriate granularity of the selected security audit events.

5779 **C.6.2 FAU_SEL.1 Selective audit**

5780 **C.6.2.1 Component rationale and application notes**

5781 This component defines the selection criteria used, and the resulting audited subsets of the set
5782 of all auditable events, based on user attributes, subject attributes, object attributes, or event
5783 types.

5784 The existence of individual user identities is not assumed for this component. This allows for
5785 TOEs such as routers that **may** not support the notion of users.

5786 For a distributed environment, the host identity **could** be used as a selection criterion for events
5787 to be audited.

5788 The management function FMT_MTD.1 Management of TSF data will handle the rights of
5789 authorized users to query or modify the selections.

5790 **C.6.2.2 Operations**

5791 In FAU_SEL.1.1, the PP, PP-Module, functional package or ST author **should** select whether the
5792 security attributes upon which audit selectivity is based, is related to object identity, user
5793 identity, subject identity, host identity, or event type.

5794 In FAU_SEL.1.1, the PP, PP-Module, functional package or ST author **should** specify any
5795 additional attributes upon which audit selectivity is based. If there are no additional rules upon
5796 which audit selectivity is based, this assignment **can** be completed with “none”.

5797 **C.7 Security audit data storage (FAU_STG)**

5798 **C.7.1 User application notes**

5799 The Security audit data storage family describes requirements for storing audit data for later
5800 use, including requirements controlling the loss of audit information due to TOE failure, attack
5801 and/or exhaustion of storage space.

5802 **C.7.2 FAU_STG.1 Audit data storage location**

5803 **C.7.2.1 Component rationale and application notes**

5804 In a distributed environment, as the location of the audit trail is in the TSF, but not necessarily
5805 co-located with the function generating the audit data, the PP, PP-Module, functional package or
5806 ST author could request authentication of the originator of the audit record, or non-repudiation
5807 of the origin of the record prior to storing this record in the audit trail.

5808 The TSF will protect the stored audit records in the audit trail from unauthorised deletion and
5809 modification. It is noted that in some TOEs the auditor (role) might not be authorized to delete
5810 the audit records for a certain period of time.

5811 FAU_STG.1.1 is dependent upon FTP_ITC.1 Inter-TSF trusted channel, if “transmit the generated
5812 audit data to an external IT entity using a trusted channel according to FTP_ITC” is not selected
5813 then the PP, PP-Module, functional package or ST author can satisfy the dependency by
5814 providing the rationale explaining why it was not selected.

5815 **C.7.2.2 Operations**

5816 In FAU_STG.1.1 the PP, PP-Module, functional package or ST author **should** select where the
5817 audit data is stored. Audit data may be stored on the TOE itself, be transmitted to an external
5818 entity using a trusted channel, or other storage options can be specified in the assignment.

5819 If additional or alternative storage locations for audit data need to be specified by the PP, PP-
5820 Module, functional package or ST author then this requirement can be specified in FAU_STG.1.1
5821 using the assignment found within the selection.

5822 **C.7.3 FAU_STG.2 Protected audit data storage**

5823 **C.7.3.1 Component rationale and application notes**

5824 In a distributed environment, as the location of the audit trail is in the TSF, but not necessarily
5825 co-located with the function generating the audit data, the PP, PP-Module, functional package or
5826 ST author **could** request authentication of the originator of the audit record, or non-repudiation
5827 of the origin of the record prior storing this record in the audit trail.

5828 The TSF will protect the stored audit data in the audit trail from unauthorized deletion and
5829 modification. It is noted that in some TOEs the auditor (role) might not be authorized to delete
5830 the audit records for a certain period of time.

5831 **C.7.3.2 Operations**

5832 In FAU_STG.2.2, the PP, PP-Module, functional package or ST author **should** specify whether the
5833 TSF **shall** prevent or only be able to detect modifications of the stored audit data in the audit
5834 trail. Only one of these options **may** be chosen.

5835 **C.7.4 FAU_STG.3 Guarantees of audit data availability**

5836 **C.7.4.1 Component rationale and application notes**

5837 This component allows the PP, PP-Module, functional package or ST author to specify to which
5838 metrics the audit trail **should** conform.

5839 In a distributed environment, as the location of the audit trail is in the TSF, but not necessarily
5840 co-located with the function generating the audit data, the PP, PP-Module, functional package or
5841 ST author **could** request authentication of the originator of the audit record, or non-repudiation
5842 of the origin of the record prior storing this record in the audit trail.

5843 **C.7.4.2 Operations**

5844 In FAU_STG.3.3, PP, PP-Module, functional package or ST author should specify the metric that
5845 the TSF must ensure with respect to the stored audit records. This metric limits the data loss by
5846 enumerating the number of records that must be kept, or the time that records are guaranteed
5847 to be maintained.

5848 **EXAMPLE**

5849 An example of the metric **could** be “100,000” indicating that 100,000 audit records **can** be stored.

5850 In FAU_STG.3.3 the PP, PP-Module, functional package or ST author **should** specify the condition
5851 under which the TSF **shall** still be able to maintain a defined amount of audit data. This
5852 condition **can** be any of the following: audit storage exhaustion, failure, attack.

5853 **C.7.5 FAU_STG.4 Prevention of audit data loss**

5854 **C.7.5.1 Component rationale and application notes**

5855 This component specifies the behaviour of the TOE if the audit trail is full: either audit records
 5856 are ignored, or the TOE is frozen such that no audited events **can** take place. The requirement
 5857 also states that no matter how the requirement is instantiated, the authorized user with specific
 5858 rights to this effect, **can** continue to generate audited events (actions). The reason is that
 5859 otherwise the authorized user **could** not even reset the TOE. Consideration **should** be given to
 5860 the choice of the action to be taken by the TSF in the case of audit storage exhaustion, as
 5861 ignoring events, which provides better availability of the TOE, will also permit actions to be
 5862 performed without being recorded and without the user being accountable.

5863 **C.7.5.2 Operations**

5864 In FAU_STG.5.1, the PP, PP-Module, functional package or ST author **should** select whether the
 5865 TSF **shall** ignore audited actions, or whether it **should** prevent audited actions from happening,
 5866 or whether the oldest audit records **should** be overwritten when the TSF **can** no longer store
 5867 audit records. Only one of these options **may** be chosen.

5868 In FAU_STG.5.1, the PP, PP-Module, functional package or ST author **should** specify other
 5869 actions that **should** be taken in case of audit storage failure, such as informing the authorized
 5870 user. If there is no other action to be taken in case of audit storage failure, this assignment **can**
 5871 be completed with “none”.

5872 **C.7.6 FAU_STG.5 Action in case of possible audit data loss**

5873 **C.7.6.1 Component rationale and application notes**

5874 This component requires that actions will be taken when the audit trail exceeds certain pre-
 5875 defined limits.

5876 **C.7.6.2 Operations**

5877 In FAU_STG.5 Prevention of audit data loss, the PP, PP-Module, functional package or ST author
 5878 **should** indicate the pre-defined limit. If the management functions indicate that this number
 5879 might be changed by the authorized user, this value is the default value. The PP, PP-Module,
 5880 functional package or ST author might choose to let the authorized user define this limit.

5881 **EXAMPLE**

5882 In the case that an authorized user defines the limit, an example of the assignment can be “an authorized user set
 5883 limit”.

5884 In FAU_STG.5 Prevention of audit data loss,, the PP, PP-Module, functional package or ST author
 5885 **should** specify actions that **should** be taken in case of imminent audit storage failure indicated
 5886 by exceeding the threshold. Actions might include informing an authorized user.

Annex D (normative)

Class FCO: Communication- application notes

5891 **D.1 General information**

5892 This class describes requirements specifically of interest for TOEs that are used for the
5893 transport of information. Families within this class deal with non-repudiation.

5894 In this class, the concept of “information” is used. This information **should** be interpreted as the
5895 object being communicated, and **could** contain an electronic mail message, a file, or a set of
5896 predefined attribute types.

5897 In the literature, the terms “proof of receipt” and “proof of origin” are commonly used terms.
5898 However, it is recognized that the term “proof” might be interpreted in a legal sense to imply a
5899 form of mathematical rationale. The components in this class interpret the de-facto use of the
5900 word “proof” in the context of “evidence” that the TSF demonstrates the non-repudiated
5901 transport of types of information.

5902 **D.2 Non-repudiation of origin (FCO_NRO)**

5903 **D.2.1 User application notes**

5904 Non-repudiation of origin defines requirements to provide evidence to users/subjects about the
5905 identity of the originator of some information. The originator cannot successfully deny having
5906 sent the information because evidence of origin provides evidence of the binding between the
5907 originator and the information sent. The recipient or a third party **can** verify the evidence of
5908 origin. This evidence **should** not be forgeable.

5909 EXAMPLE 1

5910 Evidence of origin could be a digital signature

5911 If the information or the associated attributes are altered in any way, validation of the evidence
5912 of origin might fail. Therefore, a PP, PP-Module, functional package or ST author **should**
5913 consider including integrity requirements such as FDP_UIT.1 Data exchange integrity in the PP,
5914 PP-Module, functional package or ST.

5915 In non-repudiation, there are several different roles involved, each of which **could** be combined
5916 in one or more subjects. The first role is a subject that requests evidence of origin (only in
5917 FCO_NRO.1 Selective proof of origin). The second role is the recipient and/or other subjects to
5918 which the evidence is provided. The third role is a subject that requests verification of the
5919 evidence of origin.

5920 EXAMPLE 2

5921 Subject that requests evidence of origin: a recipient or a third party such as an arbiter.

5922 Subject to which the evidence is provided: A notary

5923 The PP, PP-Module, functional package or ST author must specify the conditions that must be
5924 met to be able to verify the validity of the evidence.

5925 EXAMPLE 3

5926 An example of a condition which **could** be specified is where the verification of evidence must occur within 24 hours.

5927 These conditions, therefore, allow the tailoring of the non-repudiation to legal requirements,
5928 such as being able to provide evidence for several years.

5929 In most cases, the identity of the recipient will be the identity of the user who received the
5930 transmission. In some instances, the PP, PP-Module, functional package or ST author does not

5931 want the user identity to be exported. In that case, the PP, PP-Module, functional package or ST
 5932 author must consider whether it is appropriate to include this class, or whether the identity of
 5933 the transport service provider or the identity of the host **should** be used.

5934 In addition to (or instead of) the user identity, a PP, PP-Module, functional package or ST author
 5935 might be more concerned about the time the information was transmitted.

5936 EXAMPLE 4

5937 For example, requests for proposals must be transmitted before a certain date in order to be considered.

5938 In such instances, these requirements **can** be customized to provide a timestamp indication
 5939 (time of origin).

5940 **D.2.2 FCO_NRO.1 Selective proof of origin**

5941 **D.2.2.1 Operations**

5942 In FCO_NRO.1.1, the PP, PP-Module, functional package or ST author **should** fill in the types of
 5943 information subject to the evidence of origin function.

5944 EXAMPLE

5945 An example of the type of information is “electronic mail messages”

5946 In FCO_NRO.1.1, the PP, PP-Module, functional package or ST author **should** specify the
 5947 user/subject who **can** request evidence of origin.

5948 In FCO_NRO.1.1, the PP, PP-Module, functional package or ST author, dependent on the
 5949 selection, **should** specify the third parties that **can** request evidence of origin.

5950 EXAMPLE 1

5951 A third party **could** be an arbiter, judge, or legal body.

5952 In FCO_NRO.1.2, the PP, PP-Module, functional package or ST author **should** fill in the list of the
 5953 attributes that **shall** be linked to the information;

5954 EXAMPLE 2

5955 Attributes include originator identity, time of origin, and location of origin.

5956 In FCO_NRO.1.2, the PP, PP-Module, functional package or ST author **should** fill in the list of
 5957 information fields within the information over which the attributes provide evidence of origin,
 5958 such as the body of a message.

5959 In FCO_NRO.1.3, the PP, PP-Module, functional package or ST author **should** specify the
 5960 user/subject who **can** verify the evidence of origin.

5961 In FCO_NRO.1.3, the PP, PP-Module, functional package or ST author **should** fill in the list of
 5962 limitations under which the evidence **can** be verified.

5963 EXAMPLE

5964 An example of a limitation is “the evidence **can** only be verified within a 24-hour time interval.”

5965 An assignment of “immediate” or “indefinite” is acceptable.

5966 In FCO_NRO.1.3, the PP, PP-Module, functional package or ST author, dependent on the
 5967 selection, **should** specify the third parties that **can** verify the evidence of origin.

5968 **D.2.3 FCO_NRO.2 Enforced proof of origin**

5969 **D.2.3.1 Operations**

5970 In FCO_NRO.2.1, the PP, PP-Module, functional package or ST author **should** fill in the types of
 5971 information subject to the evidence of origin function.

5972 EXAMPLE

5973 electronic mail messages.

5974 In FCO_NRO.2.2, the PP, PP-Module, functional package or ST author **should** fill in the list of the
5975 attributes that **shall** be linked to the information; for example, originator identity, time of origin,
5976 and location of origin.

5977 In FCO_NRO.2.2, the PP, PP-Module, functional package or ST author **should** fill in the list of
5978 information fields within the information over which the attributes provide evidence of origin,
5979 such as the body of a message.

5980 In FCO_NRO.2.3, the PP, PP-Module, functional package or ST author **should** specify the
5981 user/subject who **can** verify the evidence of origin.

5982 In FCO_NRO.2.3, the PP, PP-Module, functional package or ST author **should** fill in the list of
5983 limitations under which the evidence **can** be verified.

5984 EXAMPLE

5985 For example, the evidence **can** only be verified within a 24-hour time interval.

5986 An assignment of “immediate” or “indefinite” is acceptable.

5987 In FCO_NRO.2.3, the PP, PP-Module, functional package or ST author, dependent on the
5988 selection, **should** specify the third parties that **can** verify the evidence of origin.

5989 EXAMPLE 2

5990 A third party **could** be an arbiter, judge, or legal body.

5991 **D.3 Non-repudiation of receipt (FCO_NRR)**

5992 **D.3.1 User application notes**

5993 Non-repudiation of receipt defines requirements to provide evidence to other users/subjects
5994 that the information was received by the recipient. The recipient cannot successfully deny
5995 having received the information because evidence of receipt provides evidence of the binding
5996 between the recipient attributes and the information. The originator or a third party **can** verify
5997 the evidence of receipt. This evidence **should** not be forgeable.

5998 EXAMPLE 1

5999 An example of a receipt is a digital signature

6000 It **should** be noted that the provision of evidence that the information was received does not
6001 necessarily imply that the information was read or comprehended, but only delivered.

6002 If the information or the associated attributes are altered in any way, validation of the evidence
6003 of receipt with respect to the original information might fail. Therefore, a PP, PP-Module,
6004 functional package or ST author **should** consider including integrity requirements such as
6005 FDP_UIT.1 Data exchange integrity in the PP, PP-Module, functional package or ST.

6006 In non-repudiation, there are several different roles involved, each of which **could** be combined
6007 in one or more subjects. The first role is a subject that requests evidence of receipt (only in
6008 FCO_NRR.1 Selective proof of receipt). The second role is the recipient and/or other subjects to
6009 which the evidence is provided). The third role is a subject that requests verification of the
6010 evidence of receipt, for example, an originator or a third party such as an arbiter.

6011 EXAMPLE 2

6012 A recipient or subject **could** be a notary.

6013 The PP, PP-Module, functional package or ST author must specify the conditions that must be
6014 met to be able to verify the validity of the evidence. An example of a condition which **could** be
6015 specified is where the verification of evidence must occur within 24 hours. These conditions,
6016 therefore, allow the tailoring of the non-repudiation to legal requirements, such as being able to
6017 provide evidence for several years.

6018 In most cases, the identity of the recipient will be the identity of the user who received the
6019 transmission. In some instances, the PP, PP-Module, functional package or ST author does not

6020 want the user identity to be exported. In that case, the PP, PP-Module, functional package or ST
 6021 author must consider whether it is appropriate to include this class, or whether the identity of
 6022 the transport service provider or the identity of the host **should** be used.

6023 In addition to (or instead of) the user identity, a PP, PP-Module, functional package or ST author
 6024 might be more concerned about the time the information was received.

6025 EXAMPLE 3

6026 When an offer expires at a certain date, orders must be received before a certain date in order to be considered.

6027 In such instances, these requirements **can** be customized to provide a timestamp indication
 6028 (time of receipt).

6029 **D.3.2 FCO_NRR.1 Selective proof of receipt**

6030 **D.3.2.1 Operations**

6031 In FCO_NRR.1.1, the PP, PP-Module, functional package or ST author **should** fill in the types of
 6032 information subject to the evidence of receipt function, for example, electronic mail messages.

6033 In FCO_NRR.1.1, the PP, PP-Module, functional package or ST author **should** specify the
 6034 user/subject who **can** request evidence of receipt.

6035 In FCO_NRR.1.1, the PP, PP-Module, functional package or ST author, dependent on the
 6036 selection, **should** specify the third parties that **can** request evidence of receipt.

6037 EXAMPLE

6038 A third party **could** be an arbiter, judge, or legal body.

6039 In FCO_NRR.1.2, the PP, PP-Module, functional package or ST author **should** fill in the list of the
 6040 attributes that **shall** be linked to the information; for example, recipient identity, time of receipt,
 6041 and location of receipt.

6042 In FCO_NRR.1.2, the PP, PP-Module, functional package or ST author **should** fill in the list of
 6043 information fields with the fields within the information over which the attributes provide
 6044 evidence of receipt, such as the body a message.

6045 In FCO_NRR.1.3, the PP, PP-Module, functional package or ST author **should** specify the
 6046 user/subjects who **can** verify the evidence of receipt.

6047 In FCO_NRR.1.3, the PP, PP-Module, functional package or ST author **should** fill in the list of
 6048 limitations under which the evidence **can** be verified. For example, the evidence **can** only be
 6049 verified within a 24-hour time interval. An assignment of “immediate” or “indefinite” is
 6050 acceptable.

6051 In FCO_NRR.1.3, the PP, PP-Module, functional package or ST author, dependent on the
 6052 selection, **should** specify the third parties that **can** verify the evidence of receipt.

6053 **D.3.3 FCO_NRR.2 Enforced proof of receipt**

6054 **D.3.3.1 Operations**

6055 In FCO_NRR.2.1, the PP, PP-Module, functional package or ST author **should** fill in the types of
 6056 information subject to the evidence of receipt function,

6057 EXAMPLE 1

6058 For example, electronic mail messages.

6059 In FCO_NRR.2.2, the PP, PP-Module, functional package or ST author **should** fill in the list of the
 6060 attributes that **shall** be linked to the information;

6061 EXAMPLE 2

6062 For example, recipient identity, time of receipt, and location of receipt.

6063 In FCO_NRR.2.2, the PP, PP-Module, functional package or ST author **should** fill in the list of
6064 information fields with the fields within the information over which the attributes provide
6065 evidence of receipt, such as the body of a message.

6066 In FCO_NRR.2.3, the PP, PP-Module, functional package or ST author **should** specify the
6067 user/subjects who **can** verify the evidence of receipt.

6068 In FCO_NRR.2.3, the PP, PP-Module, functional package or ST author **should** fill in the list of
6069 limitations under which the evidence **can** be verified. An assignment of “immediate” or
6070 “indefinite” is acceptable.

6071 EXAMPLE

6072 For example, the evidence **can** only be verified within a 24-hour time interval.

6073 In FCO_NRR.2.3, the PP, PP-Module, functional package or ST author, dependent on the
6074 selection, **should** specify the third parties that **can** verify the evidence of receipt. A third party
6075 **could** be an arbiter, judge or legal body.

Annex E (normative)

6076
6077
6078
6079

Class FCS: Cryptographic support- application notes

6080 E.1 General information

6081 The TSF **may** employ cryptographic functionality to help satisfy several high-level security
6082 objectives. These include, but are not limited to: identification and authentication, non-
6083 repudiation, trusted path, trusted channel and data separation. This class is used when the TOE
6084 implements cryptographic functions, the implementation of which **could** be in hardware,
6085 firmware and/or software.

6086 The FCS: Cryptographic support class is composed of four families: Cryptographic key
6087 management (FCS_CKM), Cryptographic operation (FCS_COP), Random bit generation
6088 (FCS_RBG), and Generation of random numbers (FCS_RNG).

6089 The Cryptographic key management (FCS_CKM) family addresses the management aspects of
6090 cryptographic keys; the Cryptographic operation (FCS_COP) family is concerned with the
6091 operational use of those cryptographic keys; the Random bit generation (FCS_RBG) family
6092 provides requirements for generating random bits; and the Generation of random numbers
6093 (FCS_RNG) is concerned with ensuring that random numbers meet defined quality metrics.

6094 For each cryptographic key generation method implemented by the TOE, if any, the PP, PP-
6095 Module, functional package or ST author **should** select the FCS_CKM.1 Cryptographic key
6096 generation component.

6097 For each cryptographic key distribution method implemented by the TOE, if any, the PP, PP-
6098 Module, functional package or ST author **should** select the FCS_CKM.2 Cryptographic key
6099 distribution.

6100 For each cryptographic key access method implemented by the TOE, if any, the PP, PP-Module,
6101 functional package or ST author **should** select the FCS_CKM.3 Cryptographic key access.

6102 For each cryptographic key derivation method implemented by the TOE, if any, the PP, PP-
6103 Module, functional package or ST author **should** select the FCS_CKM.5 Cryptographic key
6104 derivation.

6105 For each cryptographic key destruction method implemented by the TOE, if any, the PP, PP-
6106 Module, functional package or ST author **should** select the FCS_CKM.6 Timing and event of
6107 cryptographic key destruction component.

6108 For each cryptographic operation (such as digital signature, data encryption, key agreement,
6109 secure hash, etc.) performed by the TOE, if any, the PP, PP-Module, functional package or ST
6110 author **should** select the FCS_COP.1 Cryptographic operation component.

6111 For each deterministic random bit generation service implemented by the TOE, if any, the PP,
6112 PP-Module, functional package or ST author **should** select the FCS_RBG.1 Random bit
6113 generation (RBG) component.

6114 For each external seeding source used by the TOE, if any, the PP, PP-Module, functional package
6115 or ST author **should** select the FCS_RBG.2 Random bit generation (external seeding) component.

6116 For each internal seeding source (single) used by the TOE, if any, the PP, PP-Module, functional
6117 package or ST author **should** select the FCS_RBG.3 Random bit generation (internal seeding –
6118 single source) component.

6119 Where internal seeding source (multiple) is to be specified, the PP, PP-Module, functional
6120 package or ST author **should** select the FCS_RBG.4 Random bit generation (internal seeding –
6121 multiple sources) component.

6122 For cases where the TOE combines entropy sources, the FCS_RBG.5 Random bit generation
 6123 (combining entropy sources) component **should** be specified by PP, PP-Module, functional
 6124 package or ST author.

6125 For each random bit generation service implemented by the TOE, the PP, PP-Module, functional
 6126 package or ST author **should** specify the FCS_RBG.6 Random bit generation service component.

6127 For each random number generation service implemented by the TOE, the PP, PP-Module,
 6128 functional package or ST author **should** specify the FCS_RNG.1 Random number generation
 6129 component.

6130 Cryptographic functionality **may** be used to meet objectives specified in class FCO:
 6131 Communication, and in families Data authentication (FDP_DAU), Stored data integrity
 6132 (FDP_SDI), Inter-TSF user data confidentiality transfer protection (FDP_UCT), Inter-TSF user
 6133 data integrity transfer protection (FDP_UIT), Specification of secrets (FIA_SOS), User
 6134 authentication (FIA_UAU), to meet a variety of objectives. In the cases where cryptographic
 6135 functionality is used to meet objectives for other classes, the individual functional components
 6136 specify the objectives that cryptographic functionality must satisfy. The objectives in class FCS:
 6137 Cryptographic support **should** be used when cryptographic functionality of the TOE is sought by
 6138 consumers.

6139 **E.2 Cryptographic key management (FCS_CKM)**

6140 **E.2.1 User application notes**

6141 Cryptographic keys must be managed throughout their lifetime. The typical events in the
 6142 lifecycle of a cryptographic key include but are not limited to: key generation or derivation,
 6143 distribution, entry, storage, access, and destruction.

6144 EXAMPLE 1

- 6145 — backup
- 6146 — escrow
- 6147 — archive
- 6148 — recovery

6149 The inclusion of other stages is dependent on the key management strategy being implemented,
 6150 as the TOE is not always involved in all of the key life-cycle phases.

6151 EXAMPLE 2

6152 The TOE **may** only generate and distribute cryptographic keys.

6153 This family is intended to support the cryptographic key lifecycle and consequently defines
 6154 requirements for the following activities: cryptographic key generation, cryptographic key
 6155 derivation, cryptographic key distribution, cryptographic key access, and cryptographic key
 6156 destruction. This family **should** be included whenever there are functional requirements for the
 6157 management of cryptographic keys.

6158 If Security audit data generation (FAU_GEN) is included in the PP, PP-Module, functional
 6159 package or ST then, in the context of the events being audited:

- 6160 a) The object attributes **may** include the assigned user for the cryptographic key, the
 6161 user role, the cryptographic operation that the cryptographic key is to be used for,
 6162 the cryptographic key identifier and the cryptographic key validity period.
- 6163 b) The object value **may** include the values of cryptographic key(s) and parameters
 6164 excluding any sensitive information (such as secret or private cryptographic keys).

6165 Typically, random numbers are used to generate cryptographic keys. If this is the case, then
 6166 FCS_CKM.1 Cryptographic key generation **should** be used instead of the component FIA_SOS.2
 6167 TSF Generation of secrets. In cases where random number generation is required for purposes

6168 other than for the generation of cryptographic keys, the component FIA_SOS.2 TSF Generation
6169 of secrets **should** be used.

6170 **E.2.2 Evaluator notes**

6171 Evaluators should refer to ISO/IEC 15408-1:20XX, Annex B.4 for information in regard to the
6172 use of standards specified in FCS_CKM.5.

6173 FCS_CKM.5 has a dependency on FCS_CKM.6, The dependency should be understood as the
6174 dependency of two directions, 1) destruction of key derivation key, and 2) destruction of
6175 derived keys. Evaluators should keep in mind that the dependency of two directions has to be
6176 fulfilled, and should also consider any intermediate values (such as those from key
6177 establishment) that should be destroyed in order to preserve the confidentiality of the key.

6178 **E.2.3 FCS_CKM.1 Cryptographic key generation**

6179 **E.2.3.1 Component rationale and application notes**

6180 This component requires the cryptographic key sizes and method used to generate
6181 cryptographic keys to be specified, this **may** be in accordance with an assigned standard. It
6182 **should** be used to specify the cryptographic key sizes and the method used to generate the
6183 cryptographic keys. Only one instance of the component is needed for the same method and
6184 multiple key sizes. The key size **may** be common or different for the various entities and **may** be
6185 either the input to or the output from the method.

6186 EXAMPLE

6187 An example of a method is an algorithm.

6188 **E.2.3.2 Operations**

6189 In FCS_CKM.1.1, the PP, PP-Module, functional package or ST author **should** specify the
6190 cryptographic key generation algorithm to be used.

6191 In FCS_CKM.1.1, the PP, PP-Module, functional package or ST author **should** specify the
6192 cryptographic key sizes to be used. The key sizes specified **should** be appropriate for the
6193 algorithm and its intended use.

6194 In FCS_CKM.1.1, the PP, PP-Module, functional package or ST author **should** specify the assigned
6195 standard that documents the method used to generate cryptographic keys. The assigned
6196 standard **may** comprise none, one or more actual standards publications, for example, from
6197 international, national, industry or organizational standards.

6198 **E.2.4 FCS_CKM.2 Cryptographic key distribution**

6199 **E.2.4.1 Component rationale and application notes**

6200 This component requires the method used to distribute cryptographic keys to be specified, this
6201 **may** be in accordance with an assigned standard. See ISO/IEC 15408-1 for information on using
6202 standards in PPs, PP-Modules, functional packages and STs.

6203 **E.2.4.2 Operations**

6204 In FCS_CKM.2.1 the PP, PP-Module, functional package or ST author **should** specify the
6205 cryptographic key distribution method to be used.

6206 In FCS_CKM.2.1 the PP, PP-Module, functional package or ST author **should** specify the assigned
6207 standard that documents the method used to distribute cryptographic keys. The assigned
6208 standard **may** comprise none, one or more actual standards publications, for example, from
6209 international, national, industry or organizational standards.

6210 **E.2.5 FCS_CKM.3 Cryptographic key access**

6211 **E.2.5.1 Component rationale and application notes**

6212 This component is intended to allow the specification of requirements on the usage of keys
6213 outside the TOE (backup, archival, escrow, recovery, etc.) and requires the methods used to
6214 access cryptographic keys be specified, this **may** be in accordance with an assigned standard.

6215 FCS_CKM.3 Cryptographic key access is not intended to postulate the access control on
6216 cryptographic keys.

6217 **E.2.5.2 Operations**

6218 In FCS_CKM.3.1, the PP, PP-Module, functional package or ST author **should** specify the type of
6219 cryptographic key access being used.

6220 EXAMPLE

6221 Examples of types of cryptographic key access include (but are not limited to) cryptographic key backup,
6222 cryptographic key archival, cryptographic key escrow, and cryptographic key recovery.

6223 In FCS_CKM.3.1, the PP, PP-Module, functional package or ST author **should** specify the
6224 cryptographic key access method to be used.

6225 In FCS_CKM.3.1, the PP, PP-Module, functional package or ST author **should** specify the assigned
6226 standard that documents the method used to access cryptographic keys. The assigned standard
6227 **may** comprise none, one or more actual standards publications, for example, from international,
6228 national, industry or organizational standards.

6229 **E.2.6 FCS_CKM.5 Cryptographic key derivation**

6230 **E.2.6.1 Component rationale and application notes**

6231 This component requires the specification of the methods and parameters associated with the
6232 key derivation for a specified type of key.

6233 FCS_CKM.5 has a dependency on FCS_CKM.6, The dependency should be understood as the
6234 dependency of two directions, 1) destruction of key derivation key, and 2) destruction of
6235 derived keys. PP, PP-Module, functional package and ST authors should keep in mind that the
6236 dependency of two directions has to be fulfilled, and should also consider any intermediate
6237 values (such as those from key establishment) that should be destroyed in order to preserve the
6238 confidentiality of the key.

6239 **E.2.7 FCS_CKM.6 Timing and event of cryptographic key destruction**

6240 **E.2.7.1 Component rationale and application notes**

6241 This component requires the list of keys, including any keying material and the method used to
6242 destroy cryptographic keys to be specified, this can be in accordance with an assigned standard.

6243 The purpose of the destruction of cryptographic keys and keying material is to prevent their
6244 recovery in consideration of threats related to their compromise.

6245 NOTE Keying material includes keys and initialization vectors necessary to establish and maintain
6246 cryptographic keying relationships

6247 **E.2.7.2 Operations**

6248 In FCS_CKM.6.1, the PP, PP-Module, functional package or ST author provides a list of
6249 cryptographic keys and keying material that should be destroyed under certain circumstances.

6250 In FCS_CKM.6.2, the PP, PP-Module, functional package or ST author provides the cryptographic
6251 key destruction method and the standards specifying the cryptographic key destruction
6252 method.

6253 In FCS_CKM.6.1, the PP, PP-Module, functional package or ST author selects the circumstances
6254 of the destruction of key or key material.

6255 **E.3 Cryptographic operation (FCS_COP)**

6256 E.3.1 User application notes

6257 A cryptographic operation **may** have cryptographic mode(s) of operation associated with it. If
6258 this is the case, then the cryptographic mode(s) **shall** be specified.

6259 EXAMPLE

6260 Examples of cryptographic modes of operation are cipher block chaining, output feedback mode, electronic code
6261 book mode, and cipher feedback mode.

6262 Cryptographic operations **may** be used to support one or more TOE security services. The
6263 Cryptographic operation (FCS_COP) component **may** need to be iterated more than once
6264 depending on:

- 6265 a) the user application for which the security service is being used,
- 6266 b) the use of different cryptographic algorithms and/or cryptographic key sizes,
- 6267 c) the type or sensitivity of the data being operated on.

6268 If Security audit data generation (FAU_GEN) Security audit data generation is included in the
6269 PP, PP-Module, functional package or ST then, in the context of the cryptographic operation
6270 events being audited:

- 6271 a) The types of cryptographic operation **may** include digital signature generation
6272 and/or verification, cryptographic checksum generation for integrity and/or for
6273 verification of checksum, secure hash (message digest) computation, data
6274 encryption and/or decryption, cryptographic key encryption and/or decryption,
6275 cryptographic key agreement, and random number generation.
- 6276 b) The subject attributes **may** include subject role(s) and user(s) associated with the
6277 subject.
- 6278 c) The object attributes **may** include the assigned user for the cryptographic key, user
6279 role, cryptographic operation the cryptographic key is to be used for, cryptographic
6280 key identifier, and the cryptographic key validity period.

6281 When specifying cryptographic operations, the PP, PP-Module, functional package or ST author
6282 should perform due diligence in order to have confidence that the specified cryptographic
6283 operations are appropriate for the selected assurance requirements and in consideration of the
6284 technology types, environment and use cases of the TOE.

6285 NOTE In some cases, certification bodies can apply policies in regard to the selection of cryptographic
6286 operations. (See ISO/IEC 18045 A.5 n).

6287 E.3.2 FCS_COP.1 Cryptographic operation

6288 E.3.2.1 Component rationale and application notes

6289 This component requires the cryptographic algorithm and key size used to perform specified
6290 cryptographic operation(s) which **can** be based on an assigned standard.

6291 The dependencies to FCS_RBG.1 or FCS_RNG.1 will be required for cryptographic algorithm
6292 operations which internally generate random numbers.

6293 EXAMPLE 1

6294 DSA signature generation, ECDSA signature generation, RSASSA-PSS signature generation.

6295 The dependencies to FCS_RBG.1 or FCS_RNG.1 may not be necessary for deterministic
6296 cryptographic algorithm operations.

6297 EXAMPLE 2

6298 AES encryption / decryption in ECB mode.

6299 E.3.2.2 Operations

6300 In FCS_COP.1.1, the PP, PP-Module, functional package or ST author specifies the cryptographic
6301 operations being performed. Typical cryptographic operations include digital signature
6302 generation and/or verification, cryptographic checksum generation for integrity and/or for
6303 verification of checksum, secure hash (message digest) computation, data encryption and/or
6304 decryption, cryptographic key encryption and/or decryption, cryptographic key agreement, and
6305 random number generation. The cryptographic operation **may** be performed on user data or
6306 TSF data.

6307 In FCS_COP.1.1, the PP, PP-Module, functional package or ST author **should** specify the
6308 cryptographic algorithm to be used.

6309 EXAMPLE

6310 Examples of typical cryptographic algorithms include, but are not limited to, DES, RSA and IDEA.

6311 In FCS_COP.1.1, the PP, PP-Module, functional package or ST author **should** specify the
6312 cryptographic key sizes to be used. The key sizes specified **should** be appropriate for the
6313 algorithm and its intended use.

6314 In FCS_COP.1.1, the PP, PP-Module, functional package or ST author **should** specify the assigned
6315 standard that documents how the identified cryptographic operation(s) are performed. The
6316 assigned standard **may** comprise none, one or more actual standards publications, these **may**
6317 include standards from international, national, industry or organizational standards.

6318 **E.4 Random bit generation (FCS_RBG)**

6319 **E.4.1 User application notes**

6320 When specifying random bit generation methods, the PP, PP-Module, functional package or ST
6321 author should perform due diligence in order to have confidence that the specifications are
6322 appropriate for the selected assurance requirements and in consideration of the technology
6323 types, environment and use cases of the TOE.

6324 NOTE In some cases, certification bodies can apply policies in regard to the selection of random bit generators.
6325 (See ISO/IEC 18045 A.5 n).

6326 **E.4.2 FCS_RBG.1 Random bit generation (RBG)**

6327 **E.4.2.1 Component rationale and application notes**

6328 For FCS_RBG.1, these dependencies **shall** always be met.

6329 NOTE ISO/IEC 15408-1:20XX 8.3 item c) allowing a justification to be provided if a dependency is not met is not
6330 allowed for this component.

6331 In the RBG State Update Table the ST author **shall** include a row for initialization (Source1).
6332 Other rows are optional, depending on the noise sources supported by the TSF. The identifier
6333 values identify the specific source, so there should be a row for every unique source, and if the
6334 same source is used for more than one update type then the same identifier is given.

6335 If reseeding is not feasible, the TSF will unstantiate RBGs (and instantiate a new RBG), rather
6336 than produce output that is of insufficient quality. The listed standards should specify the
6337 reseed interval, and procedure for uninstantiating and reseeding. The 'Condition' selection
6338 allows the PP Author to require application-specific conditions for reseeding.

6339 "Unstantiate" means that the internal state of the DRBG is no longer available for use.

6340 In the 'Condition' selection, "on demand" means, that an interface to reseed is presented as a
6341 TSFI.

6342 EXAMPLE

6343 An example of an interface is an API call.

6344 Health tests for the RBG are specified in FPT_TST.1.

6345 **E.4.3 FCS_RBG.2 Random bit generation (external seeding)**

6346 **E.4.3.1 Component rationale and application notes**

6347 For this component, the interface to obtain the entropy noise source can be used multiple times
6348 to provide input. For instance, if the input length is 128 bits, it could be used twice to gather 256
6349 bits. In this instance, the 128 bits would not be provided to the DRBG, since the DRBG can only
6350 be instantiated once, rather a function would gather the 128 bits twice and provide the DRBG
6351 with 256 bits of entropy noise source.

6352 This component does not describe requirements on seed quality: it is the responsibility of the
6353 operational environment to define their requirement in this regard and to ensure that it is met
6354 by the external source.

6355 Guidance in the introduction to PP, PP-Module, functional package or ST authors should address
6356 protection from modification and disclosure of the value from the external noise source, as well
6357 as the leaking of any pertinent information (e.g., internal state) regarding the RBG.

6358 **E.4.4 FCS_RBG.3 Random bit generation (internal seeding – single source)**

6359 **E.4.4.1 Component rationale and application notes**

6360 If an ST Author wishes to use multiple internal noise sources, they iterate this requirement for
6361 each noise source being used by the TSF.

6362 Hardware-based noise sources are sources whose primary function is noise generation, such as
6363 ring oscillators, diodes, and thermal noise. While software is used to collect the noise from these
6364 hardware sources, these are not software-based. Software-based noise sources are those
6365 sources that have some other primary function and the noise is a byproduct of their normal
6366 operation. Examples of software-based noise sources are user or system-based events, reading
6367 the least significant bits from an event timer, etc.

6368 Hardware-based noise sources may be stochastically modeled, in which case the amount of
6369 entropy is well understood. Software-based noise sources are usually less well understood and
6370 therefore will typically take a more conservative approach, gathering larger numbers of bits
6371 than required and then performing a compression function to derive the final output. Software-
6372 based noise sources often rely on an entropy estimator.

6373 **E.5 Generation of random numbers (FCS_RNG)**

6374 **Previous calls for contributions for the application notes for FCS_RNG have been made.**

6375 **In the absence of any contribution, the editors have drawn from PP-0084B**
6376 **(https://www.commoncriteriaportal.org/files/ppfiles/pp0084b_pdf.pdf)**

6377 **The editor requests careful review by SMEs.**

6378 **E.5.1 User application notes**

6379 When specifying random number generation methods, the PP, PP-Module, functional package
6380 or ST author should perform due diligence in order to have confidence that the specifications
6381 are appropriate for the selected assurance requirements and in consideration of the technology
6382 types, environment and use cases of the TOE.

6383 NOTE In some cases, certification bodies can apply policies in regard to the selection of random bit generators.
6384 (See ISO/IEC 18045 A.5 n).

6385 **E.5.2 FCS_RNG.1 Random number generation**

6386 **E.5.2.1 Component rationale and application notes**

6387 The ST writer **shall** perform the missing operation appropriate for cryptographic application of
6388 the random numbers in the elements FCS_RNG.1.1 and FCS_RNG_1.2. The ST writer **shall**

6389 perform the selections for specification of the security capabilities provided by the random
6390 number generator of the TOE.

6391 NOTE Some users of FCS_RNG may find The National Institute of Standards and Technology (NIST) Special
6392 Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators,
6393 June 2015 and NIST Special Publication 800-90B Recommendation for the Entropy Sources Used for Random Bit
6394 Generation, January 2018 useful.

6395 The evaluation of the random number generator shall follow a recognized methodology,

6396 EXAMPLE

6397 An example of a recognized methodology is AIS31.

6398 E.5.2.2 Operations

6399 In FCS_RNG.1 .1 the PP, PP-Module, functional package or ST author should specify the list of
6400 security capabilities.

6401 EXAMPLE 1

6402 Examples of security capabilities include

6403 — A total failure test detects a total failure of entropy source immediately when the RNG has started. When a
6404 total failure is detected, no random numbers will be output.

6405 — If a total failure of the entropy source occurs while the RNG is being operated, the RNG [selection: prevents
6406 the output of any internal random number that depends on some raw random numbers that have been
6407 generated after the total failure of the entropy source, generates the internal random numbers with a post-
6408 processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output
6409 entropy].

6410 — The online test shall detect non-tolerable statistical defects of the raw random number sequence (i)
6411 immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output
6412 any random numbers before the power-up online test has finished successfully or when a defect has been
6413 detected.

6414 — The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers
6415 soon.

6416 — The online test procedure checks the quality of the raw random number sequence. It is triggered [selection:
6417 externally, at regular intervals, continuously, applied upon specified internal events]. The online test is
6418 suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random
6419 numbers within an acceptable period of time.

6420 — Failure or severe degradation of the noise source shall be detectable.

6421 — Continuous tests or other mechanisms in the entropy source shall protect against producing output during
6422 malfunctions.

6423 NOTE In the case of a PP, PP-Module or functional package, FCS_RNG.1 .1 could be completed with a more
6424 restrictive language such as:

6425 — assignment: list of additional security capabilities.

6426 In FCS_RNG.1.2 the PP, PP-Module, functional package or ST author should make the
6427 appropriate selection in regard to the quality metric.

6428 EXAMPLE 2

6429 Examples of quality metrics include

6430 — Test procedure A [assignment: additional standard test suites] does not distinguish the internal random
6431 numbers from output sequences of an ideal RNG.

6432 NOTE The assignment for additional standard statistical test suite may be empty.

6433 — The average Shannon entropy per internal random bit exceeds 0.99751.

6434 — each output bit is independent of all other output bits,

6435 NOTE In the case of a PP, PP-Module or functional package, FCS_RNG.1 .2 could be completed with a more
6436 restrictive language such as:

6437 — [selection:

6438 ○ full entropy output,

6439 ○ [assignment: bias and entropy rate of the output]]

6440

6441 EXAMPLE 3

6442 In the case of a hybrid deterministic RNG, the following is an example:

6443 FCS_RNG.1.1/HD

6444 The TSF shall provide a hybrid deterministic⁵⁵ random number generator that implements: [selection: CTR_DRBG,
6445 Hash_DRBG, HMAC_DRBG] as defined in NIST Special Publication 800-90A.

6446 FCS_RNG.1.2/HD

6447 The TSF shall provide [selection: bits, octets of bits, numbers [assignment: format of the numbers]] that meet
6448 [assignment: security bits].

Annex F (normative)

Class FDP: User data protection- application notes

6453 F.1 General information

6454 This class contains families specifying requirements related to protecting user data. This class
6455 differs from FIA and FPT in that FDP: User data protection specifies components to protect user
6456 data, FIA specifies components to protect attributes associated with the user, and FPT specifies
6457 components to protect TSF information.

6458 The class does not contain explicit requirements for traditional Mandatory Access Controls
6459 (MAC) or traditional Discretionary Access Controls (DAC); however, such requirements **may** be
6460 constructed using components from this class.

6461 FDP: User data protection does not explicitly deal with confidentiality, integrity, or availability,
6462 as all three are most often intertwined in the policy and mechanisms. However, the TOE
6463 security policy **shall** adequately cover these three objectives in the PP, PP-Module, functional
6464 package or ST.

6465 A final aspect of this class is that it specifies access control in terms of “operations”. An
6466 operation is defined as a specific type of access on a specific object. It depends on the level of
6467 abstraction of the PP, PP-Module, functional package or ST author whether these operations are
6468 described as “read” and/or “write” operations, or as more complex operations such as “update
6469 the database”.

6470 The access control policies are policies that control access to the information container. The
6471 attributes represent attributes of the container. Once the information is out of the container, the
6472 accessor is free to modify that information, including writing the information into a different
6473 container with different attributes. By contrast, an information flow policy controls access to
6474 the information, independent of the container. The attributes of the information, which **may** be
6475 associated with the attributes of the container (or **may** not, as in the case of a multi-level
6476 database) stay with the information as it moves. The accessor does not have the ability, in the
6477 absence of an explicit authorization, to change the attributes of the information.

6478 This class is not meant to be a complete taxonomy of IT access policies, as others **can** be
6479 imagined. Those policies included here are simply those for which current experience with
6480 actual systems provides a basis for specifying requirements. There **may** be other forms of intent
6481 that are not captured in the definitions here.

6482 EXAMPLE 1

6483 For example, a goal of having user-imposed (and user-defined) controls on information flow (such as. an automated
6484 implementation of the NO FOREIGN handling caveat).

6485 Such concepts **could** be handled as refinements of, or extensions to the FDP: User data
6486 protection components.

6487 Finally, it is important when looking at the components in FDP: User data protection to
6488 remember that these components are requirements for functions that **may** be implemented by a
6489 mechanism that also serves or **could** serve another purpose.

6490 EXAMPLE 2

6491 It is possible to build an access control policy (Access control policy (FDP_ACC)) that uses labels (FDP_1FF.1 Simple
6492 security attributes) as the basis of the access control mechanism.

6493 A set of SFRs **may** encompass many security function policies (SFPs), each to be identified by
6494 the two policy-oriented components Access control policy (FDP_ACC), and Information flow
6495 control policy (FDP_IFC). These policies will typically take confidentiality, integrity, and
6496 availability aspects into consideration as required, to satisfy the TOE requirements. Care **should**

6497 be taken to ensure that all objects are covered by at least one SFP and that there are no conflicts
6498 arising from implementing the multiple SFPs.

6499 When building a PP, PP-Module, functional package or ST using components from the FDP: User
6500 data protection class, the following information provides guidance on where to look and what
6501 to select from the class.

6502 The requirements in the FDP: User data protection class are defined in terms of a set of SFRs
6503 that will implement a SFP. Since a TOE **may** implement multiple SFPs simultaneously, the PP,
6504 PP-Module, functional package or ST author **shall** specify the name for each SFP, so it **can** be
6505 referenced in other families. This name will then be used in each component selected to indicate
6506 that it is being used as part of the definition of requirements for that SFP. This allows the author
6507 to easily indicate the scope for operations such as objects covered, operations covered,
6508 authorized users, etc.

6509 Each instantiation of a component **can** apply to only one SFP. Therefore, if an SFP is specified in
6510 a component then this SFP will apply to all the elements in this component. The components
6511 **may** be instantiated multiple times within a PP, PP-Module, functional package or ST to account
6512 for different policies if so desired.

6513 The key to selecting components from this family is to have a well-defined set of TOE security
6514 objectives to enable proper selection of the components from the two policy components;
6515 Access control policy (FDP_ACC) and Information flow control policy (FDP_IFC). In Access
6516 control policy (FDP_ACC) and Information flow control policy (FDP_IFC) respectively, all access
6517 control policies and all information flow control policies are named. Furthermore, the scope of
6518 control of these components in terms of the subjects, objects and operations covered by this
6519 security functionality. The names of these policies are meant to be used throughout the
6520 remainder of the functional components that have an operation that calls for an assignment or
6521 selection of an “access control SFP” or an “information flow control SFP”. The rules that define
6522 the functionality of the named access control and information flow control SFPs will be defined
6523 in the Access control functions (FDP_ACF) and Information flow control functions (FDP_IFF)
6524 families (respectively).

6525 The following steps are guidance on how this class is applied in the construction of a PP, PP-
6526 Module, functional package or ST:

- 6527 a) Identify the policies to be enforced from the Access control policy (FDP_ACC), and
6528 Information flow control policy (FDP_IFC) families. These families define scope of
6529 control for the policy, granularity of control and **may** identify some rules to go with
6530 the policy.
- 6531 b) Identify the components and perform any applicable operations in the policy
6532 components. The assignment operations **may** be performed generally (such as with
6533 a statement “All files”) or specifically (“The files “A”, “B”, etc.) depending upon the
6534 level of detail known.
- 6535 c) Identify any applicable function components from the Access control functions
6536 (FDP_ACF) and Information flow control functions (FDP_IFF) families to address
6537 the named policy families from Access control policy (FDP_ACC) and Information
6538 flow control policy (FDP_IFC). Perform the operations to make the components
6539 define the rules to be enforced by the named policies. This **should** make the
6540 components fit the requirements of the selected function envisioned or to be built.
- 6541 d) Identify who will have the ability to control and change security attributes under
6542 the function, such as only a security administrator, only the owner of the object, etc.
6543 Select the appropriate components from FMT: Security management and perform
6544 the operations. Refinements **may** be useful here to identify missing features, such
6545 as that some or all changes **shall** be done via trusted path.
- 6546 e) Identify any appropriate components from the FMT: Security management for
6547 initial values for new objects and subjects.

- 6548 f) Identify any applicable rollback components from the Rollback (FDP_ROL) family.
- 6549 g) Identify any applicable residual information protection requirements from the
6550 Residual information protection (FDP_RIP) family.
- 6551 h) Identify any applicable import or export components, and how security attributes
6552 **should** be handled during import and export, from the Import from outside of the
6553 TOE (FDP_ITC) and Export from the TOE (FDP_ETC) families.
- 6554 i) Identify any applicable internal TOE communication components from the Internal
6555 TOE transfer (FDP_ITT) family.
- 6556 j) Identify any requirements for integrity protection of stored information from the
6557 Stored data integrity (FDP_SDI).
- 6558 k) Identify any applicable inter-TSF communication components from the Inter-TSF
6559 user data confidentiality transfer protection (FDP_UCT) or Inter-TSF user data
6560 integrity transfer protection (FDP_UIT) families.

6561 **F.2 Access control policy (FDP_ACC)**

6562 **F.2.1 User application notes**

6563 This family is based upon the concept of arbitrary controls on the interaction of subjects and
6564 objects. The scope and purpose of the controls is based upon the attributes of the accessor
6565 (subject), the attributes of the container being accessed (object), the actions (operations) and
6566 any associated access control rules.

6567 The components in this family are capable of identifying the access control SFPs (by name) to
6568 be enforced by the traditional Discretionary Access Control (DAC) mechanisms. It further
6569 defines the subjects, objects and operations that are covered by identified access control SFPs.
6570 The rules that define the functionality of an access control SFP will be defined by other families,
6571 such as Access control functions (FDP_ACF) and Export from the TOE (FDP_ETC). The names of
6572 the access control SFPs defined in Access control policy (FDP_ACC) are meant to be used
6573 throughout the remainder of the functional components that have an operation that calls for an
6574 assignment or selection of an “access control SFP.”

6575 The access control SFP covers a set of triplets: subject, object, and operations. Therefore, a
6576 subject **can** be covered by multiple access control SFPs but only with respect to a different
6577 operation or a different object. Of course, the same applies to objects and operations.

6578 A critical aspect of an access control function that enforces an access control SFP is the ability
6579 for users to modify the attributes involved in access control decisions. The Access control policy
6580 (FDP_ACC) family does not address these aspects. Some of these requirements are left
6581 undefined, but **can** be added as refinements, while others are covered elsewhere in other
6582 families and classes such as FMT: Security management.

6583 There are no audit requirements in Access control policy (FDP_ACC) as this family specifies
6584 access control SFP requirements. Audit requirements will be found in families specifying
6585 functions to satisfy the access control SFPs identified in this family.

6586 This family provides a PP, PP-Module, functional package or ST author the capability to specify
6587 several policies, for example, a fixed access control SFP to be applied to one scope of control,
6588 and a flexible access control SFP to be defined for a different scope of control. To specify more
6589 than one access control policy, the components from this family **can** be iterated multiple times
6590 in a PP, PP-Module, functional package or ST to different subsets of operations and objects. This
6591 will accommodate TOEs that contain multiple policies, each addressing a particular set of
6592 operations and objects. In other words, the PP, PP-Module, functional package or ST author
6593 **should** specify the required information in the ACC component for each of the access control
6594 SFPs that the TSF will enforce. For example, a TOE incorporating three access control SFPs, each
6595 covering only a subset of the objects, subjects, and operations within the TOE, will contain one

6596 FDP_ACC.1 Subset access control component for each of the three access-control SFPs,
6597 necessitating a total of three FDP_ACC.1 Subset access control components.

6598 **F.2.2 FDP_ACC.1 Subset access control**

6599 **F.2.2.1 Component rationale and application notes**

6600 The terms object and subject refer to generic elements in the TOE. For a policy to be
6601 implementable, the entities **shall** be clearly identified. For a PP, the objects and operations
6602 might be expressed as types such as: named objects, data repositories, observe accesses, etc.
6603 For a specific TOE these generic terms (subject, object) **shall** be refined.

6604 EXAMPLE

6605 files, registers, ports, daemons, open calls, etc.

6606 This component specifies that the policy cover some well-defined set of operations on some
6607 subset of the objects. It places no constraints on any operations outside the set - including
6608 operations on objects for which other operations are controlled.

6609 **F.2.2.2 Operations**

6610 In FDP_ACC.1.1, the PP, PP-Module, functional package or ST author **should** specify a uniquely
6611 named access control SFP to be enforced by the TSF.

6612 In FDP_ACC.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
6613 subjects, objects, and operations among subjects and objects covered by the SFP.

6614 **F.2.3 FDP_ACC.2 Complete access control**

6615 **F.2.3.1 Component rationale and application notes**

6616 This component requires that all possible operations on objects, that are included in the SFP,
6617 are covered by an access control SFP.

6618 The PP, PP-Module, functional package or ST author **shall** demonstrate that each combination of
6619 objects and subjects is covered by an access control SFP.

6620 **F.2.3.2 Operations**

6621 In FDP_ACC.2.1, the PP, PP-Module, functional package or ST author **should** specify a uniquely
6622 named access control SFP to be enforced by the TSF.

6623 In FDP_ACC.2.1, the PP, PP-Module, functional package or ST author **should** specify the list of
6624 subjects and objects covered by the SFP. All operations among those subjects and objects will be
6625 covered by the SFP.

6626 **F.3 Access control functions (FDP_ACF)**

6627 **F.3.1 User application notes**

6628 This family describes the rules for the specific functions that **can** implement an access control
6629 policy named in Access control policy (FDP_ACC) which also specifies the scope of control of the
6630 policy.

6631 This family provides a PP, PP-Module, functional package or ST author the capability to describe
6632 the rules for access control. This results in a TOE where the access to objects will not change. An
6633 example of such an object is "Message of the Day", which is readable by all, and changeable only
6634 by the authorized administrator. This family also provides the PP, PP-Module, functional
6635 package or ST author with the ability to describe rules that provide for exceptions to the
6636 general access control rules. Such exceptions would either explicitly allow or deny
6637 authorization to access an object.

6638 There are no explicit components to specify other possible functions such as two-person
6639 control, sequence rules for operations, or exclusion controls. However, these mechanisms, as

6640 well as traditional DAC mechanisms, **can** be represented with the existing components, by
 6641 careful drafting of the access control rules.

6642 A variety of acceptable access control functionality **may** be specified in this family.

6643 EXAMPLE

- 6644 — Access control lists (ACLs)
- 6645 — Time-based access control specifications
- 6646 — Origin-based access control specifications
- 6647 — Owner-controlled access control attributes

6648 **F.3.2 FDP_ACF.1 Security attribute based access control**

6649 **F.3.2.1 Component rationale and application notes**

6650 This component provides requirements for a mechanism that mediates access control based on
 6651 security attributes associated with subjects and objects. Each object and subject has a set of
 6652 associated attributes, such as location, time of creation, access rights such as Access Control
 6653 Lists (ACLs)). This component allows the PP, PP-Module, functional package or ST author to
 6654 specify the attributes that will be used for the access control mediation. This component allows
 6655 access control rules, using these attributes, to be specified.

6656 EXAMPLE

6657 Examples of the attributes that a PP, PP-Module, functional package or ST author might assign are:

6658 An identity attribute may be associated with users, subjects, or objects to be used for mediation. Examples of such
 6659 attributes might be the name of the program image used in the creation of the subject, or a security attribute
 6660 assigned to the program image.

6661 A time attribute can be used to specify that access will be authorized during certain times of the day, during certain
 6662 days of the week, or during a certain calendar year.

6663 A location attribute **could** specify whether the location is the location of the request for the operation, the location
 6664 where the operation will be carried out, or both. It **could** be based upon internal tables to translate the logical
 6665 interfaces of the TSF into locations such as through terminal locations, CPU locations, etc.

6666 A grouping attribute allows a single group of users to be associated with an operation for the purposes of access
 6667 control. If required, the refinement operation should be used to specify the maximum number of definable groups,
 6668 the maximum membership of a group, and the maximum number of groups to which a user can concurrently be
 6669 associated.

6670 This component also provides requirements for the access control security functions to be able
 6671 to explicitly authorize or deny access to an object based upon security attributes. This **could** be
 6672 used to provide privilege, access rights, or access authorizations within the TOE. Such
 6673 privileges, rights, or authorizations **could** apply to users, subjects (representing users or
 6674 applications), and objects.

6675 **F.3.2.2 Operations**

6676 In FDP_ACF.1.1, the PP, PP-Module, functional package or ST author **should** specify an access
 6677 control SFP name that the TSF is to enforce. The name of the access control SFP, and the scope
 6678 of control for that policy are defined in components from Access control policy (FDP_ACC).

6679 In FDP_ACF.1.1, the PP, PP-Module, functional package or ST author **should** specify, for each
 6680 controlled subject and object, the security attributes and/or named groups of security
 6681 attributes that the function will use in the specification of the rules. For example, such attributes
 6682 **may** be things such as the user identity, subject identity, role, time of day, location, ACLs, or any
 6683 other attribute specified by the PP, PP-Module, functional package or ST author. Named groups
 6684 of security attributes **can** be specified to provide a convenient means to refer to multiple
 6685 security attributes. Named groups **could** provide a useful way to associate “roles” defined in
 6686 Security management roles (FMT_SMR), and all of their relevant attributes, with subjects. In
 6687 other words, each role **could** relate to a named group of attributes.

6688 In FDP_ACF.1.2, the PP, PP-Module, functional package or ST author **should** specify the SFP rules
 6689 governing access among controlled subjects and controlled objects using controlled operations
 6690 on controlled objects. These rules specify when access is granted or denied. It **can** specify
 6691 general access control functions or granular access control functions.

6692 EXAMPLE

6693 General access control functions: typical permission bits

6694 Granular access control: Access Control Lists (ACL)

6695 In FDP_ACF.1.3, the PP, PP-Module, functional package or ST author **should** specify the rules,
 6696 based on security attributes, that explicitly authorize access of subjects to objects that will be
 6697 used to explicitly authorize access. These rules are in addition to those specified in FDP_ACF.1.1.
 6698 They are included in FDP_ACF.1.3 as they are intended to contain exceptions to the rules in
 6699 FDP_ACF.1.1. An example of rules to explicitly authorize access is based on a privilege vector
 6700 associated with a subject that always grants access to objects covered by the access control SFP
 6701 that has been specified. If such a capability is not desired, then the PP, PP-Module, functional
 6702 package or ST author **should** specify “none”.

6703 In FDP_ACF.1.4, the PP, PP-Module, functional package or ST author **should** specify the rules,
 6704 based on security attributes, that explicitly deny access of subjects to objects. These rules are in
 6705 addition to those specified in FDP_ACF.1.1 . They are included in FDP_ACF.1.4 as they are
 6706 intended to contain exceptions to the rules in FDP_ACF.1.1 . An example of rules to explicitly
 6707 deny access is based on a privilege vector associated with a subject that always denies access to
 6708 objects covered by the access control SFP that has been specified. If such a capability is not
 6709 desired, then the PP, PP-Module, functional package or ST author **should** specify “none”.

6710 **F.4 Data authentication (FDP_DAU)**

6711 **F.4.1 User application notes**

6712 This family describes specific functions that **can** be used to authenticate “static” data.

6713 Components in this family are to be used when there is a requirement for “static” data
 6714 authentication, i.e. where data is to be signed but not transmitted.

6715 Note the Non-repudiation of origin (FCO_NRO) family provides for non-repudiation of origin of information
 6716 received during a data exchange.

6717 **F.4.2 FDP_DAU.1 Basic Data Authentication**

6718 **F.4.2.1 Component rationale and application notes**

6719 This component **may** be satisfied by one-way hash functions to generate a hash value for a
 6720 definitive document that **may** be used as verification of the validity or authenticity of its
 6721 information content.

6722 EXAMPLE

6723 cryptographic checksum, fingerprint, message digest

6724 **F.4.2.2 Operations**

6725 In FDP_DAU.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
 6726 objects or information types for which the TSF **shall** be capable of generating data
 6727 authentication evidence.

6728 In FDP_DAU.1.2, the PP, PP-Module, functional package or ST author **should** specify the list of
 6729 subjects that will have the ability to verify data authentication evidence for the objects
 6730 identified in the previous element. The list of subjects **could** be very specific, if the subjects are
 6731 known, or it **could** be more generic and refer to a “type” of subject such as an identified role.

6732 **F.4.3 FDP_DAU.2 Data Authentication with Identity of Guarantor**

6733 **F.4.3.1 Component rationale and application notes**

6734 This component additionally requires the ability to verify the identity of the user that provided
6735 the guarantee of authenticity

6736 EXAMPLE

6737 A trusted third party.

6738 **F.4.3.2 Operations**

6739 In FDP_DAU.2.1, the PP, PP-Module, functional package or ST author **should** specify the list of
6740 objects or information types for which the TSF **shall** be capable of generating data
6741 authentication evidence.

6742 In FDP_DAU.2.2, the PP, PP-Module, functional package or ST author **should** specify the list of
6743 subjects that will have the ability to verify data authentication evidence for the objects
6744 identified in the previous element as well as the identity of the user that created the data
6745 authentication evidence.

6746 **F.5 Export from the TOE (FDP_ETC)**

6747 **F.5.1 User application notes**

6748 This family defines functions for TSF-mediated exporting of user data from the TOE such that its
6749 security attributes either **can** be explicitly preserved or **can** be ignored once it has been
6750 exported. Consistency of these security attributes are addressed by Inter-TSF TSF data
6751 consistency (FPT_TDC).

6752 Export from the TOE (FDP_ETC) is concerned with limitations on export and association of
6753 security attributes with the exported user data.

6754 This family, and the corresponding Import family Import from outside of the TOE (FDP_ITC),
6755 address how the TOE deals with user data transferred into and outside its control. In principle,
6756 this family is concerned with the TSF-mediated exporting of user data and its related security
6757 attributes.

6758 A variety of activities might be involved here:

- 6759 a) exporting of user data without any security attributes;
- 6760 b) exporting user data including security attributes where the two are associated with
6761 one another and the security attributes unambiguously represent the exported
6762 user data.

6763 If there are multiple SFPs (access control and/or information flow control) then it **may** be
6764 appropriate to iterate these components once for each named SFP.

6765 **F.5.2 FDP_ETC.1 Export of user data without security attributes**

6766 **F.5.2.1 Component rationale and application notes**

6767 This component is used to specify the TSF-mediated exporting of user data without the export
6768 of its security attributes.

6769 **F.5.2.2 Operations**

6770 In FDP_ETC.1.1, the PP, PP-Module, functional package or ST author **should** specify the access
6771 control SFP(s) and/or information flow control SFP(s) that will be enforced when exporting
6772 user data. The user data that this function exports is scoped by the assignment of these SFPs.

6773 **F.5.3 FDP_ETC.2 Export of user data with security attributes**

6774 **F.5.3.1 Component rationale and application notes**

6775 The user data is exported together with its security attributes. The security attributes are
 6776 unambiguously associated with the user data. There are several ways of achieving this
 6777 association. One way that this **can** be achieved is by physically collocating the user data and the
 6778 security attributes.

6779 EXAMPLE

6780 On the same external media

6781 or by using cryptographic techniques such as secure signatures to associate the attributes and
 6782 the user data. Inter-TSF trusted channel (FTP_ITC) **could** be used to assure that the attributes
 6783 are correctly received at the other trusted IT product while Inter-TSF TSF data consistency
 6784 (FPT_TDC) **can** be used to make sure that those attributes are properly interpreted.

6785 Furthermore, Trusted path (FTP_TRP) **could** be used to make sure that the export is being
 6786 initiated by the proper user.

6787 **F.5.3.2 Operations**

6788 In FDP_ETC.2.1, the PP, PP-Module, functional package or ST author **should** specify the access
 6789 control SFP(s) and/or information flow control SFP(s) that will be enforced when exporting
 6790 user data. The user data that this function exports is scoped by the assignment of these SFPs.

6791 In FDP_ETC.2.5, the PP, PP-Module, functional package or ST author **should** specify any
 6792 additional exportation control rules or “none” if there are no additional exportation control
 6793 rules. These rules will be enforced by the TSF in addition to the access control SFPs and/or
 6794 information flow control SFPs selected in FDP_ETC.2.1.

6795 **F.6 Information flow control policy (FDP_IFC)**

6796 **F.6.1 User application notes**

6797 This family covers the identification of information flow control SFPs; and, for each, specifies
 6798 the scope of control of the SFP.

6799 The components in this family are capable of identifying the information flow control SFPs to be
 6800 enforced by the traditional Mandatory Access Control (MAC) mechanisms that would be found
 6801 in a TOE. However, they go beyond just the traditional MAC mechanisms and **can** be used to
 6802 identify and describe non-interference policies and state-transitions. It further defines the
 6803 subjects under control of the policy, the information under control of the policy, and operations
 6804 which cause controlled information to flow to and from controlled subjects for each information
 6805 flow control SFP in the TOE. The information flow control SFP will be defined by other families
 6806 such as Information flow control functions (FDP_IFF) and Export from the TOE (FDP_ETC). The
 6807 information flow control SFPs named here in Information flow control policy (FDP_IFC) are
 6808 meant to be used throughout the remainder of the functional components that have an
 6809 operation that calls for an assignment or selection of an “information flow control SFP.”

6810 These components are quite flexible. They allow the domain of flow control to be specified and
 6811 there is no requirement that the mechanism be based upon labels. The different elements of the
 6812 information flow control components also permit different degrees of exception to the policy.

6813 Each SFP covers a set of triplets: subject, information, and operations that cause information to
 6814 flow to and from subjects. Some information flow control policies **may** be at a very low level of
 6815 detail and explicitly describe subjects in terms of processes within an operating system. Other
 6816 information flow control policies **may** be at a high level and describe subjects in the generic
 6817 sense of users or input/output channels. If the information flow control policy is at too high a
 6818 level of detail, it **may** not clearly define the desired IT security functions. In such cases, it is
 6819 more appropriate to include such descriptions of information flow control policies as objectives.
 6820 Then the desired IT security functions **can** be specified as supportive of those objectives.

6821 In the second component (FDP_IFC.2 Complete information flow control), each information flow
 6822 control SFP will cover all possible operations that cause information covered by that SFP to flow
 6823 to and from subjects covered by that SFP. Furthermore, all information flows will need to be

6824 covered by a SFP. Therefore, for each action that causes information to flow, there will be a set
6825 of rules that define whether the action is allowed. If there are multiple SFPs that are applicable
6826 for a given information flow, all involved SFPs **shall** allow this flow before it is permitted to take
6827 place.

6828 An information flow control SFP covers a well-defined set of operations. The SFPs coverage **may**
6829 be “complete” with respect to some information flows, or it **may** address only some of the
6830 operations that affect the information flow.

6831 An access control SFP controls access to the objects that contain information. An information
6832 flow control SFP controls access to the information, independent of its container. The attributes
6833 of the information, which **may** be associated with the attributes of the container (or **may** not, as
6834 in the case of a multi-level database) stay with the information as it flows. The accessor does
6835 not have the ability, in the absence of an explicit authorization, to change the attributes of the
6836 information.

6837 Information flows and operations **can** be expressed at multiple levels. In the case of a ST, the
6838 information flows and operations might be specified at a system-specific level: TCP/IP packets
6839 flowing through a firewall based upon known IP addresses. For a PP, the information flows and
6840 operations might be expressed as types: email, data repositories, observe accesses, etc.

6841 The components in this family **can** be applied multiple times in a PP, PP-Module, functional
6842 package or ST to different subsets of operations and objects. This will accommodate TOEs that
6843 contain multiple policies, each addressing a particular set of objects, subjects, and operations.

6844 **F.6.2 FDP_IFC.1 Subset information flow control**

6845 **F.6.2.1 Component rationale and application notes**

6846 This component requires that an information flow control policy apply to a subset of the
6847 possible operations in the TOE.

6848 **F.6.2.2 Operations**

6849 In FDP_IFC.1.1, the PP, PP-Module, functional package or ST author **should** specify a uniquely
6850 named information flow control SFP to be enforced by the TSF.

6851 In FDP_IFC.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
6852 subjects, information, and operations which cause controlled information to flow to and from
6853 controlled subjects covered by the SFP. As mentioned above, the list of subjects **could** be at
6854 various levels of detail depending on the needs of the PP, PP-Module, functional package or ST
6855 author.

6856 **EXAMPLE**

6857 It **could** specify users, machines, or processes.

6858 Information **could** refer to data such as email or network protocols, or more specific objects
6859 similar to those specified under an access control policy. If the information that is specified is
6860 contained within an object that is subject to an access control policy, then both the access
6861 control policy and information flow control policy **shall** be enforced before the specified
6862 information **could** flow to or from the object.

6863 **F.6.3 FDP_IFC.2 Complete information flow control**

6864 **F.6.3.1 Component rationale and application notes**

6865 This component requires that all possible operations that cause information to flow to and from
6866 subjects included in the SFP, are covered by an information flow control SFP.

6867 The PP, PP-Module, functional package or ST author **shall** demonstrate that each combination of
6868 information flows and subjects is covered by an information flow control SFP.

6869 **F.6.3.2 Operations**

6870 In FDP_IFC.2.1, the PP, PP-Module, functional package or ST author **should** specify a uniquely
6871 named information flow control SFP to be enforced by the TSF.

6872 In FDP_IFC.2.1, the PP, PP-Module, functional package or ST author **should** specify the list of
6873 subjects and information that will be covered by the SFP. All operations that cause that
6874 information to flow to and from subjects will be covered by the SFP. As mentioned above, the
6875 list of subjects **could** be at various levels of detail depending on the needs of the PP, PP-Module,
6876 functional package or ST author.

6877 EXAMPLE

6878 It **could** specify users, machines, or processes.

6879 Information **could** refer to data such as email or network protocols, or more specific objects
6880 similar to those specified under an access control policy. If the information that is specified is
6881 contained within an object that is subject to an access control policy, then both the access
6882 control policy and information flow control policy **shall** be enforced before the specified
6883 information **could** flow to or from the object.

6884 **F.7 Information flow control functions (FDP_IFF)**

6885 **F.7.1 User application notes**

6886 This family describes the rules for the specific functions that **can** implement the information
6887 flow control SFPs named in Information flow control policy (FDP_IFC), which also specifies the
6888 scope of control of the policies. It consists of two “trees:” one addressing the common
6889 information flow control function issues, and a second addressing illicit information flows (i.e.
6890 covert channels) with respect to one or more information flow control SFPs. This division arises
6891 because the issues concerning illicit information flows are, in some sense, orthogonal to the rest
6892 of an SFP. Illicit information flows are flows in violation of policy; thus, they are not a policy
6893 issue.

6894 In order to implement strong protection against disclosure or modification in the face of
6895 untrusted software, controls on information flow are required. Access controls alone are not
6896 sufficient because they only control access to containers, allowing the information they contain
6897 to flow, without controls, throughout a system.

6898 In this family, the phrase “types of illicit information flows” is used. This phrase **may** be used to
6899 refer to the categorization of flows as “Storage Channels” or “Timing Channels”, or it **can** refer to
6900 improved categorizations reflective of the needs of a PP, PP-Module, functional package or ST
6901 author.

6902 The flexibility of these components allows the definition of a privilege policy within FDP_IFF.1
6903 Simple security attributes and FDP_IFF.2 Hierarchical security attributes to allow the controlled
6904 bypass of all or part of a particular SFP. If there is a need for a predefined approach to SFP
6905 bypass, the PP, PP-Module, functional package or ST author **should** consider incorporating a
6906 privilege policy.

6907 **F.7.2 FDP_IFF.1 Simple security attributes**

6908 **F.7.2.1 Component rationale and application notes**

6909 This component requires security attributes on information, and on subjects that cause that
6910 information to flow and subjects that act as recipients of that information. The attributes of the
6911 containers of the information **should** also be considered if it is desired that they **should** play a
6912 part in information flow control decisions or if they are covered by an access control policy.
6913 This component specifies the key rules that are enforced and describes how security attributes
6914 are derived.

6915 This component does not specify the details of how a security attribute is assigned (i.e. user
6916 versus process). Flexibility in policy is provided by having assignments that allow specification
6917 of additional policy and function requirements, as necessary.

6918 This component also provides requirements for the information flow control functions to be
 6919 able to explicitly authorize and deny an information flow based upon security attributes. This
 6920 **could** be used to implement a privilege policy that covers exceptions to the basic policy defined
 6921 in this component.

6922 **F.7.2.2 Operations**

6923 In FDP_IFF.1.1, the PP, PP-Module, functional package or ST author **should** specify the
 6924 information flow control SFPs enforced by the TSF. The name of the information flow control
 6925 SFP, and the scope of control for that policy are defined in components from Information flow
 6926 control policy (FDP_IFC).

6927 In FDP_IFF.1.1, the PP, PP-Module, functional package or ST author **should** specify, for each type
 6928 of controlled subject and information, the security attributes that are relevant to the
 6929 specification of the SFP rules.

6930 EXAMPLE 1

6931 For example, such security attributes **may** be things such the subject identifier, subject sensitivity label, subject
 6932 clearance label, information sensitivity label, etc.

6933 The types of security attributes **should** be sufficient to support the environmental needs.

6934 In FDP_IFF.1.2, the PP, PP-Module, functional package or ST author **should** specify for each
 6935 operation, the security attribute-based relationship that must hold between subject and
 6936 information security attributes that the TSF will enforce.

6937 In FDP_IFF.1.3, the PP, PP-Module, functional package or ST author **should** specify any
 6938 additional information flow control SFP rules that the TSF is to enforce. This includes all rules of
 6939 the SFP that are either not based on the security attributes of the information and the subject or
 6940 rules that automatically modify the security attributes of information or subjects as a result of
 6941 an access operation. An example for the first case is a rule of the SFP controlling a threshold
 6942 value for specific types of information. This would for example be the case when the
 6943 information flow SFP contains rules on access to statistical data where a subject is only allowed
 6944 to access this type of information up to a specific number of accesses. An example for the
 6945 second case would be a rule stating under which conditions and how the security attributes of a
 6946 subject or object change as the result of an access operation. Some information flow policies for
 6947 example **may** limit the number of access operations to information with specific security
 6948 attributes. If there are no additional rules then the PP, PP-Module, functional package or ST
 6949 author **should** specify “none”.

6950 In FDP_IFF.1.4, the PP, PP-Module, functional package or ST author **should** specify the rules,
 6951 based on security attributes, that explicitly authorize information flows. These rules are in
 6952 addition to those specified in the preceding elements. They are included in FDP_IFF.1.4 as they
 6953 are intended to contain exceptions to the rules in the preceding elements.

6954 EXAMPLE 2

6955 An example of rules to explicitly authorize information flows is based on a privilege vector associated with a subject
 6956 that always grants the subject the ability to cause an information flow for information that is covered by the SFP that
 6957 has been specified.

6958 If such a capability is not desired, then the PP, PP-Module, functional package or ST author
 6959 **should** specify “none”.

6960 In FDP_IFF.1.5, the PP, PP-Module, functional package or ST author **should** specify the rules,
 6961 based on security attributes, that explicitly deny information flows. These rules are in addition
 6962 to those specified in the preceding elements. They are included in FDP_IFF.1.5 as they are
 6963 intended to contain exceptions to the rules in the preceding elements. An example of rules to
 6964 explicitly deny information flows is based on a privilege vector associated with a subject that
 6965 always denies the subject the ability to cause an information flow for information that is
 6966 covered by the SFP that has been specified. If such a capability is not desired, then the PP, PP-
 6967 Module, functional package or ST author **should** specify “none”.

6968 **F.7.3 FDP_IFF.2 Hierarchical security attributes**

6969 **F.7.3.1 Component rationale and application notes**

6970 This component requires that the named information flow control SFP uses hierarchical
6971 security attributes that form a lattice.

6972 It is important to note that the hierarchical relationship requirements identified in FDP_IFF.2.4
6973 need only apply to the information flow control security attributes for the information flow
6974 control SFPs that have been identified in FDP_IFF.2.1. This component is not meant to apply to
6975 other SFPs such as access control SFPs.

6976 FDP_IFF.2.6 phrases the requirements for the set of security attributes to form a lattice. A
6977 number of information flow policies defined in the literature and implemented in IT products
6978 are based on a set of security attributes that form a lattice. FDP_IFF.2.6 is specifically included
6979 to address this type of information flow policies.

6980 If it is the case that multiple information flow control SFPs are to be specified, and that each of
6981 these SFPs will have their own security attributes that are not related to one another, then the
6982 PP, PP-Module, functional package or ST author **should** iterate this component once for each of
6983 those SFPs. Otherwise a conflict might arise with the sub-items of FDP_IFF.2.4 since the
6984 required relationships will not exist.

6985 **F.7.3.2 Operations**

6986 In FDP_IFF.2.1, the PP, PP-Module, functional package or ST author **should** specify the
6987 information flow control SFPs enforced by the TSF. The name of the information flow control
6988 SFP, and the scope of control for that policy are defined in components from Information flow
6989 control policy (FDP_IFC).

6990 In FDP_IFF.2.1, the PP, PP-Module, functional package or ST author **should** specify, for each type
6991 of controlled subject and information, the security attributes that are relevant to the
6992 specification of the SFP rules. For example, such security attributes **may** be things such the
6993 subject identifier, subject sensitivity label, subject clearance label, information sensitivity label,
6994 etc. The types of security attributes **should** be sufficient to support the environmental needs.

6995 In FDP_IFF.2.2, the PP, PP-Module, functional package or ST author **should** specify for each
6996 operation, the security attribute-based relationship that must hold between subject and
6997 information security attributes that the TSF will enforce. These relationships **should** be based
6998 upon the ordering relationships between the security attributes.

6999 In FDP_IFF.2.3, the PP, PP-Module, functional package or ST author **should** specify any
7000 additional information flow control SFP rules that the TSF is to enforce. This includes all rules of
7001 the SFP that are either not based on the security attributes of the information and the subject or
7002 rules that automatically modify the security attributes of information or subjects as a result of
7003 an access operation. An example for the first case is a rule of the SFP controlling a threshold
7004 value for specific types of information.

7005 EXAMPLE 1

7006 This would for example be the case when the information flow SFP contains rules on access to statistical data where
7007 a subject is only allowed to access this type of information up to a specific number of accesses. An example for the
7008 second case would be a rule stating under which conditions and how the security attributes of a subject or object
7009 change as the result of an access operation.

7010 Some information flow policies **may** limit the number of access operations to information with
7011 specific security attributes. If there are no additional rules then the PP, PP-Module, functional
7012 package or ST author **should** specify "none".

7013 In FDP_IFF.2.4, the PP, PP-Module, functional package or ST author **should** specify the rules,
7014 based on security attributes, that explicitly authorize information flows. These rules are in
7015 addition to those specified in the preceding elements. They are included in FDP_IFF.2.4 as they
7016 are intended to contain exceptions to the rules in the preceding elements.

7017 EXAMPLE 2

7018 An example of rules to explicitly authorize information flows is based on a privilege vector associated with a subject
7019 that always grants the subject the ability to cause an information flow for information that is covered by the SFP that
7020 has been specified.

7021 If such a capability is not desired, then the PP, PP-Module, functional package or ST author
7022 **should** specify “none”.

7023 In FDP_IFF.2.5, the PP, PP-Module, functional package or ST author **should** specify the rules,
7024 based on security attributes, that explicitly deny information flows. These rules are in addition
7025 to those specified in the preceding elements. They are included in FDP_IFF.2.5 as they are
7026 intended to contain exceptions to the rules in the preceding elements. An example of rules to
7027 explicitly deny information flows is based on a privilege vector associated with a subject that
7028 always denies the subject the ability to cause an information flow for information that is
7029 covered by the SFP that has been specified. If such a capability is not desired, then the PP, PP-
7030 Module, functional package or ST author **should** specify “none”.

7031 **F.7.4 FDP_IFF.3 Limited illicit information flows**

7032 **F.7.4.1 Component rationale and application notes**

7033 This component **should** be used when at least one of the SFPs that requires control of illicit
7034 information flows does not require elimination of flows.

7035 For the specified illicit information flows, certain maximum capacities **should** be provided. In
7036 addition, a PP, PP-Module, functional package or ST author has the ability to specify whether
7037 the illicit information flows must be audited.

7038 **F.7.4.2 Operations**

7039 In FDP_IFF.3.1, the PP, PP-Module, functional package or ST author **should** specify the
7040 information flow control SFPs enforced by the TSF. The name of the information flow control
7041 SFP, and the scope of control for that policy are defined in components from Information flow
7042 control policy (FDP_IFC).

7043 In FDP_IFF.3.1, the PP, PP-Module, functional package or ST author **should** specify the types of
7044 illicit information flows that are subject to a maximum capacity limitation.

7045 In FDP_IFF.3.1, the PP, PP-Module, functional package or ST author **should** specify the maximum
7046 capacity permitted for any identified illicit information flows.

7047 **F.7.5 FDP_IFF.4 Partial elimination of illicit information flows**

7048 **F.7.5.1 Component rationale and application notes**

7049 This component **should** be used when all the SFPs that requires control of illicit information
7050 flows require elimination of some (but not necessarily all) illicit information flows.

7051 **F.7.5.2 Operations**

7052 In FDP_IFF.4.1, the PP, PP-Module, functional package or ST author **should** specify the
7053 information flow control SFPs enforced by the TSF. The name of the information flow control
7054 SFP, and the scope of control for that policy are defined in components from Information flow
7055 control policy (FDP_IFC).

7056 In FDP_IFF.4.1, the PP, PP-Module, functional package or ST author **should** specify the types of
7057 illicit information flows which are subject to a maximum capacity limitation.

7058 In FDP_IFF.4.1, the PP, PP-Module, functional package or ST author **should** specify the maximum
7059 capacity permitted for any identified illicit information flows.

7060 In FDP_IFF.4.2, the PP, PP-Module, functional package or ST author **should** specify the types of
7061 illicit information flows to be eliminated. This list **may** not be empty as this component requires
7062 that some illicit information flows are to be eliminated.

7063 **F.7.6 FDP_IFF.5 No illicit information flows**

7064 **F.7.6.1 Component rationale and application notes**

7065 This component **should** be used when the SFPs that require control of illicit information flows
 7066 require elimination of all illicit information flows. However, the PP, PP-Module, functional
 7067 package or ST author **should** carefully consider the potential impact that eliminating all illicit
 7068 information flows might have on the normal functional operation of the TOE. Many practical
 7069 applications have shown that there is an indirect relationship between illicit information flows
 7070 and normal functionality within a TOE and eliminating all illicit information flows **may** result in
 7071 less than desired functionality.

7072 **F.7.6.2 Operations**

7073 In FDP_IFF.5.1, the PP, PP-Module, functional package or ST author **should** specify the
 7074 information flow control SFP for which illicit information flows are to be eliminated. The name
 7075 of the information flow control SFP, and the scope of control for that policy are defined in
 7076 components from Information flow control policy (FDP_IFC).

7077 **F.7.7 FDP_IFF.6 Illicit information flow monitoring**

7078 **F.7.7.1 Component rationale and application notes**

7079 This component **should** be used when it is desired that the TSF provide the ability to monitor
 7080 the use of illicit information flows that exceed a specified capacity. If it is desired that such flows
 7081 be audited, then this component **could** serve as the source of audit events to be used by
 7082 components from the Security audit data generation (FAU_GEN) family.

7083 **F.7.7.2 Operations**

7084 In FDP_IFF.6.1, the PP, PP-Module, functional package or ST author **should** specify the
 7085 information flow control SFPs enforced by the TSF. The name of the information flow control
 7086 SFP, and the scope of control for that policy are defined in components from Information flow
 7087 control policy (FDP_IFC).

7088 In FDP_IFF.6.1, the PP, PP-Module, functional package or ST author **should** specify the types of
 7089 illicit information flows that will be monitored for exceeding a maximum capacity.

7090 In FDP_IFF.6.1, the PP, PP-Module, functional package or ST author **should** specify the maximum
 7091 capacity above which illicit information flows will be monitored by the TSF.

7092 **F.8 Information retention control (FDP_IRC)**

7093 **F.8.1 User application notes**

7094 While a great aspect of the elimination of the objects as required by FDP_IRC refers to the
 7095 information stored within the object as a container, it also includes all attributes (also in the
 7096 meaning of metadata) that may be associated with the object.

7097 In this aspect, the focus of FDP_IRC differs from other components related to access or
 7098 information flow control policies, such as FDP_IFF and FDP_IFC. More important, objects here
 7099 are always considered in the context of selected activities that are performed on these objects.
 7100 In contrast to residual information protection (FDP_RIP), FDP_IRC excludes objects from any
 7101 access or information flow and deletes them, irreversibly and untraceably when they are no
 7102 longer needed by a set of activities.

7103 While it may not be completely clear, which objects to consider, it is essential that the list of
 7104 objects is assigned by the PP, PP-Module, functional package or ST author at the very latest in
 7105 order to allow for concrete tests. In any case the list of objects shall be derived from a
 7106 structured analysis.

7107 **F.8.2 FDP_IRC.1 Information retention control**

7108 **F.8.2.1 Component rationale and application notes**

7109 The Information erasure policy as defined in FDP_IRC.1 serves to protect all information that is
7110 contained in the assigned objects from being misused, regardless of whether the information is
7111 primary content or any kind of attribute. The policy covers combinations of objects and
7112 activities. The policy's coverage **may** be "complete" with respect to all the objects related to one
7113 or more activities, or it **may** address only some of objects related to one or more activities.

7114 The term "promptly" in FDP_IRC.1 specifically refers to the fact that the objects shall be
7115 terminated in a manner that ensures that they cannot be accessed as before.

7116 **F.8.2.2 Operations**

7117 In FDP_IRC.1.1, the PP, PP-Module, functional package or ST author **should** specify a uniquely
7118 named information erasure policy to be enforced by the TSF.

7119 In FDP_IRC.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
7120 objects that are required for the respective list of activities, e.g. "all message objects".

7121 In FDP_IRC.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
7122 activities that the information erasure policy is concerned with, e.g. "all activities related to
7123 passing a message on, such as receiving a message, cryptographic handling of a message,
7124 sending a message".

7125 In FDP_IRC.1.2, the PP, PP-Module, functional package or ST author **should** specify the list of
7126 objects that are required for the respective list of activities. This assignment **shall** be identical to
7127 the assigned objects in FDP_IRC.1.1.

7128 **F.9 Import from outside of the TOE (FDP_ITC)**

7129 **F.9.1 User application notes**

7130 This family defines mechanisms for TSF-mediated importing of user data from outside the TOE
7131 into the TOE such that the user data security attributes **can** be preserved. Consistency of these
7132 security attributes are addressed by Inter-TSF TSF data consistency (FPT_TDC).

7133 Import from outside of the TOE (FDP_ITC) is concerned with limitations on import, user
7134 specification of security attributes, and association of security attributes with the user data.

7135 This family, and the corresponding export family Export from the TOE (FDP_ETC), address how
7136 the TOE deals with user data outside its control. This family is concerned with assigning and
7137 abstraction of the user data security attributes.

7138 EXAMPLE 1

7139 A variety of activities might be involved here:

- 7140 a) importing user data from an unformatted medium (such as., tape, scanner, video or audio signal), without
7141 including any security attributes, and physically marking the medium to indicate its contents;
- 7142 b) importing user data, including security attributes, from a medium and verifying that the object security
7143 attributes are appropriate;
- 7144 c) importing user data, including security attributes, from a medium using a cryptographic sealing technique
7145 to protect the association of user data and security attributes.

7146 This family is not concerned with the determination of whether the user data **may** be imported.
7147 It is concerned with the values of the security attributes to associate with the imported user
7148 data.

7149 There are two possibilities for the import of user data: either the user data is unambiguously
7150 associated with reliable object security attributes (values and meaning of the security attributes
7151 is not modified), or no reliable security attributes (or no security attributes at all) are available
7152 from the import source. This family addresses both cases.

7153 If there are reliable security attributes available, they **may** have been associated with the user
 7154 data by physical means (the security attributes are on the same media), or by logical means (the
 7155 security attributes are distributed differently but include unique object identification).

7156 EXAMPLE 2

7157 Cryptographic checksum

7158 This family is concerned with TSF-mediated importing of user data and maintaining the
 7159 association of security attributes as required by the SFP. Other families are concerned with
 7160 other import aspects such as consistency, trusted channels, and integrity that are beyond the
 7161 scope of this family. Furthermore, Import from outside of the TOE (FDP_ITC) is only concerned
 7162 with the interface to the import medium. Export from the TOE (FDP_ETC) is responsible for the
 7163 other end point of the medium (the source).

7164 Some of the well-known import requirements are:

- 7165 a) importing of user data without any security attributes;
- 7166 b) importing of user data including security attributes where the two are associated
 7167 with one another and the security attributes unambiguously represent the
 7168 information being imported.

7169 These import requirements **may** be handled by the TSF with or without human intervention,
 7170 depending on the IT limitations and the organizational security policy. For example, if user data
 7171 is received on a “confidential” channel, the security attributes of the objects will be set to
 7172 “confidential”.

7173 If there are multiple SFPs (access control and/or information flow control) then it **may** be
 7174 appropriate to iterate these components once for each named SFP.

7175 **F.9.2 FDP_ITC.1 Import of user data without security attributes**

7176 **F.9.2.1 Component rationale and application notes**

7177 This component is used to specify the import of user data that does not have reliable (or any)
 7178 security attributes associated with it. This function requires that the security attributes for the
 7179 imported user data be initialized within the TSF. It **could** also be the case that the PP, PP-
 7180 Module, functional package or ST author specifies the rules for import. It **may** be appropriate, in
 7181 some environments, to require that these attributes be supplied via a trusted path or a trusted
 7182 channel mechanism.

7183 **F.9.2.2 Operations**

7184 In FDP_ITC.1.1, the PP, PP-Module, functional package or ST author **should** specify the access
 7185 control SFP(s) and/or information flow control SFP(s) that will be enforced when importing
 7186 user data from outside of the TOE. The user data that this function imports is scoped by the
 7187 assignment of these SFPs.

7188 In FDP_ITC.1.3, the PP, PP-Module, functional package or ST author **should** specify any
 7189 additional importation control rules or “none” if there are no additional importation control
 7190 rules. These rules will be enforced by the TSF in addition to the access control SFPs and/or
 7191 information flow control SFPs selected in FDP_ITC.1.1.

7192 **F.9.3 FDP_ITC.2 Import of user data with security attributes**

7193 **F.9.3.1 Component rationale and application notes**

7194 This component is used to specify the import of user data that has reliable security attributes
 7195 associated with it. This function relies upon the security attributes that are accurately and
 7196 unambiguously associated with the objects on the import medium. Once imported, those
 7197 objects will have those same attributes. This requires Inter-TSF TSF data consistency
 7198 (FPT_TDC) to ensure the consistency of the data. It **could** also be the case that the PP, PP-
 7199 Module, functional package or ST author specifies the rules for import.

7200 **F.9.3.2 Operations**

7201 In FDP_ITC.2.1, the PP, PP-Module, functional package or ST author **should** specify the access
7202 control SFP(s) and/or information flow control SFP(s) that will be enforced when importing
7203 user data from outside of the TOE. The user data that this function imports is scoped by the
7204 assignment of these SFPs.

7205 In FDP_ITC.2.5, the PP, PP-Module, functional package or ST author **should** specify any
7206 additional importation control rules or “none” if there are no additional importation control
7207 rules. These rules will be enforced by the TSF in addition to the access control SFPs and/or
7208 information flow control SFPs selected in FDP_ITC.2.1.

7209 **F.10 Internal TOE transfer (FDP_ITT)**7210 **F.10.1 User application notes**

7211 This family provides requirements that address protection of user data when it is transferred
7212 between parts of a TOE across an internal channel. This **may** be contrasted with the Inter-TSF
7213 user data confidentiality transfer protection (FDP_UCT) and Inter-TSF user data integrity
7214 transfer protection (FDP_UIT) family, which provide protection for user data when it is
7215 transferred between distinct TSFs across an external channel, and Export from the TOE
7216 (FDP_ETC) and Import from outside of the TOE (FDP_ITC), which address TSF-mediated
7217 transfer of data to or from outside the TOE.

7218 The requirements in this family allow a PP, PP-Module, functional package or ST author to
7219 specify the desired security for user data while in transit within the TOE. This security **could** be
7220 protection against disclosure, modification, or loss of availability.

7221 The determination of the degree of physical separation above which this family **should** apply
7222 depends on the intended environment of use. In a hostile environment, there **may** be risks
7223 arising from transfers between parts of the TOE separated by only a system bus. In more benign
7224 environments, the transfers **may** be across more traditional network media.

7225 If there are multiple SFPs (access control and/or information flow control) then it **may** be
7226 appropriate to iterate these components once for each named SFP.

7227 **F.10.2 FDP_ITT.1 Basic internal transfer protection**7228 **F.10.2.1 Operations**

7229 In FDP_ITT.1.1, the PP, PP-Module, functional package or ST author **should** specify the access
7230 control SFP(s) and/or information flow control SFP(s) covering the information being
7231 transferred.

7232 In FDP_ITT.1.1, the PP, PP-Module, functional package or ST author **should** specify the types of
7233 transmission errors that the TSF **should** prevent occurring for user data while in transport. The
7234 options are disclosure, modification, loss of use.

7235 **F.10.3 FDP_ITT.2 Transmission separation by attribute**7236 **F.10.3.1 Component rationale and application notes**

7237 This component **could**, for example, be used to provide different forms of protection to
7238 information with different clearance levels.

7239 One of the ways to achieve separation of data when it is transmitted is through the use of
7240 separate logical or physical channels.

7241 **F.10.3.2 Operations**

7242 In FDP_ITT.2.1, the PP, PP-Module, functional package or ST author **should** specify the access
7243 control SFP(s) and/or information flow control SFP(s) covering the information being
7244 transferred.

7245 In FDP_ITT.2.1, the PP, PP-Module, functional package or ST author **should** specify the types of
 7246 transmission errors that the TSF **should** prevent occurring for user data while in transport. The
 7247 options are disclosure, modification, loss of use.

7248 In FDP_ITT.2.2, the PP, PP-Module, functional package or ST author **should** specify the security
 7249 attributes, the values of which the TSF will use to determine when to separate data that is being
 7250 transmitted between physically-separated parts of the TOE. An example is that user data
 7251 associated with the identity of one owner is transmitted separately from the user data
 7252 associated with the identify of a different owner. In this case, the value of the identity of the
 7253 owner of the data is what is used to determine when to separate the data for transmission.

7254 **F.10.4 FDP_ITT.3 Integrity monitoring**

7255 **F.10.4.1 Component rationale and application notes**

7256 This component is used in combination with either FDP_ITT.1 Basic internal transfer protection
 7257 or FDP_ITT.2 Transmission separation by attribute. It ensures that the TSF checks received user
 7258 data (and their attributes) for integrity. FDP_ITT.1 Basic internal transfer protection or
 7259 FDP_ITT.2 Transmission separation by attribute will provide the data in a manner such that it is
 7260 protected from modification (so that FDP_ITT.3 Integrity monitoring **can** detect any
 7261 modifications).

7262 The PP, PP-Module, functional package or ST author has to specify the types of errors that must
 7263 be detected. The PP, PP-Module, functional package or ST author **should** consider: modification
 7264 of data, substitution of data, unrecoverable ordering change of data, replay of data, incomplete
 7265 data, in addition to other integrity errors.

7266 The PP, PP-Module, functional package or ST author must specify the actions that the TSF
 7267 **should** take on detection of a failure.

7268 EXAMPLE

7269 For example: ignore the user data, request the data again, inform the authorized administrator, reroute traffic for
 7270 other lines.

7271 **F.10.4.2 Operations**

7272 In FDP_ITT.3.1, the PP, PP-Module, functional package or ST author **should** specify the access
 7273 control SFP(s) and/or information flow control SFP(s) covering the information being
 7274 transferred and monitored for integrity errors.

7275 In FDP_ITT.3.1, the PP, PP-Module, functional package or ST author **should** specify the type of
 7276 possible integrity errors to be monitored during transmission of the user data.

7277 In FDP_ITT.3.2, the PP, PP-Module, functional package or ST author **should** specify the action to
 7278 be taken by the TSF when an integrity error is encountered.

7279 EXAMPLE

7280 An example is that the TSF should request the resubmission of the user data. The SFP(s) specified in FDP_ITT.3.1 will
 7281 be enforced as the actions are taken by the TSF.

7282 **F.10.5 FDP_ITT.4 Attribute-based integrity monitoring**

7283 **F.10.5.1 Component rationale and application notes**

7284 This component is used in combination with FDP_ITT.2 Transmission separation by attribute. It
 7285 ensures that the TSF checks received user data, that has been transmitted by separate channels
 7286 (based on values of specified security attributes), for integrity. It allows the PP, PP-Module,
 7287 functional package or ST author to specify actions to be taken upon detection of an integrity
 7288 error.

7289 EXAMPLE 1

7290 This component **could** be used to provide different integrity error detection and action for information at different
 7291 integrity levels.

7292 The PP, PP-Module, functional package or ST author has to specify the types of errors that must
7293 be detected. The PP, PP-Module, functional package or ST author **should** consider: modification
7294 of data, substitution of data, unrecoverable ordering change of data, replay of data, incomplete
7295 data, in addition to other integrity errors.

7296 The PP, PP-Module, functional package or ST author **should** specify the attributes (and
7297 associated transmission channels) that necessitate integrity error monitoring.

7298 The PP, PP-Module, functional package or ST author must specify the actions that the TSF
7299 **should** take on detection of a failure.

7300 EXAMPLE 2

7301 For example: ignore the user data, request the data again, inform the authorized administrator, reroute traffic for
7302 other lines.

7303 **F.10.5.2 Operations**

7304 In FDP_ITT.4.1, the PP, PP-Module, functional package or ST author **should** specify the access
7305 control SFP(s) and/or information flow control SFP(s) covering the information being
7306 transferred and monitored for integrity errors.

7307 In FDP_ITT.4.1, the PP, PP-Module, functional package or ST author **should** specify the type of
7308 possible integrity errors to be monitored during transmission of the user data.

7309 In FDP_ITT.4.1, the PP, PP-Module, functional package or ST author **should** specify a list of
7310 security attributes that require separate transmission channels. This list is used to determine
7311 which user data to monitor for integrity errors., based on its security attributes and its
7312 transmission channel. This element is directly related to FDP_ITT.2 Transmission separation by
7313 attribute.

7314 In FDP_ITT.4.2, the PP, PP-Module, functional package or ST author **should** specify the action to
7315 be taken by the TSF when an integrity error is encountered. An example might be that the TSF
7316 **should** request the resubmission of the user data. The SFP(s) specified in FDP_ITT.4.1 will be
7317 enforced as the actions are taken by the TSF.

7318 **F.11 Residual information protection (FDP_RIP)**

7319 **F.11.1 User application notes**

7320 Residual information protection ensures that TSF-controlled resources when de-allocated from
7321 an object and before they are reallocated to another object are treated by the TSF in a way that
7322 it is not possible to reconstruct all or part of the data contained in the resource before it was de-
7323 allocated.

7324 A TOE usually has a number of functions that potentially de-allocate resources from an object
7325 and potentially re-allocate those resources to objects. Some, but not all of those resources **may**
7326 have been used to store critical data from the previous use of the resource and for those
7327 resources FDP_RIP requires that they are prepared for reuse. Object reuse applies to explicit
7328 requests of a subject or user to release resources as well as implicit actions of the TSF that
7329 result in the de-allocation and subsequent re-allocation of resources to different objects.

7330 EXAMPLE

7331 Examples of explicit requests are the deletion or truncation of a file or the release of an area of main memory.
7332 Examples of implicit actions of the TSF are the de-allocation and re-allocation of cache regions.

7333 The requirement for object reuse is related to the content of the resource belonging to an
7334 object, not all information about the resource or object that **may** be stored elsewhere in the TSF.
7335 As an example, to satisfy the FDP_RIP requirement for files as objects requires that all sectors
7336 that make up the file need to be prepared for re-use.

7337 It also applies to resources that are serially reused by different subjects within the system. For
7338 example, most operating systems typically rely upon hardware registers (resources) to support

7339 processes within the system. As processes are swapped from a “run” state to a “sleep” state
 7340 (and vice versa), these registers are serially reused by different subjects. While this “swapping”
 7341 action **may** not be considered an allocation or deallocation of a resource, Residual information
 7342 protection (FDP_RIP) **could** apply to such events and resources.

7343 Residual information protection (FDP_RIP) typically controls access to information that is not
 7344 part of any currently defined or accessible object; however, in certain cases this **may** not be
 7345 true. For example, object “A” is a file and object “B” is the disk upon which that file resides. If
 7346 object “A” is deleted, the information from object “A” is under the control of Residual
 7347 information protection (FDP_RIP) even though it is still part of object “B”.

7348 It is important to note that Residual information protection (FDP_RIP) applies only to on-line
 7349 objects and not off-line objects such as those backed-up on tapes. For example, if a file is deleted
 7350 in the TOE, Residual information protection (FDP_RIP) **can** be instantiated to require that no
 7351 residual information exists upon deallocation; however, the TSF cannot extend this
 7352 enforcement to that same file that exists on the off-line back-up. Therefore, that same file is still
 7353 available. If this is a concern, then the PP, PP-Module, functional package or ST author **should**
 7354 make sure that the proper environmental objectives are in place to support operational user
 7355 guidance to address off-line objects.

7356 Residual information protection (FDP_RIP) and Rollback (FDP_ROL) **can** conflict when Residual
 7357 information protection (FDP_RIP) is instantiated to require that residual information be cleared
 7358 at the time the application releases the object to the TSF (i.e. upon deallocation). Therefore, the
 7359 Residual information protection (FDP_RIP) selection of “deallocation” **should** not be used with
 7360 Rollback (FDP_ROL) since there would be no information to roll back. The other selection,
 7361 “unavailability upon allocation”, **may** be used with Rollback (FDP_ROL), but there is the risk that
 7362 the resource which held the information has been allocated to a new object before the roll back
 7363 took place. If that were to occur, then the roll back would not be possible.

7364 There are no audit requirements in Residual information protection (FDP_RIP) because this is
 7365 not a user-invokable function. Auditing of allocated or deallocated resources would be auditable
 7366 as part of the access control SFP or the information flow control SFP operations.

7367 This family **should** apply to the objects specified in the access control SFP(s) or the information
 7368 flow control SFP(s) as specified by the PP, PP-Module, functional package or ST author.

7369 **F.11.2 FDP_RIP.1 Subset residual information protection**

7370 **F.11.2.1 Component rationale and application notes**

7371 This component requires that, for a subset of the objects in the TOE, the TSF will ensure that
 7372 there is no available residual information contained in a resource allocated to those objects or
 7373 deallocated from those objects.

7374 **F.11.2.2 Operations**

7375 In FDP_RIP.1.1, the PP, PP-Module, functional package or ST author **should** specify the event,
 7376 allocation of the resource to or deallocation of the resource from, that invokes the residual
 7377 information protection function.

7378 In FDP_RIP.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
 7379 objects subject to residual information protection.

7380 **F.11.3 FDP_RIP.2 Full residual information protection**

7381 **F.11.3.1 Component rationale and application notes**

7382 This component requires that for all objects in the TOE, the TSF will ensure that there is no
 7383 available residual information contained in a resource allocated to those objects or deallocated
 7384 from those objects.

7385 **F.11.3.2 Operations**

7386 In FDP_RIP.2.1, the PP, PP-Module, functional package or ST author **should** specify the event,
7387 allocation of the resource to or deallocation of the resource from, that invokes the residual
7388 information protection function.

7389 **F.12 Rollback (FDP_ROL)**

7390 **F.12.1 User application notes**

7391 This family addresses the need to return to a well-defined valid state, such as the need of a user
7392 to undo modifications to a file or to undo transactions in case of an incomplete series of
7393 transaction as in the case of databases.

7394 This family is intended to assist a user in returning to a well-defined valid state after the user
7395 undoes the last set of actions, or, in distributed databases, the return of all of the distributed
7396 copies of the databases to the state before an operation failed.

7397 Residual information protection (FDP_RIP) and Rollback (FDP_ROL) conflict when Residual
7398 information protection (FDP_RIP) enforces that the contents will be made unavailable at the
7399 time that a resource is deallocated from an object. Therefore, this use of Residual information
7400 protection (FDP_RIP) cannot be combined with Rollback (FDP_ROL) as there would be no
7401 information to roll back. Residual information protection (FDP_RIP) **can** be used only with
7402 Rollback (FDP_ROL) when it enforces that the contents will be unavailable at the time that a
7403 resource is allocated to an object. This is because the Rollback (FDP_ROL) mechanism will have
7404 an opportunity to access the previous information that **may** still be present in the TOE in order
7405 to successfully roll back the operation.

7406 The rollback requirement is bounded by certain limits.

7407 EXAMPLE

7408 For example, a text editor typically only allows you roll back up to a certain number of commands. Another example
7409 would be backups. If backup tapes are rotated, after a tape is reused, the information **can** no longer be retrieved. This
7410 also poses a bound on the rollback requirement.

7411 **F.12.2 FDP_ROL.1 Basic rollback**

7412 **F.12.2.1 Component rationale and application notes**

7413 This component allows a user or subject to undo a set of operations on a predefined set of
7414 objects. The undo is only possible within certain limits, for example up to a number of
7415 characters or up to a time limit.

7416 **F.12.2.2 Operations**

7417 In FDP_ROL.1.1, the PP, PP-Module, functional package or ST author **should** specify the access
7418 control SFP(s) and/or information flow control SFP(s) that will be enforced when performing
7419 rollback operations. This is necessary to make sure that roll back is not used to circumvent the
7420 specified SFPs.

7421 In FDP_ROL.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
7422 operations that **can** be rolled back.

7423 In FDP_ROL.1.1, the PP, PP-Module, functional package or ST author **should** specify the
7424 information and/or list of objects that are subjected to the rollback policy.

7425 In FDP_ROL.1.2, the PP, PP-Module, functional package or ST author **should** specify the
7426 boundary limit to which rollback operations **may** be performed. The boundary **may** be specified
7427 as a predefined period of time,

7428 EXAMPLE

7429 Operations **may** be undone which were performed within the past two minutes. Other possible boundaries **may** be
7430 defined as the maximum number of operations allowable or the size of a buffer.

7431 **F.12.3 FDP_ROL.2 Advanced rollback**

7432 **F.12.3.1 Component rationale and application notes**

7433 This component enforces that the TSF provide the capability to rollback all operations;
7434 however, the user **can** choose to rollback only a part of them.

7435 **F.12.3.2 Operations**

7436 In FDP_ROL.2.1, the PP, PP-Module, functional package or ST author **should** specify the access
7437 control SFP(s) and/or information flow control SFP(s) that will be enforced when performing
7438 rollback operations. This is necessary to make sure that roll back is not used to circumvent the
7439 specified SFPs.

7440 In FDP_ROL.2.1, the PP, PP-Module, functional package or ST author **should** specify the list of
7441 objects that are subjected to the rollback policy.

7442 In FDP_ROL.2.2, the PP, PP-Module, functional package or ST author **should** specify the
7443 boundary limit to which rollback operations **may** be performed. The boundary **may** be specified
7444 as a predefined period of time,

7445 EXAMPLE

7446 For example, operations **may** be undone which were performed within the past two minutes.

7447 Other possible boundaries **may** be defined as the maximum number of operations allowable or
7448 the size of a buffer.

7449 **F.13 Stored data confidentiality (FDP_SDC)**

7450 **F.13.1 User application notes**

7451 This family provides requirements that address protection of user data confidentiality while the
7452 data is stored within memory areas protected by the TSF. The TSF provides access to the data in
7453 the memory through the specified interfaces only and prevents compromise of their
7454 information bypassing these interfaces. It complements the family Stored data integrity
7455 (FDP_SDI) which protects the user data from integrity errors while being stored in the memory.

7456 **F.13.2 Evaluator notes**

7457 In practice, the dependency to FCS_COP.1 **may** be satisfied by a PP, PP-Module, functional
7458 package or ST author by providing a rationale explaining an alternative method to
7459 cryptography is used in some dedicated cases.

7460 **F.13.3 FDP_SDC.1 Stored data confidentiality**

7461 **F.13.3.1 Component rationale and application notes**

7462 In FDP_SDC.1 Stored data confidentiality, the PP, PP-Module, functional package or ST author
7463 specifies which user data is to be protected and in which type of memory the user data is
7464 requested to be protected. In the second selection the PP, PP-Module, functional package or ST
7465 author provides the memory type where the user data is to be protected.

7466 **F.13.3.2 Operations**

7467 In FDP_SDC.1.1 the PP, PP-Module, functional package or ST author **shall** select either “all user
7468 data” or provide a list of user data using the assignment below. In the second selection, the PP,
7469 PP-Module, functional package or ST author can specify either temporary memory, persistent
7470 memory or any memory. “Any memory” includes both temporary (volatile) and persistent (non-
7471 volatile) memory.

7472 In FDP_SDC.1.1 the PP, PP-Module, functional package or ST author provides a list of the user
7473 data that is to be protected in memory.

7474 **F.13.4 FDP_SDC.2 Stored data confidentiality with dedicated method**

7475 **F.13.4.1 Component rationale and application notes**

7476 FDP_SDC.2 Stored data confidentiality with dedicated method refines the FDP_SDC.1.1 element
7477 by allowing the PP, PP-Module, functional package or ST author to refine the list of user data
7478 using a variety of data characteristics.

7479 **F.13.4.2 Operations**

7480 The operations of selection and the first assignment are the same as that in FDP_SDC.1.

7481 For the second assignment the PP, PP-Module, functional package or ST author provides the
7482 data characteristics. Data characteristics can include items such as data length (shorter or
7483 longer than a threshold), data type (binary, text, image, sound, video), and data representation
7484 (binary, vector, character, frame).

7485 **F.14 Stored data integrity (FDP_SDI)**

7486 **F.14.1 User application notes**

7487 This family provides requirements that address protection of user data while it is stored within
7488 containers controlled by the TSF.

7489 Hardware glitches or errors **may** affect data stored in memory. This family provides
7490 requirements to detect these unintentional errors. The integrity of user data while stored on
7491 storage devices controlled by the TSF are also addressed by this family.

7492 To prevent a subject from modifying the data, the Information flow control functions (FDP_IFF)
7493 or Access control functions (FDP_ACF) families are required (rather than this family).

7494 This family differs from Internal TOE transfer (FDP_ITT) that protects the user data from
7495 integrity errors while being transferred within the TOE.

7496 **F.14.2 FDP_SDI.1 Stored data integrity monitoring**

7497 **F.14.2.1 Component rationale and application notes**

7498 This component monitors data stored on media for integrity errors. The PP, PP-Module,
7499 functional package or ST author **can** specify different kinds of user data attributes that will be
7500 used as the basis for monitoring.

7501 **F.14.2.2 Operations**

7502 In FDP_SDI.1.1, the PP, PP-Module, functional package or ST author **should** specify the integrity
7503 errors that the TSF will detect.

7504 In FDP_SDI.1.1, the PP, PP-Module, functional package or ST author **should** specify the user data
7505 attributes that will be used as the basis for the monitoring.

7506 **F.14.3 FDP_SDI.2 Stored data integrity monitoring and action**

7507 **F.14.3.1 Component rationale and application notes**

7508 This component monitors data stored on media for integrity errors. The PP, PP-Module,
7509 functional package or ST author **can** specify which action **should** be taken in case an integrity
7510 error is detected.

7511 **F.14.3.2 Operations**

7512 In FDP_SDI.2.1, the PP, PP-Module, functional package or ST author **should** specify the integrity
7513 errors that the TSF will detect.

7514 In FDP_SDI.2.1, the PP, PP-Module, functional package or ST author **should** specify the user data
7515 attributes that will be used as the basis for the monitoring.

7516 In FDP_SDI.2.2, the PP, PP-Module, functional package or ST author **should** specify the actions to
7517 be taken in case an integrity error is detected.

7518 **F.15 Inter-TSF user data confidentiality transfer protection (FDP_UCT)**

7519 **F.15.1 User application notes**

7520 This family defines the requirements for ensuring the confidentiality of user data when it is
7521 transferred using an external channel between the TOE and another trusted IT product.

7522 Confidentiality is enforced by preventing unauthorized disclosure of user data in transit
7523 between the two end points. The end points **may** be a TSF or a user.

7524 This family provides a requirement for the protection of user data during transit. In contrast,
7525 Confidentiality of exported TSF data (FPT_ITC) handles TSF data.

7526 **F.15.2 FDP_UCT.1 Basic data exchange confidentiality**

7527 **F.15.2.1 Component rationale and application notes**

7528 Depending on the access control or information flow policies the TSF is required to send or
7529 receive user data in a manner such that the confidentiality of the user data is protected.

7530 **F.15.2.2 Operations**

7531 In FDP_UCT.1.1, the PP, PP-Module, functional package or ST author **should** specify the access
7532 control SFP(s) and/or information flow control SFP(s) that will be enforced when exchanging
7533 user data. The specified policies will be enforced to make decisions about who **can** exchange
7534 data and which data **can** be exchanged.

7535 In FDP_UCT.1.1, the PP, PP-Module, functional package or ST author **should** specify whether this
7536 element applies to a mechanism that transmits or receives user data.

7537 **F.16 Inter-TSF user data integrity transfer protection (FDP_UIT)**

7538 **F.16.1 User application notes**

7539 This family defines the requirements for providing integrity for user data in transit between the
7540 TSF and another trusted IT product and recovering from detectable errors. At a minimum, this
7541 family monitors the integrity of user data for modifications. Furthermore, this family supports
7542 different ways of correcting detected integrity errors.

7543 This family defines the requirements for providing integrity for user data in transit; while
7544 Integrity of exported TSF data (FPT_ITI) handles TSF data.

7545 Inter-TSF user data integrity transfer protection (FDP_UIT) and Inter-TSF user data
7546 confidentiality transfer protection (FDP_UCT) are duals of each other, as Inter-TSF user data
7547 confidentiality transfer protection (FDP_UCT) addresses user data confidentiality. Therefore,
7548 the same mechanism that implements Inter-TSF user data integrity transfer protection
7549 (FDP_UIT) **could** possibly be used to implement other families such as Inter-TSF user data
7550 confidentiality transfer protection (FDP_UCT) and Import from outside of the TOE (FDP_ITC).

7551 **F.16.2 FDP_UIT.1 Data exchange integrity**

7552 **F.16.2.1 Component rationale and application notes**

7553 Depending on the access control or information flow policies the TSF is required to send or
7554 receive user data in a manner such that modification of the user data is detected. There is no
7555 requirement for a TSF mechanism to attempt to recover from the modification.

7556 **F.16.2.2 Operations**

7557 In FDP_UIT.1.1, the PP, PP-Module, functional package or ST author **should** specify the access
7558 control SFP(s) and/or information flow control SFP(s) that will be enforced on the transmitted
7559 data or on the received data. The specified policies will be enforced to make decisions about
7560 who **can** transmit or who **can** receive data, and which data **can** be transmitted or received.

7561 In FDP_UIT.1.1, the PP, PP-Module, functional package or ST author **should** specify whether this
7562 element applies to a TSF that is transmitting or receiving objects.

7563 In FDP_UIT.1.1, the PP, PP-Module, functional package or ST author **should** specify whether the
7564 data **should** be protected from modification, deletion, insertion, or replay.

7565 In FDP_UIT.1.2, the PP, PP-Module, functional package or ST author **should** specify whether the
7566 errors of the type: modification, deletion, insertion, or replay are detected.

7567 **F.16.3 FDP_UIT.2 Source data exchange recovery**

7568 **F.16.3.1 Component rationale and application notes**

7569 This component provides the ability to recover from a set of identified transmission errors, if
7570 required, with the help of the other trusted IT product. As the other trusted IT product is
7571 outside the TOE, the TSF cannot control its behaviour. However, it **can** provide functions that
7572 have the ability to cooperate with the other trusted IT product for the purposes of recovery.

7573 **EXAMPLE**

7574 For example, the TSF **could** include functions that depend upon the source trusted IT product to re-send the data in
7575 the event that an error is detected.

7576 This component deals with the ability of the TSF to handle such an error recovery.

7577 **F.16.3.2 Operations**

7578 In FDP_UIT.2.1, the PP, PP-Module, functional package or ST author **should** specify the access
7579 control SFP(s) and/or information flow control SFP(s) that will be enforced when recovering
7580 user data. The specified policies will be enforced to make decisions about which data **can** be
7581 recovered and how it **can** be recovered.

7582 In FDP_UIT.2.1, the PP, PP-Module, functional package or ST author **should** specify the list of
7583 integrity errors from which the TSF, with the help of the source trusted IT product, is be able to
7584 recover the original user data.

7585 **F.16.4 FDP_UIT.3 Destination data exchange recovery**

7586 **F.16.4.1 Component rationale and application notes**

7587 This component provides the ability to recover from a set of identified transmission errors. It
7588 accomplishes this task without help from the source trusted IT product. For example, if certain
7589 errors are detected, the transmission protocol must be robust enough to allow the TSF to
7590 recover from the error based on checksums and other information available within that
7591 protocol.

7592 **F.16.4.2 Operations**

7593 In FDP_UIT.3.1, the PP, PP-Module, functional package or ST author **should** specify the access
7594 control SFP(s) and/or information flow control SFP(s) that will be enforced when recovering
7595 user data. The specified policies will be enforced to make decisions about which data **can** be
7596 recovered and how it **can** be recovered.

7597 In FDP_UIT.3.1, the PP, PP-Module, functional package or ST author **should** specify the list of
7598 integrity errors from which the receiving TSF, alone, is able to recover the original user data.

7599
7600
7601
7602

Annex G (normative)

Class FIA: Identification and authentication- application notes

7603 G.1 General Information

7604 A common security requirement is to unambiguously identify the person and/or entity
7605 performing functions in a TOE. This involves not only establishing the claimed identity of each
7606 user, but also verifying that each user is indeed who he/she claims to be. This is achieved by
7607 requiring users to provide the TSF with some information that is known by the TSF to be
7608 associated with the user in question.

7609 Families in this class address the requirements for functions to establish and verify a claimed
7610 user identity. Identification and Authentication is required to ensure that users are associated
7611 with the proper security attributes.

7612 EXAMPLE

7613 Security attributes include identity, groups, roles, security, or integrity levels.

7614 The unambiguous identification of authorized users and the correct association of security
7615 attributes with users and subjects is critical to the enforcement of the security policies.

7616 The Authentication failures (FIA_AFL) family addresses defining limits on repeated
7617 unsuccessful authentication attempts.

7618 The Authentication proof of identity (FIA_API) family addresses defining the functionality
7619 provided by the TOE to prove its identity and to be verified by an external entity in the TOE IT
7620 environment.

7621 The User attribute definition (FIA_ATD) family addresses the definition of user attributes that
7622 are used in the enforcement of the SFRs.

7623 The Specification of secrets (FIA_SOS) family addresses the generation and verification of
7624 secrets that satisfy a defined metric.

7625 The User authentication (FIA_UAU) family addresses verifying the identity of a user.

7626 The User identification (FIA_UID) family addresses determining the identity of a user.

7627 The User-subject binding (FIA_USB) family addresses the correct association of security
7628 attributes for each authorized user.

7629 G.2 Authentication failures (FIA_AFL)

7630 G.2.1 User application notes

7631 This family addresses requirements for defining values for authentication attempts and TSF
7632 actions in cases of authentication attempt failure. Parameters include, but are not limited to, the
7633 number of attempts and time thresholds.

7634 The session establishment process is the interaction with the user to perform the session
7635 establishment independent of the actual implementation. If the number of unsuccessful
7636 authentication attempts exceeds the indicated threshold, either the user account or the terminal
7637 (or both) will be locked. If the user account is disabled, the user cannot log-on to the system. If
7638 the terminal is disabled, the terminal (or the address that the terminal has) cannot be used for
7639 any log-on. Both of these situations continue until the condition for re-establishment is
7640 satisfied.

7641 G.2.2 FIA_AFL.1 Authentication failure handling

7642 **G.2.2.1 Component rationale and application notes**

7643 The PP, PP-Module, functional package or ST author **may** define the number of unsuccessful
7644 authentication attempts or **may** choose to let the TOE developer or the authorized user to
7645 define this number. The unsuccessful authentication attempts need not be consecutive, but
7646 rather related to an authentication event. Such an authentication event **could** be the count from
7647 the last successful session establishment at a given terminal.

7648 The PP, PP-Module, functional package or ST author **could** specify a list of actions that the TSF
7649 **shall** take in the case of authentication failure. An authorized administrator **could** also be
7650 allowed to manage the events, if deemed opportune by the PP, PP-Module, functional package
7651 or ST author. These actions **could** be, among other things, terminal deactivation, user account
7652 deactivation, or administrator alarm. The conditions under which the situation will be restored
7653 to normal must be specified on the action.

7654 In order to prevent denial of service, TOEs usually ensure that there is at least one user account
7655 that cannot be disabled.

7656 Further actions for the TSF **can** be stated by the PP, PP-Module, functional package or ST author,
7657 including rules for re-enabling the user session establishment process, or sending an alarm to
7658 the administrator.

7659 EXAMPLE

7660 Examples of these actions are: until a specified time has lapsed, until the authorized administrator re-enables the
7661 terminal/account, a time related to failed previous attempts (every time the attempt fails, the disabling time is
7662 doubled).

7663 **G.2.2.2 Operations**

7664 In FIA_AFL.1 Authentication failure handling, the PP, PP-Module, functional package or ST
7665 author **should** select either the assignment of a positive integer, or the phrase “an administrator
7666 configurable positive integer” specifying the range of acceptable values.

7667 In FIA_AFL.1 Authentication failure handling, the PP, PP-Module, functional package or ST
7668 author **should** specify the authentication events. Examples of these authentication events are:
7669 the unsuccessful authentication attempts since the last successful authentication for the
7670 indicated user identity, the unsuccessful authentication attempts since the last successful
7671 authentication for the current terminal, the number of unsuccessful authentication attempts in
7672 the last 10 minutes. At least one authentication event must be specified.

7673 In FIA_AFL.1 Authentication failure handling, if the assignment of a positive integer is selected,
7674 the PP, PP-Module, functional package or ST author **should** specify the default number (positive
7675 integer) of unsuccessful authentication attempts that, when met or surpassed, will trigger the
7676 events.

7677 In FIA_AFL.1 Authentication failure handling, if an administrator configurable positive integer is
7678 selected, the PP, PP-Module, functional package or ST author **should** specify the range of
7679 acceptable values from which the administrator of the TOE **may** configure the number of
7680 unsuccessful authentication attempts. The number of authentication attempts **should** be less
7681 than or equal to the upper bound and greater or equal to the lower bound values.

7682 In FIA_AFL.1.2, the PP, PP-Module, functional package or ST author **should** select whether the
7683 event of meeting or surpassing the defined number of unsuccessful authentication attempts
7684 **shall** trigger an action by the TSF.

7685 In FIA_AFL.1.2, the PP, PP-Module, functional package or ST author **should** specify the actions to
7686 be taken in case the threshold is met or surpassed, as selected. These actions **could** be disabling
7687 of an account for 5 minutes, disabling the terminal for an increasing amount of time (2 to the
7688 power of the number of unsuccessful attempts in seconds), or disabling of the account until
7689 unlocked by the administrator and simultaneously informing the administrator. The actions
7690 **should** specify the measures and if applicable the duration of the measure (or the conditions
7691 under which the measure will be ended).

7692 **G.3 Authentication proof of identity (FIA_API)**

7693 **G.3.1 User application notes**

7694 The other families of the Class FIA describe only the authentication verification of users'
7695 identity performed by the TOE and do not describe the functionality of the user to prove their
7696 identity. The family FIA_API allows the specification the functionality allowing a TOE to prove
7697 its own identity.

7698 **G.3.2 FIA_API.1 Authentication proof of identity**

7699 **Editors' Note:**

7700 **Since no contributions were received for this text, the editors have proposed the text below.**
7701 **The editors request that SMEs review carefully.**

7702 **If no comments are received during the commenting period for this draft, the editors' proposal**
7703 **will be accepted in the next draft.**

7704 **G.3.2.1 Component rationale and application notes**

7705 FIA_API.1 Authentication proof of identity allows the specification of the authentication
7706 mechanism used to support proving the identity of the TOE to external entities.

7707 **G.3.2.2 Operations**

7708 The first assignment is where a PP, PP-Module, functional package or ST author specifies the
7709 authentication mechanism to be used.

7710 EXAMPLE

7711 Examples of such an authentication method is "an Authentication Mechanism based on Triple-DES" and "Chip
7712 Authentication Protocol according to TR-03110"

7713 The second assignment allows the PP, PP-Module, functional package or ST author to specify to
7714 what the proof of identity is associated with. This can be an object, authorized user or a role.

7715 **Editors' Note:**

7716 **Editors observe that in many STs using this component the second completion is for "TOE"**
7717 **which is neither an object, authorized user or a role. Should FIA_API.1.1 be updated to allow for**
7718 **the specification of TOE in the assignment?**

7719 **The editors' welcome comments on this point. If no comments are received then then text will**
7720 **remain unaltered in the next draft.**

7721 **G.4 User attribute definition (FIA_ATD)**

7722 **G.4.1 User application notes**

7723 All authorized users **may** have a set of security attributes, other than the user's identity, that are
7724 used to enforce the SFRs. This family defines the requirements for associating user security
7725 attributes with users as needed to support the TSF in making security decisions.

7726 There are dependencies on the individual security policy (SFP) definitions. These individual
7727 definitions **should** contain the listing of attributes that are necessary for policy enforcement.

7728 **G.4.2 FIA_ATD.1 User attribute definition**

7729 **G.4.2.1 Component rationale and application notes**

7730 This component specifies the security attributes that **should** be maintained at the level of the
7731 user. This means that the security attributes listed are assigned to and **can** be changed at the
7732 level of the user. In other words, changing a security attribute in this list associated with a user
7733 **should** have no impact on the security attributes of any other user.

7734 In case security attributes belong to a group of users (such as Capability List for a group), the
7735 user will need to have a reference (as security attribute) to the relevant group.

7736 **G.4.2.2 Operations**

7737 In FIA_ATD.1.1, the PP, PP-Module, functional package or ST author **should** specify the security
7738 attributes that are associated to an individual user.

7739 EXAMPLE

7740 An example of such a list is {"clearance", "group identifier", "rights"}.

7741 **G.5 Specification of secrets (FIA_SOS)**

7742 **G.5.1 User application notes**

7743 This family defines requirements for mechanisms that enforce defined quality metrics on
7744 provided secrets and generate secrets to satisfy the defined metric. Examples of such
7745 mechanisms **may** include automated checking of user supplied passwords, or automated
7746 password generation.

7747 A secret **can** be generated outside the TOE.

7748 EXAMPLE

7749 An example of a secret generated outside of the TOE could be one that is selected by the user and introduced in the
7750 TOE.

7751 In such cases, the FIA_SOS.1 Verification of secrets component **can** be used to ensure that the
7752 external generated secret adheres to certain standards, for example a minimum size, not
7753 present in a dictionary, and/or not previously used.

7754 Secrets **can** also be generated by the TOE. In those cases, the FIA_SOS.2 TSF Generation of
7755 secrets component **can** be used to require the TOE to ensure that the secrets that will adhere to
7756 some specified metrics.

7757 Secrets contain the authentication data provided by the user for an authentication mechanism
7758 that is based on knowledge the user possesses. When cryptographic keys are employed, the
7759 class FCS: Cryptographic support **should** be used instead of this family.

7760 **G.5.2 FIA_SOS.1 Verification of secrets**

7761 **G.5.2.1 Component rationale and application notes**

7762 Secrets **can** be generated by the user. This component ensures that those user generated secrets
7763 **can** be verified to meet a certain quality metric.

7764 **G.5.2.2 Operations**

7765 In FIA_SOS.1.1, the PP, PP-Module, functional package or ST author **should** provide a defined
7766 quality metric. The quality metric specification **can** be as simple as a description of the quality
7767 checks to be performed, or as formal as a reference to a government published standard that
7768 defines the quality metrics that secrets must meet.

7769 EXAMPLE

7770 quality metrics **could** include a description of the alphanumeric structure of acceptable secrets and/or the space size
7771 that acceptable secrets must meet.

7772 **G.5.3 FIA_SOS.2 TSF Generation of secrets**

7773 **G.5.3.1 Component rationale and application notes**

7774 This component allows the TSF to generate secrets for specific functions such as authentication
7775 by means of passwords.

7776 When a pseudo-random number generator is used in a secret generation algorithm, it **should**
7777 accept as input random data that would provide output that has a high degree of

7778 unpredictability. This random data (seed) **can** be derived from a number of available
 7779 parameters such as a system clock, system registers, date, time, etc. The parameters **should** be
 7780 selected to ensure that the number of unique seeds that **can** be generated from these inputs
 7781 **should** be at least equal to the minimum number of secrets that must be generated.

7782 **G.5.3.2 Operations**

7783 In FIA_SOS.2.1, the PP, PP-Module, functional package or ST author **should** provide a defined
 7784 quality metric. The quality metric specification **can** be as simple as a description of the quality
 7785 checks to be performed or as formal as a reference to a government published standard that
 7786 defines the quality metrics that secrets must meet.

7787 EXAMPLE

7788 quality metrics **could** include a description of the alphanumeric structure of acceptable secrets and/or the space size
 7789 that acceptable secrets must meet.

7790 In FIA_SOS.2.2, the PP, PP-Module, functional package or ST author **should** provide a list of TSF
 7791 functions for which the TSF generated secrets must be used. An example of such a function
 7792 **could** include a password-based authentication mechanism.

7793 **G.6 User authentication (FIA_UAU)**

7794 **G.6.1 User application notes**

7795 This family defines the types of user authentication mechanisms supported by the TSF. This
 7796 family defines the required attributes on which the user authentication mechanisms must be
 7797 based.

7798 **G.6.2 FIA_UAU.1 Timing of authentication**

7799 **G.6.2.1 Component rationale and application notes**

7800 This component requires that the PP, PP-Module, functional package or ST author define the
 7801 TSF-mediated actions that **can** be performed by the TSF on behalf of the user before the claimed
 7802 identity of the user is authenticated. The TSF-mediated actions **should** have no security
 7803 concerns with users incorrectly identifying themselves prior to being authenticated. For all
 7804 other TSF-mediated actions not in the list, the user must be authenticated before the action **can**
 7805 be performed by the TSF on behalf of the user.

7806 This component cannot control whether the actions **can** also be performed before the
 7807 identification took place. This requires the use of either FIA_UID.1 Timing of identification or
 7808 FIA_UID.2 User identification before any action with the appropriate assignments.

7809 **G.6.2.2 Operations**

7810 In FIA_UAU.1.1, the PP, PP-Module, functional package or ST author **should** specify a list of TSF-
 7811 mediated actions that **can** be performed by the TSF on behalf of a user before the claimed
 7812 identity of the user is authenticated. This list cannot be empty. If no actions are appropriate,
 7813 component FIA_UAU.2 User authentication before any action **should** be used instead.

7814 EXAMPLE

7815 Such an action might include the request for help on the login procedure.

7816 **G.6.3 FIA_UAU.2 User authentication before any action**

7817 **G.6.3.1 Component rationale and application notes**

7818 This component requires that a user is authenticated before any other TSF-mediated action **can**
 7819 take place on behalf of that user.

7820 **G.6.4 FIA_UAU.3 Unforgeable authentication**

7821 **G.6.4.1 Component rationale and application notes**

7822 This component addresses requirements for mechanisms that provide protection of
7823 authentication data. Authentication data that is copied from another user, or is in some way
7824 constructed **should** be detected and/or rejected. These mechanisms provide confidence that
7825 users authenticated by the TSF are actually who they claim to be.

7826 This component **may** be useful only with authentication mechanisms that are based on
7827 authentication data that cannot be shared. It is impossible for a TSF to detect or prevent the
7828 sharing of passwords outside the control of the TSF.

7829 **EXAMPLE**

7830 *An example of authentication data that cannot be shared is biometrics*

7831 **Editors' Note**

7832 **Is this a good example? Editors' suggest that replay attacks could be "sharing" biometrics.**

7833 **If no comments are received, then this example will be deleted in the next draft**

7834 **G.6.4.2 Operations**

7835 In FIA_UAU.3.1, the PP, PP-Module, functional package or ST author **should** specify whether the
7836 TSF will detect, prevent, or detect and prevent forging of authentication data.

7837 In FIA_UAU.3.2, the PP, PP-Module, functional package or ST author **should** specify whether the
7838 TSF will detect, prevent, or detect and prevent copying of authentication data.

7839 **G.6.5 FIA_UAU.4 Single-use authentication mechanisms**

7840 **G.6.5.1 Component rationale and application notes**

7841 This component addresses requirements for authentication mechanisms based on single-use
7842 authentication data. Single-use authentication data **can** be something the user has or knows, but
7843 not something the user is.

7844 **EXAMPLE**

7845 Single-use authentication data include single-use passwords, encrypted time-stamps, and/or random numbers from
7846 a secret lookup table.

7847 The PP, PP-Module, functional package or ST author **can** specify to which authentication
7848 mechanism(s) this requirement applies.

7849 **G.6.5.2 Operations**

7850 In FIA_UAU.4.1, the PP, PP-Module, functional package or ST author **should** specify the list of
7851 authentication mechanisms to which this requirement applies. This assignment **can** be "all
7852 authentication mechanisms". An example of this assignment **could** be "the authentication
7853 mechanism employed to authenticate people on the external network".

7854 **G.6.6 FIA_UAU.5 Multiple authentication mechanisms**

7855 **G.6.6.1 Component rationale and application notes**

7856 The use of this component allows specification of requirements for more than one
7857 authentication mechanism to be used within a TOE. For each distinct mechanism, applicable
7858 requirements must be chosen from the FIA: Identification and authentication class to be applied
7859 to each mechanism. It is possible that the same component **could** be selected multiple times in
7860 order to reflect different requirements for the different use of the authentication mechanism.

7861 The management functions in the class FMT **may** provide maintenance capabilities for the set of
7862 authentication mechanisms, as well as the rules that determine whether the authentication was
7863 successful.

7864 To allow anonymous users to interact with the TOE, a "none" authentication mechanism **can** be
7865 incorporated. The use of such access **should** be clearly explained in the rules of FIA_UAU.5.2.

7866 **G.6.6.2 Operations**

7867 In FIA_UAU.5.1, the PP, PP-Module, functional package or ST author **should** define the available
7868 authentication mechanisms.

7869 EXAMPLE 1

7870 Such a list **could** be: “none, password mechanism, biometric (retinal scan), S/key mechanism”.

7871 In FIA_UAU.5.2, the PP, PP-Module, functional package or ST author **should** specify the rules
7872 that describe how the authentication mechanisms provide authentication and when each is to
7873 be used. This means that for each situation the set of mechanisms that might be used for
7874 authenticating the user must be described.

7875 EXAMPLE 2

7876 A list of such rules is: “if the user has special privileges a password mechanism and a biometric mechanism both **shall**
7877 be used, with success only if both succeed; for all other users a password mechanism **shall** be used.”

7878 The PP, PP-Module, functional package or ST author might give the boundaries within which the
7879 authorized administrator **may** specify specific rules. An example of a rule is: “the user **shall**
7880 always be authenticated by means of a token; the administrator might specify additional
7881 authentication mechanisms that also must be used.” The PP, PP-Module, functional package or
7882 ST author also might choose not to specify any boundaries but leave the authentication
7883 mechanisms and their rules completely up to the authorized administrator.

7884 **G.6.7 FIA_UAU.6 Re-authenticating**

7885 **G.6.7.1 Component rationale and application notes**

7886 This component addresses potential needs to re-authenticate users at defined points in time.
7887 These **may** include user requests for the TSF to perform security relevant actions, as well as
7888 requests from non-TSF entities for re-authentication.

7889 EXAMPLE

7890 A server application requesting that the TSF re-authenticate the client it is serving.

7891 **G.6.7.2 Operations**

7892 In FIA_UAU.6.1, the PP, PP-Module, functional package or ST author **should** specify the list of
7893 conditions requiring re-authentication. This list **could** include a specified user inactivity period
7894 that has elapsed, the user requesting a change in active security attributes, or the user
7895 requesting the TSF to perform some security critical function.

7896 The PP, PP-Module, functional package or ST author might give the boundaries within which the
7897 re-authentication **should** occur and leave the specifics to the authorized administrator.

7898 EXAMPLE

7899 “the user **shall** always be re-authenticated at least once a day; the administrator might specify that the re-
7900 authentication **should** happen more often but not more often than once every 10 minutes.”

7901 **G.6.8 FIA_UAU.7 Protected authentication feedback**

7902 **G.6.8.1 Component rationale and application notes**

7903 This component addresses the feedback on the authentication process that will be provided to
7904 the user. In some systems, the feedback consists of indicating how many characters have been
7905 typed but not showing the characters themselves, in other systems even this information might
7906 not be appropriate.

7907 This component requires that the authentication data is not provided as-is back to the user. In a
7908 workstation environment, it **could** display a “dummy” for each password character provided,
7909 and not the original character.

7910 Example

7911 A “dummy” **could** be a star “*” character.

7912 **G.6.8.2 Operations**

7913 In FIA_UAU.7 Protected authentication feedback, the PP, PP-Module, functional package or ST
7914 author **should** specify the feedback related to the authentication process that will be provided
7915 to the user.

7916 EXAMPLE

7917 A feedback assignment **could** be “the number of characters typed”, another type of feedback is “the authentication
7918 mechanism that failed the authentication”.

7919 **G.7 User identification (FIA_UID)**

7920 **G.7.1 User application notes**

7921 This family defines the conditions under which users are required to identify themselves before
7922 performing any other actions that are to be mediated by the TSF and that require user
7923 identification.

7924 **G.7.2 FIA_UID.1 Timing of identification**

7925 **G.7.2.1 Component rationale and application notes**

7926 This component poses requirements for the user to be identified. The PP, PP-Module, functional
7927 package or ST author **can** indicate specific actions that **can** be performed before the
7928 identification takes place.

7929 If FIA_UID.1 Timing of identification is used, the TSF-mediated actions mentioned in FIA_UID.1
7930 Timing of identification **should** also appear in this FIA_UAU.1 Timing of authentication.

7931 **G.7.2.2 Operations**

7932 In FIA_UID.1.1, the PP, PP-Module, functional package or ST author **should** specify a list of TSF-
7933 mediated actions that **can** be performed by the TSF on behalf of a user before the user has to
7934 identify itself. If no actions are appropriate, component FIA_UID.2 User identification before any
7935 action **should** be used instead. An example of such an action might include the request for help
7936 on the login procedure.

7937 **G.7.3 FIA_UID.2 User identification before any action**

7938 **G.7.3.1 Component rationale and application notes**

7939 In this component users will be identified. A user is not allowed by the TSF to perform any
7940 action before being identified.

7941 **G.8 User-subject binding (FIA_USB)**

7942 **G.8.1 User application notes**

7943 An authenticated user, in order to use the TOE, typically activates a subject. The user's security
7944 attributes are associated (totally or partially) with this subject. This family defines
7945 requirements to create and maintain the association of the user's security attributes to a subject
7946 acting on the user's behalf.

7947 **G.8.2 FIA_USB.1 User-subject binding**

7948 **G.8.2.1 Component rationale and application notes**

7949 It is intended that a subject is acting on behalf of the user who caused the subject to come into
7950 being or to be activated to perform a certain task.

7951 Therefore, when a subject is created, that subject is acting on behalf of the user who initiated
7952 the creation. In cases where anonymity is used, the subject is still acting on behalf of a user, but

7953 the identity of that user is unknown. A special category of subjects is those subjects that serve
7954 multiple users. In such cases the user that created this subject is assumed to be the “owner”.

7955 EXAMPLE

7956 An example of a user is a server process.

7957 **G.8.2.2 Operations**

7958 In FIA_USB.1.1, the PP, PP-Module, functional package or ST author **should** specify a list of the
7959 user security attributes that are to be bound to subjects.

7960 In FIA_USB.1.2, the PP, PP-Module, functional package or ST author **should** specify any rules
7961 that are to apply upon initial association of attributes with subjects, or “none”.

7962 In FIA_USB.1.3, the PP, PP-Module, functional package or ST author **should** specify any rules
7963 that are to apply when changes are made to the user security attributes associated with
7964 subjects acting on behalf of users, or “none”.

Annex H (normative)

Class FMT: Security management- application notes

7969 H.1 General information

7970 This class specifies the management of several aspects of the TSF: security attributes, TSF data
7971 and functions in the TSF. The different management roles and their interaction, such as
7972 separation of capability, **can** also be specified.

7973 In an environment where the TOE is made up of multiple physically separated parts, the timing
7974 issues with respect to propagation of security attributes, TSF data, and function modification
7975 become very complex, especially if the information is required to be replicated across the parts
7976 of the TOE. This **should** be considered when selecting components such as FMT_REV.1
7977 Revocation, or FMT_SAE.1 Time-limited authorization, where the behaviour might be impaired.
7978 In such situations, use of components from Internal TOE TSF data replication consistency
7979 (FPT_TRC) is advisable.

7980 The FMT_LIM family provides requirements that allow the specification of a policy that limits
7981 the capabilities and the availability of TSF functions. This is useful when a PP, PP-Module,
7982 functional package or ST author needs to enforce design principles such as least privilege and
7983 attack surface minimization.

7984 Note These, and other architectural and design principles along with appropriate evaluation considerations
7985 are discussed in ISO/IEC 19249, Information technology — Security techniques — Catalogue of architectural and
7986 design principles for secure products, systems, and applications.

7987 H.2 Limited capabilities and availability (FMT_LIM)

7988 H.2.1 User application notes

7989 The functional requirements FMT_LIM.1 and FMT_LIM.2 assume that there are two types of
7990 mechanisms (limitation of capabilities and limitation of availability) which together **shall**
7991 provide protection in order to enforce the policy. This also allows that

- 7992 a) the TSF is provided without restrictions in the product in its user environment but its
7993 capabilities are so limited that the policy is enforced or conversely
- 7994 b) the TSF is designed with high functionality but is removed or disabled in the product in
7995 its user environment.

7996 The combination of both requirements **shall** enforce the policy.

7997 H.2.2 FMT_LIM.1 Limited capabilities

7998 H.2.2.1 Component rationale and application notes

7999 An example of a limited capability JTAG interface enablement, which **could** be either enabled or
8000 disabled.

8001 H.2.2.2 Operations

8002 In FMT_LIM.1.1, the PP, PP-Module, functional package or ST author **should** specify the limited
8003 capability policy.

8004 H.2.3 FMT_LIM.2 Limited availability

8005 H.2.3.1 Component rationale and application notes

8006 EXAMPLE

8007 An example of a limited availability is JTAG interface enablement, which could be either enabled or disabled before
8008 operational use of the TOE.

8009 H.2.3.2 Operations

8010 In FMT_LIM.2.1, the PP, PP-Module, functional package or ST author should specify the limited
8011 availability policy.

8012 H.3 Management of functions in TSF (FMT_MOF)

8013 H.3.1 User application notes

8014 The TSF management functions enable authorized users to set up and control the secure
8015 operation of the TOE. These administrative functions typically fall into a number of different
8016 categories:

8017 a) Management functions that relate to access control, accountability and
8018 authentication controls enforced by the TOE. For example, definition and update of
8019 user security characteristics or definition and update of auditing system controls,
8020 definition and update of per-user policy attributes, definition of known system
8021 access control labels, and control and management of user groups.

8022 EXAMPLE

8023 User security characteristics: unique identifiers associated with user names, user accounts, system
8024 entry parameters

8025 Auditing system controls: selection of audit events, management of audit trails, audit trail analysis,
8026 and audit report generation

8027 User policy attributes: user clearance

8028 b) Management functions that relate to controls over availability. For example,
8029 definition and update of availability parameters or resource quotas.

8030 c) Management functions that relate to general installation and configuration. For
8031 example, TOE configuration, manual recovery, installation of TOE security fixes (if
8032 any), repair and reinstallation of hardware.

8033 d) Management functions that relate to routine control and maintenance of TOE
8034 resources. For example, enabling and disabling peripheral devices, mounting of
8035 removable storage media, backup, and recovery.

8036 NOTE These functions need to be present in a TOE based on the families included in the PP or ST. It is the
8037 responsibility of the PP, PP-Module, functional package or ST author to ensure that adequate functions will be
8038 provided to manage the TOE in a secure fashion.

8039 The TSF might contain functions that **can** be controlled by an administrator. For example, the
8040 auditing functions **could** be switched off, the time synchronization **could** be switchable, and/or
8041 the authentication mechanism **could** be modifiable.

8042 H.3.2 FMT_MOF.1 Management of security functions behaviour

8043 H.3.2.1 Component rationale and application notes

8044 This component allows identified roles to manage the security functions of the TSF. This might
8045 entail obtaining the current status of a security function, disabling, or enabling the security
8046 function, or modifying the behaviour of the security function.

8047 EXAMPLE

8048 modifying the behaviour of the security functions is changing of authentication mechanisms.

8049 H.3.2.2 Operations

8050 In FMT_MOF.1.1, the PP, PP-Module, functional package or ST author **should** select whether the
8051 role **can** determine the behaviour of, disable, enable, and/or modify the behaviour of the
8052 security functions.

8053 In FMT_MOF.1.1, the PP, PP-Module, functional package or ST author **should** specify the
8054 functions that **can** be modified by the identified roles. Examples include auditing and time
8055 determination.

8056 In FMT_MOF.1.1, the PP, PP-Module, functional package or ST author **should** specify the roles
8057 that are allowed to modify the functions in the TSF. The possible roles are specified in
8058 FMT_SMR.1 Security roles.

8059 **H.4 Management of security attributes (FMT_MSA)**

8060 **H.4.1 User application notes**

8061 This family defines the requirements on the management of security attributes.

8062 Security attributes affect the behaviour of the TSF.

8063 EXAMPLE

8064 Examples of security attributes are the groups to which a user belongs, the roles he/she might assume, the priority of
8065 a process (subject), and the rights belonging to a role or a user.

8066 These security attributes might need to be managed by the user, a subject, a specific authorized
8067 user (a user with explicitly given rights for this management) or inherit values according to a
8068 given policy/set of rules.

8069 It is noted that the right to assign rights to users is itself a security attribute and/or potentially
8070 subject to management by FMT_MSA.1 Management of security attributes.

8071 FMT_MSA.2 Secure security attributes **can** be used to ensure that any accepted combination of
8072 security attributes is within a secure state. The definition of what “secure” means is left to the
8073 TOE guidance.

8074 In some instances, subjects, objects, or user accounts are created. If no explicit values for the
8075 related security attributes are given, default values need to be used. FMT_MSA.1 Management of
8076 security attributes **can** be used to specify that these default values **can** be managed.

8077 **H.4.2 FMT_MSA.1 Management of security attributes**

8078 **H.4.2.1 Component rationale and application notes**

8079 This component allows users acting in certain roles to manage identified security attributes.
8080 The users are assigned to a role within the component FMT_SMR.1 Security roles.

8081 The default value of a parameter is the value the parameter takes when it is instantiated
8082 without specifically assigned values. An initial value is provided during the instantiation
8083 (creation) of a parameter and overrides the default value.

8084 **H.4.2.2 Operations**

8085 In FMT_MSA.1.1, the PP, PP-Module, functional package or ST author **should** list the access
8086 control SFP(s) or the information flow control SFP(s) for which the security attributes are
8087 applicable.

8088 In FMT_MSA.1.1, the PP, PP-Module, functional package or ST author **should** specify the
8089 operations that **can** be applied to the identified security attributes. The PP, PP-Module,
8090 functional package or ST author **can** specify that the role **can** modify the default value
8091 (change_default), query, modify the security attribute, delete the security attributes entirely or
8092 define their own operation.

8093 In FMT_MSA.1.1, the PP, PP-Module, functional package or ST author **should** specify the security
8094 attributes that **can** be operated on by the identified roles. It is possible for the PP, PP-Module,
8095 functional package or ST author to specify that the default value such as default access-rights
8096 **can** be managed.

8097 EXAMPLE 1

- 8098 Examples of these security attributes are user-clearance, priority of service level, access control list, default access
8099 rights.
- 8100 In FMT_MSA.1.1, the PP, PP-Module, functional package or ST author **should** specify the roles
8101 that are allowed to operate on the security attributes. The possible roles are specified in
8102 FMT_SMR.1 Security roles.
- 8103 In FMT_MSA.1.1, if selected, the PP, PP-Module, functional package or ST author **should** specify
8104 which other operations the role **could** perform.
- 8105 EXAMPLE 2
- 8106 An example of such an operation **could** be “create”.
- 8107 **H.4.3 FMT_MSA.2 Secure security attributes**
- 8108 **H.4.3.1 Component rationale and application notes**
- 8109 This component contains requirements on the values that **can** be assigned to security attributes.
8110 The assigned values **should** be such that the TOE will remain in a secure state.
- 8111 The definition of what “secure” means is not answered in this component but is left to the
8112 development of the TOE and the resulting information in the guidance. An example **could** be
8113 that if a user account is created, it **should** have a non-trivial password.
- 8114 **H.4.3.2 Operations**
- 8115 In FMT_MSA.2.1, the PP, PP-Module, functional package or ST author **should** specify the list of
8116 security attributes that require only secure values to be provided.
- 8117 **H.4.4 FMT_MSA.3 Static attribute initialization**
- 8118 **H.4.4.1 Component rationale and application notes**
- 8119 This component requires that the TSF provide default values for relevant object security
8120 attributes, which **can** be overridden by an initial value. It **may** still be possible for a new object
8121 to have different security attributes at creation if a mechanism exists to specify the permissions
8122 at time of creation.
- 8123 **H.4.4.2 Operations**
- 8124 In FMT_MSA.3.1, the PP, PP-Module, functional package or ST author **should** list the access
8125 control SFP or the information flow control SFP for which the security attributes are applicable.
- 8126 In FMT_MSA.3.1, the PP, PP-Module, functional package or ST author **should** select whether the
8127 default property of the access control attribute will be restrictive, permissive, or another
8128 property. Only one of these options **may** be chosen.
- 8129 In FMT_MSA.3.1, if the PP, PP-Module, functional package or ST author selects another property,
8130 the PP, PP-Module, functional package or ST author **should** specify the desired characteristics of
8131 the default values.
- 8132 In FMT_MSA.3.2, the PP, PP-Module, functional package or ST author **should** specify the roles
8133 that are allowed to modify the values of the security attributes. The possible roles are specified
8134 in FMT_SMR.1 Security roles.
- 8135 **H.4.5 FMT_MSA.4 Security attribute value inheritance**
- 8136 **H.4.5.1 Component rationale and application notes**
- 8137 This component requires specification of the set of rules through which the security attribute
8138 inherits values and the conditions to be met for these rules to be applied.
- 8139 **H.4.5.2 Operations**

8140 In FMT_MSA.4.1, the PP, PP-Module, functional package or ST author specifies the rules
8141 governing the value that will be inherited by the specified security attribute, including the
8142 conditions that are to be met for the rules to be applied.

8143 EXAMPLE

8144 For example, if a new file or directory is created (in a multilevel filesystem), its label is the label at which the user is
8145 logged in at the time it is created.

8146 **H.5 Management of TSF data (FMT_MTD)**

8147 **H.5.1 User application notes**

8148 This component imposes requirements on the management of TSF data. Examples of TSF data
8149 are the current time and the audit trail.

8150 EXAMPLE

8151 this family allows the specification of whom **can** read, delete, or create the audit trail.

8152 **H.5.2 FMT_MTD.1 Management of TSF data**

8153 **H.5.2.1 Component rationale and application notes**

8154 This component allows users with a certain role to manage values of TSF data. The users are
8155 assigned to a role within the component FMT_SMR.1 Security roles.

8156 The default value of a parameter is the values the parameter takes when it is instantiated
8157 without specifically assigned values. An initial value is provided during the instantiation
8158 (creation) of a parameter and overrides the default value.

8159 **H.5.2.2 Operations**

8160 In FMT_MTD.1.1, the PP, PP-Module, functional package or ST author **should** specify the
8161 operations that **can** be applied to the identified TSF data. The PP, PP-Module, functional package
8162 or ST author **can** specify that the role **can** modify the default value (change_default), clear, query
8163 or modify the TSF data, or delete the TSF data entirely. If so desired the PP, PP-Module,
8164 functional package or ST author **could** specify any type of operation. To clarify “clear TSF data”
8165 means that the content of the TSF data is removed, but that the entity that stores the TSF data
8166 remains in the TOE.

8167 In FMT_MTD.1.1, the PP, PP-Module, functional package or ST author **should** specify the TSF
8168 data that **can** be operated on by the identified roles. It is possible for the PP, PP-Module,
8169 functional package or ST author to specify that the default value **can** be managed.

8170 In FMT_MTD.1.1, the PP, PP-Module, functional package or ST author **should** specify the roles
8171 that are allowed to operate on the TSF data. The possible roles are specified in FMT_SMR.1
8172 Security roles.

8173 In FMT_MTD.1.1, if selected, the PP, PP-Module, functional package or ST author **should** specify
8174 which other operations the role **could** perform.

8175 EXAMPLE

8176 An example of an operation is “create”.

8177 **H.5.3 FMT_MTD.2 Management of limits on TSF data**

8178 **H.5.3.1 Component rationale and application notes**

8179 This component specifies limits on TSF data, and actions to be taken if these limits are
8180 exceeded. This component will allow limits on the size of the audit trail to be defined, and
8181 specification of the actions to be taken when these limits are exceeded.

8182 **H.5.3.2 Operations**

8183 In FMT_MTD.2.1, the PP, PP-Module, functional package or ST author **should** specify the TSF
 8184 data that **can** have limits, and the value of those limits. An example of such TSF data is the
 8185 number of users logged-in.

8186 In FMT_MTD.2.1, the PP, PP-Module, functional package or ST author **should** specify the roles
 8187 that are allowed to modify the limits on the TSF data and the actions to be taken. The possible
 8188 roles are specified in FMT_SMR.1 Security roles.

8189 In FMT_MTD.2.2, the PP, PP-Module, functional package or ST author **should** specify the actions
 8190 to be taken if the specified limit on the specified TSF data is exceeded.

8191 EXAMPLE

8192 An example of such a TSF action is that the authorized user is informed and an audit record is generated.

8193 **H.5.4 FMT_MTD.3 Secure TSF data**

8194 **H.5.4.1 Component rationale and application notes**

8195 This component covers requirements on the values that **can** be assigned to TSF data. The
 8196 assigned values **should** be such that the TOE will remain in a secure state.

8197 The definition of what “secure” means is not answered in this component but is left to the
 8198 development of the TOE and the resulting information in the guidance.

8199 **H.5.4.2 Operations**

8200 In FMT_MTD.3.1, the PP, PP-Module, functional package or ST author **should** specify what TSF
 8201 data require only secure values to be accepted.

8202 **H.6 Revocation (FMT_REV)**

8203 **H.6.1 User application notes**

8204 This family addresses revocation of security attributes for a variety of entities within a TOE.

8205 **H.6.2 FMT_REV.1 Revocation**

8206 **H.6.2.1 Component rationale and application notes**

8207 This component specifies requirements on the revocation of rights. It requires the specification
 8208 of the revocation rules. Examples are:

- 8209 a) Revocation will take place on the next login of the user;
- 8210 b) Revocation will take place on the next attempt to open the file;
- 8211 c) Revocation will take place within a fixed time. This might mean that all open
 8212 connections are re-evaluated every x minutes.

8213 **H.6.2.2 Operations**

8214 In FMT_REV.1.1, the PP, PP-Module, functional package or ST author **should** specify which
 8215 security attributes are to be revoked when a change is made to the associated
 8216 object/subject/user/other resource.

8217 In FMT_REV.1.1, the PP, PP-Module, functional package or ST author **should** specify whether the
 8218 ability to revoke security attributes from users, subjects, objects, or any additional resources
 8219 **shall** be provided by the TSF.

8220 In FMT_REV.1.1, the PP, PP-Module, functional package or ST author **should** specify the roles
 8221 that are allowed to modify the functions in the TSF. The possible roles are specified in
 8222 FMT_SMR.1 Security roles.

8223 In FMT_REV.1.1, the PP, PP-Module, functional package or ST author **should**, if additional
 8224 resources is selected, specify whether the ability to revoke their security attributes **shall** be
 8225 provided by the TSF.

8226 In FMT_REV.1.2, the PP, PP-Module, functional package or ST author **should** specify the
8227 revocation rules. Examples of these rules **could** include: “prior to the next operation on the
8228 associated resource”, or “for all new subject creations”.

8229 **H.7 Security attribute expiration (FMT_SAE)**

8230 **H.7.1 User application notes**

8231 This family addresses the capability to enforce time limits for the validity of security attributes.
8232 This family **can** be applied to specify expiration requirements for access control attributes,
8233 identification and authentication attributes, certificates, audit attributes, etc.

8234 EXAMPLE

8235 An example of a certificate is key certificates such as ANSI X509.

8236 **H.7.2 FMT_SAE.1 Time-limited authorization**

8237 **H.7.2.1 Operations**

8238 In FMT_SAE.1.1, the PP, PP-Module, functional package or ST author **should** provide the list of
8239 security attributes for which expiration is to be supported.

8240 EXAMPLE

8241 An example of such an attribute might be a user's security clearance.

8242 In FMT_SAE.1.1, the PP, PP-Module, functional package or ST author **should** specify the roles
8243 that are allowed to modify the security attributes in the TSF. The possible roles are specified in
8244 FMT_SMR.1 Security roles.

8245 In FMT_SAE.1.2, the PP, PP-Module, functional package or ST author **should** provide a list of
8246 actions to be taken for each security attribute when it expires. An example might be that the
8247 user's security clearance, when it expires, is set to the lowest allowable clearance on the TOE. If
8248 immediate revocation is desired by the PP, PP-Module, functional package or ST, the action
8249 “immediate revocation” **should** be specified.

8250 **H.8 Specification of Management Functions (FMT_SMF)**

8251 **H.8.1 User application notes**

8252 This family allows the specification of the management functions to be provided by the TOE.
8253 Each security management function that is listed in fulfilling the assignment is either security
8254 attribute management, TSF data management, or security function management.

8255 **H.8.2 FMT_SMF.1 Specification of Management Functions**

8256 **H.8.2.1 Component rationale and application notes**

8257 This component specifies the management functions to be provided.

8258 PP, PP-Module, functional package or ST authors **should** consult the “Management” subclauses
8259 for components included in their PP, PP-Module, functional package or ST to provide a basis for
8260 the management functions to be listed via this component.

8261 **H.8.2.2 Operations**

8262 In FMT_SMF.1.1, the PP, PP-Module, functional package or ST author **should** specify the
8263 management functions to be provided by the TSF, either security attribute management, TSF
8264 data management, or security function management.

8265 **H.9 Security management roles (FMT_SMR)**

8266 **H.9.1 User application notes**

8267 This family reduces the likelihood of damage resulting from users abusing their authority by
 8268 taking actions outside their assigned functional responsibilities. It also addresses the threat that
 8269 inadequate mechanisms have been provided to securely administer the TSF.

8270 This family requires that information be maintained to identify whether a user is authorized to
 8271 use a particular security-relevant administrative function.

8272 Some management actions **can** be performed by users, others only by designated people within
 8273 the organization. This family allows the definition of different roles, such as owner, auditor,
 8274 administrator, daily-management.

8275 The roles as used in this family are security related roles. Each role **can** encompass an extensive
 8276 set of capabilities or **can** be a single right. This family defines the roles. The capabilities of the
 8277 role are defined in Limited capabilities and availability (FMT_LIM), Management of security
 8278 attributes (FMT_MSA) and Management of TSF data (FMT_MTD).

8279 EXAMPLE 1

8280 Set of capabilities: root in UNIX

8281 Single right: right to read a single object such as the helpfile.

8282 Some type of roles might be mutually exclusive.

8283 EXAMPLE 2

8284 The daily-management might be able to define and activate users but might not be able to remove users (which is
 8285 reserved for the administrator (role)).

8286 This class will allow policies such as two-person control to be specified.

8287 **H.9.2 FMT_SMR.1 Security roles**

8288 **H.9.2.1 Component rationale and application notes**

8289 This component specifies the different roles that the TSF **should** recognize. Often the system
 8290 distinguishes between the owner of an entity, an administrator, and other users.

8291 **H.9.2.2 Operations**

8292 In FMT_SMR.1.1, the PP, PP-Module, functional package or ST author **should** specify the roles
 8293 that are recognized by the system. These are the roles that users **could** occupy with respect to
 8294 security.

8295 EXAMPLE

8296 Examples of roles are: owner, auditor, and administrator.

8297 **H.9.3 FMT_SMR.2 Restrictions on security roles**

8298 **H.9.3.1 Component rationale and application notes**

8299 This component specifies the different roles that the TSF **should** recognize, and conditions on
 8300 how those roles **could** be managed. Often the system distinguishes between the owner of an
 8301 entity, an administrator, and other users.

8302 The conditions on those roles specify the interrelationship between the different roles, as well
 8303 as restrictions on when the role **can** be assumed by a user.

8304 **H.9.3.2 Operations**

8305 In FMT_SMR.2.1, the PP, PP-Module, functional package or ST author **should** specify the roles
 8306 that are recognized by the system. These are the roles that users **could** occupy with respect to
 8307 security.

8308 EXAMPLE 1

8309 Examples of roles are: owner, auditor, and administrator.

8310 In FMT_SMR.2.3, the PP, PP-Module, functional package or ST author **should** specify the
8311 conditions that govern role assignment.

8312 EXAMPLE2

8313 Examples of these conditions are: “an account cannot have both the auditor and administrator role” or “a user with
8314 the assistant role must also have the owner role”.

8315 **H.9.4 FMT_SMR.3 Assuming roles**

8316 **H.9.4.1 Component rationale and application notes**

8317 This component specifies that an explicit request must be given to assume the specific role.

8318 **H.9.4.2 Operations**

8319 In FMT_SMR.3.1, the PP, PP-Module, functional package or ST author **should** specify the roles
8320 that require an explicit request to be assumed.

8321 EXAMPLE

8322 Examples of roles are: owner, auditor, and administrator.

Annex I (normative)

8323
8324
8325
8326

Class FPR: Privacy- application notes

8327 I.1 General Information

8328 This class describes the requirements that **could** be levied to satisfy the users' privacy needs,
8329 while still allowing the system flexibility as far as possible to maintain sufficient control over
8330 the operation of the system.

8331 In the components of this class there is flexibility as to whether or not authorized users are
8332 covered by the required security functionality.

8333 EXAMPLE 1

8334 a PP, PP-Module, functional package or ST author might consider it appropriate not to require protection of the
8335 privacy of users against a suitably authorized user.

8336 This class, together with other classes (such as those concerned with audit, access control,
8337 trusted path, and non-repudiation) provides the flexibility to specify the desired privacy
8338 behaviour. On the other hand, the requirements in this class might impose limitations on the
8339 use of the components of other classes, such as FIA: Identification and authentication or FAU:
8340 Security audit.

8341 EXAMPLE 2

8342 If authorized users are not allowed to see the user identity (perhaps because of Anonymity or Pseudonymity), it will
8343 obviously not be possible to hold individual users accountable for any security relevant actions they perform that are
8344 covered by the privacy requirements. However, it may still be possible to include audit requirements in a PP, PP-
8345 Module, functional package or ST, where the fact that a particular security relevant event has occurred is more
8346 important than knowing who was responsible for it.

8347 Additional information is provided in the application notes for class FAU: Security audit, where
8348 it is explained that the definition of "identity" in the context of auditing **can** also be an alias or
8349 other information that **could** identify a user.

8350 This class describes four families: Anonymity, Pseudonymity, Unlinkability and Unobservability.
8351 Anonymity, Pseudonymity and Unlinkability have a complex interrelationship. When choosing a
8352 family, the choice **should** depend on the threats identified. For some types of privacy threats,
8353 pseudonymity will be more appropriate than anonymity.

8354 EXAMPLE 3

8355 If there is a requirement for auditing.

8356 In addition, some types of privacy threats are best countered by a combination of components
8357 from several families.

8358 All families assume that a user does not explicitly perform an action that discloses the user's
8359 own identity.

8360 EXAMPLE 4

8361 The TSF is not expected to screen the user name in electronic messages or databases.

8362 All families in this class have components that are scoped through operations. These operations
8363 allow the PP, PP-Module, functional package or ST author to state the cooperating
8364 users/subjects to which the TSF must be resistant.

8365 EXAMPLE 5

8366 An instantiation of anonymity **could** be: "The TSF shall ensure that the users and/or subjects are unable to determine
8367 the user identity bound to the teleconsulting application".

8368 It is noted that the TSF should not only provide this protection against individual users, but also against users
8369 cooperating to obtain the information.

8370 The editors' have added the following note since it seemed pertinent in this section of the
8371 application notes. If no comments to the contrary are made then this note will remain in the
8372 next draft.

8373 NOTE The reader's attention is drawn to ISO/IEC TS 19608:2018 Guidance for developing security and privacy
8374 functional requirements based on ISO/IEC 15408. ISO/IEC TS 19608:2018 provides guidance for:

- 8375 — selecting and specifying security functional requirements (SFRs) from ISO/IEC 15408-2 to protect
8376 Personally Identifiable Information (PII);
- 8377 — the procedure to define both privacy and security functional requirements in a coordinated manner; and
- 8378 — developing privacy functional requirements as extended components based on the privacy principles
8379 defined in ISO/IEC 29100 through the paradigm described in ISO/IEC 15408-2.

8380 I.2 Anonymity (FPR_ANO)

8381 I.2.1 User application notes

8382 Anonymity ensures that a subject **may** use a resource or service without disclosing its user
8383 identity.

8384 The intention of this family is to specify that a user or subject might take action without
8385 releasing its user identity to others such as users, subjects, or objects. The family provides the
8386 PP, PP-Module, functional package or ST author with a means to identify the set of users that
8387 cannot see the identity of someone performing certain actions.

8388 Therefore, if a subject, using anonymity, performs an action, another subject will not be able to
8389 determine either the identity or even a reference to the identity of the user employing the
8390 subject. The focus of the anonymity is on the protection of the user's identity, not on the
8391 protection of the subject identity; hence, the identity of the subject is not protected from
8392 disclosure.

8393 Although the identity of the subject is not released to other subjects or users, the TSF is not
8394 explicitly prohibited from obtaining the users identity. In case the TSF is not allowed to know
8395 the identity of the user, FPR_ANO.2 Anonymity without soliciting information **could** be invoked.
8396 In that case, the TSF **should** not request the user information.

8397 The interpretation of "determine" **should** be taken in the broadest sense of the word.

8398 The Components leveling and description distinguishes between the users and an authorized
8399 user. An authorized user is often excluded from the component, and therefore allowed to
8400 retrieve a user's identity. However, there is no specific requirement that an authorized user
8401 must be able to have the capability to determine the user's identity. For ultimate privacy, the
8402 components would be used to say that no user or authorized user **can** see the identity of anyone
8403 performing any action.

8404 Although some systems will provide anonymity for all services that are provided, other systems
8405 provide anonymity for certain subjects/operations. To provide this flexibility, an operation is
8406 included where the scope of the requirement is defined. If the PP, PP-Module, functional
8407 package or ST author wants to address all subjects/operations, the words "all subjects and all
8408 operations" **could** be provided.

8409 Possible applications include the ability to make enquiries of a confidential nature to public
8410 databases, respond to electronic polls, or make anonymous payments or donations.

8411 EXAMPLE

8412 Potential hostile users or subjects include providers, system operators, communication partners and users, who
8413 smuggle malicious parts (including malware) into systems. All of these users can investigate usage patterns (such as
8414 which users used which services) and misuse this information.

8415 I.2.2 FPR_ANO.1 Anonymity

8416 I.2.2.1 Component rationale and application notes

8417 This component ensures that the identity of a user is protected from disclosure. There **may** be
 8418 instances, however, that a given authorized user **can** determine who performed certain actions.
 8419 This component gives the flexibility to capture either a limited or total privacy policy.

8420 **I.2.2.2 Operations**

8421 In FPR_ANO.1.1, the PP, PP-Module, functional package or ST author **should** specify the set of
 8422 users and/or subjects against which the TSF must provide protection. For example, even if the
 8423 PP, PP-Module, functional package or ST author specifies a single user or subject role, the TSF
 8424 must not only provide protection against each individual user or subject but must protect with
 8425 respect to cooperating users and/or subjects.

8426 EXAMPLE 1

8427 A set of users **could** be a group of users which can operate under the same role or can all use the same process(es).

8428 In FPR_ANO.1.1, the PP, PP-Module, functional package or ST author **should** identify the list of
 8429 subjects and/or operations and/or objects where the real user name of the subject **should** be
 8430 protected.

8431 EXAMPLE 2

8432 An example of an object is “the voting application”.

8433 **I.2.3 FPR_ANO.2 Anonymity without soliciting information**

8434 **I.2.3.1 Component rationale and application notes**

8435 This component is used to ensure that the TSF is not allowed to know the identity of the user.

8436 **I.2.3.2 Operations**

8437 In FPR_ANO.2.1, the PP, PP-Module, functional package or ST author **should** specify the set of
 8438 users and/or subjects against which the TSF must provide protection. For example, even if the
 8439 PP, PP-Module, functional package or ST author specifies a single user or subject role, the TSF
 8440 must not only provide protection against each individual user or subject but must protect with
 8441 respect to cooperating users and/or subjects.

8442 EXAMPLE 1

8443 A set of users **could** be a group of users which can operate under the same role or can all use the same process(es).

8444 In FPR_ANO.2.1, the PP, PP-Module, functional package or ST author **should** identify the list of
 8445 subjects and/or operations and/or objects where the real user name of the subject **should** be
 8446 protected.

8447 EXAMPLE 2

8448 “the voting application”.

8449 In FPR_ANO.2.2, the PP, PP-Module, functional package or ST author **should** identify the list of
 8450 services which are subject to the anonymity requirement, for example, “the accessing of job
 8451 descriptions”.

8452 In FPR_ANO.2.2, the PP, PP-Module, functional package or ST author **should** identify the list of
 8453 subjects from which the real user name of the subject **should** be protected when the specified
 8454 services are provided.

8455 **I.3 Pseudonymity (FPR_PSE)**

8456 **I.3.1 User application notes**

8457 Pseudonymity ensures that a user **may** use a resource or service without disclosing its identity
 8458 but **can** still be accountable for that use. The user **can** be accountable by directly being related to
 8459 a reference (alias) held by the TSF, or by providing an alias that will be used for processing
 8460 purposes, such as an account number.

8461 In several respects, pseudonymity resembles anonymity. Both pseudonymity and anonymity
8462 protect the identity of the user, but in pseudonymity a reference to the user's identity is
8463 maintained for accountability or other purposes.

8464 The component FPR_PSE.1 Pseudonymity does not specify the requirements on the reference to
8465 the user's identity. For the purpose of specifying requirements on this reference two sets of
8466 requirements are presented: FPR_PSE.2 Reversible pseudonymity and FPR_PSE.3 Alias
8467 pseudonymity.

8468 A way to use the reference is by being able to obtain the original user identity.

8469 EXAMPLE 1

8470 In a digital cash environment, it would be advantageous to be able to trace the user's identity when a check has been
8471 issued multiple times (i.e. fraud).

8472 In general, the user's identity needs to be retrieved under specific conditions. The PP, PP-
8473 Module, functional package or ST author might want to incorporate FPR_PSE.2 Reversible
8474 pseudonymity to describe those services.

8475 Another usage of the reference is as an alias for a user.

8476 EXAMPLE 2

8477 A user who does not wish to be identified, **can** provide an account to which the resource utilization **should** be
8478 charged. In such cases, the reference to the user identity is an alias for the user, where other users or subjects **can** use
8479 the alias for performing their functions without ever obtaining the user's identity (for example, statistical operations
8480 on use of the system). In this case, the PP, PP-Module, functional package or ST author might wish to incorporate
8481 FPR_PSE.3 Alias pseudonymity to specify the rules to which the reference must conform.

8482 Using these constructs above, digital money **can** be created using FPR_PSE.2 Reversible
8483 pseudonymity specifying that the user identity will be protected and, if so specified in the
8484 condition, that there be a requirement to trace the user identity if the digital money is spent
8485 twice. When the user is honest, the user identity is protected; if the user tries to cheat, the user
8486 identity **can** be traced.

8487 A different kind of system **could** be a digital credit card, where the user will provide a
8488 pseudonym that indicates an account from which the cash **can** be subtracted. In such cases, for
8489 example, FPR_PSE.3 Alias pseudonymity **could** be used. This component would specify that the
8490 user identity will be protected and, furthermore, that the same user will only get assigned
8491 values for which he/she has provided money (if so specified in the conditions).

8492 It **should** be realized that the more stringent components potentially cannot be combined with
8493 other requirements, such as identification and authentication or audit. The interpretation of
8494 "determine the identity" **should** be taken in the broadest sense of the word. The information is
8495 not provided by the TSF during the operation, nor **can** the entity determine the subject or the
8496 owner of the subject that invoked the operation, nor will the TSF record information, available
8497 to the users or subjects, which might release the user identity in the future.

8498 The intent is that the TSF not reveal any information that would compromise the identity of the
8499 user.

8500 EXAMPLE 3

8501 The identity of subjects acting on the user's behalf.

8502 The information that is considered to be sensitive depends on the effort an attacker is capable
8503 of spending.

8504 Possible applications include the ability to charge a caller for premium rate telephone services
8505 without disclosing his or her identity, or to be charged for the anonymous use of an electronic
8506 payment system.

8507 EXAMPLE 4

8508 Potential hostile users include providers, system operators, communication partners and users, who smuggle
8509 malicious parts (including malware) into systems. All of these attackers **can** investigate which users used which
8510 services and misuse this information. Additionally, to Anonymity services, Pseudonymity Services contains methods

8511 for authorization without identification, especially for anonymous payment (“Digital Cash”). This helps providers to
8512 obtain their payment in a secure way while maintaining customer anonymity.

8513 **I.3.2 FPR_PSE.1 Pseudonymity**

8514 **I.3.2.1 Component rationale and application notes**

8515 This component provides the user protection against disclosure of identity to other users. The
8516 user will remain accountable for its actions.

8517 **I.3.2.2 Operations**

8518 In FPR_PSE.1.1, the PP, PP-Module, functional package or ST author **should** specify the set of
8519 users and/or subjects against which the TSF must provide protection. For example, even if the
8520 PP, PP-Module, functional package or ST author specifies a single user or subject role, the TSF
8521 must not only provide protection against each individual user or subject but must protect with
8522 respect to cooperating users and/or subjects.

8523 EXAMPLE 1

8524 A set of users **could** be a group of users which **can** operate under the same role or **can** all use the same process(es).

8525 In FPR_PSE.1.1, the PP, PP-Module, functional package or ST author **should** identify the list of
8526 subjects and/or operations and/or objects where the real user name of the subject **should** be
8527 protected.

8528 EXAMPLE 2

8529 “the accessing of job offers”.

8530 Note “objects” includes any other attributes that might enable another user or subject to derive the actual
8531 identity of the user.

8532 In FPR_PSE.1.2, the PP, PP-Module, functional package or ST author **should** identify the (one or
8533 more) number of aliases the TSF is able to provide.

8534 In FPR_PSE.1.2, the PP, PP-Module, functional package or ST author **should** identify the list of
8535 subjects to whom the TSF is able to provide an alias.

8536 In FPR_PSE.1.3, the PP, PP-Module, functional package or ST author **should** specify whether the
8537 user alias is generated by the TSF or supplied by the user. Only one of these options **may** be
8538 chosen.

8539 In FPR_PSE.1.3, the PP, PP-Module, functional package or ST author **should** identify the metric
8540 to which the TSF-generated or user-generated alias **should** conform.

8541 **I.3.3 FPR_PSE.2 Reversible pseudonymity**

8542 **I.3.3.1 Component rationale and application notes**

8543 In this component, the TSF **shall** ensure that under specified conditions the user identity related
8544 to a provided reference **can** be determined.

8545 In FPR_PSE.1 Pseudonymity the TSF **shall** provide an alias instead of the user identity. When the
8546 specified conditions are satisfied, the user identity to which the alias belong **can** be determined.

8547 EXAMPLE

8548 Such a condition in an electronic cash environment is: “The TSF shall provide the notary a capability to determine the
8549 user identity based on the provided alias only under the conditions that a check has been issued twice.”

8550 **I.3.3.2 Operations**

8551 In FPR_PSE.2.1, the PP, PP-Module, functional package or ST author **should** specify the set of
8552 users and/or subjects against which the TSF must provide protection.

8553 EXAMPLE 1

8554 Even if the PP, PP-Module, functional package or ST author specifies a single user or subject role, the TSF must not
8555 only provide protection against each individual user or subject but must protect with respect to cooperating users

8556 and/or subjects. A set of users, for example, **could** be a group of users which **can** operate under the same role or **can**
 8557 all use the same process(es).

8558 In FPR_PSE.2.1, the PP, PP-Module, functional package or ST author **should** identify the list of
 8559 subjects and/or operations and/or objects where the real user name of the subject **should** be
 8560 protected.

8561 EXAMPLE 2

8562 “The accessing of job offers”.

8563 NOTE “objects” includes any other attributes that might enable another user or subject to derive the actual
 8564 identity of the user.

8565 In FPR_PSE.2.2, the PP, PP-Module, functional package or ST author **should** identify the (one or
 8566 more) number of aliases the TSF, is able to provide.

8567 In FPR_PSE.2.2, the PP, PP-Module, functional package or ST author **should** identify the list of
 8568 subjects to whom the TSF is able to provide an alias.

8569 In FPR_PSE.2.3, the PP, PP-Module, functional package or ST author **should** specify whether the
 8570 user alias is generated by the TSF or supplied by the user. Only one of these options **may** be
 8571 chosen.

8572 In FPR_PSE.2.3, the PP, PP-Module, functional package or ST author **should** identify the metric
 8573 to which the TSF-generated or user-generated alias **should** conform.

8574 In FPR_PSE.2.4, the PP, PP-Module, functional package or ST author **should** select whether the
 8575 authorized user and/or trusted subjects **can** determine the real user name.

8576 In FPR_PSE.2.4, the PP, PP-Module, functional package or ST author **should** identify the list of
 8577 conditions under which the trusted subjects and authorized user **can** determine the real user
 8578 name based on the provided reference. These conditions **can** be conditions such as time of day,
 8579 or they **can** be administrative such as on a court order.

8580 In FPR_PSE.2.4, the PP, PP-Module, functional package or ST author **should** identify the list of
 8581 trusted subjects that **can** obtain the real user name under a specified condition.

8582 EXAMPLE

8583 A notary or special authorized user.

8584 **I.3.4 FPR_PSE.3 Alias pseudonymity**

8585 **I.3.4.1 Component rationale and application notes**

8586 In this component, the TSF **shall** ensure that the provided reference meets certain construction
 8587 rules, and thereby **can** be used in a secure way by potentially insecure subjects.

8588 If a user wants to use disk resources without disclosing its identity, pseudonymity **can** be used.
 8589 However, every time the user accesses the system, the same alias must be used. Such conditions
 8590 **can** be specified in this component.

8591 **I.3.4.2 Operations**

8592 In FPR_PSE.3.1, the PP, PP-Module, functional package or ST author **should** specify the set of
 8593 users and/or subjects against which the TSF must provide protection. For example, even if the
 8594 PP, PP-Module, functional package or ST author specifies a single user or subject role, the TSF
 8595 must not only provide protection against each individual user or subject but must protect with
 8596 respect to cooperating users and/or subjects.

8597 EXAMPLE 1

8598 A set of users **could** be a group of users which can operate under the same role or can all use the same process(es).

8599 In FPR_PSE.3.1, the PP, PP-Module, functional package or ST author **should** identify the list of
 8600 subjects and/or operations and/or objects where the real user name of the subject **should** be
 8601 protected.

8602 EXAMPLE 2

8603 “the accessing of job offers”.

8604 NOTE “objects” includes any other attributes which might enable another user or subject to derive the actual
8605 identity of the user.

8606 In FPR_PSE.3.2, the PP, PP-Module, functional package or ST author **should** identify the (one or
8607 more) number of aliases the TSF is able to provide.

8608 In FPR_PSE.3.2, the PP, PP-Module, functional package or ST author **should** identify the list of
8609 subjects to whom the TSF is able to provide an alias.

8610 In FPR_PSE.3.3, the PP, PP-Module, functional package or ST author **should** specify whether the
8611 user alias is generated by the TSF, or supplied by the user. Only one of these options **may** be
8612 chosen.

8613 In FPR_PSE.3.3, the PP, PP-Module, functional package or ST author **should** identify the metric
8614 to which the TSF-generated or user-generated alias **should** conform.

8615 In FPR_PSE.3.4, the PP, PP-Module, functional package or ST author **should** identify the list of
8616 conditions that indicate when the used reference for the real user name **shall** be identical and
8617 when it **shall** be different, for example, “when the user logs on to the same host” it will use a
8618 unique alias.

8619 I.4 Unlinkability (FPR_UNL)

8620 I.4.1 User application notes

8621 Unlinkability ensures that a user **may** make multiple uses of resources or services without
8622 others being able to link these uses together. Unlinkability differs from pseudonymity that,
8623 although in pseudonymity the user is also not known, relations between different actions **can** be
8624 provided.

8625 The requirements for unlinkability are intended to protect the user identity against the use of
8626 profiling of the operations.

8627 EXAMPLE 1

8628 For example, when a telephone smart card is employed with a unique number, the telephone company can determine
8629 the behaviour of the user of this telephone card. When a telephone profile of the users is known, the card can be
8630 linked to a specific user.

8631 Hiding the relationship between different invocations of a service or access of a resource will
8632 prevent this kind of information gathering.

8633 As a result, a requirement for unlinkability **could** imply that the subject and user identity of an
8634 operation must be protected. Otherwise this information might be used to link operations
8635 together.

8636 Unlinkability requires that different operations cannot be related. This relationship **can** take
8637 several forms.

8638 EXAMPLE 2

8639 The user associated with the operation, or the terminal which initiated the action, or the time the action was
8640 executed.

8641 The PP, PP-Module, functional package or ST author **can** specify what kind of relationships are
8642 present that must be countered.

8643 Possible applications include the ability to make multiple use of a pseudonym without creating
8644 a usage pattern that might disclose the user's identity.

8645 EXAMPLE 3

8646 Potential hostile subjects and users include providers, system operators, communication partners and users, who
8647 smuggle malicious parts, (including malware) into systems, they do not operate but want to get information about.
8648 All of these attackers **can** investigate (such as which users used which services) and misuse this information.

8649 Unlinkability protects users from linkages, which **could** be drawn between several actions of a
8650 customer.

8651 EXAMPLE 4

8652 a series of phone calls made by an anonymous customer to different partners, where the combination of the partner's
8653 identities might disclose the identity of the customer.

8654 **I.4.2 FPR_UNL.1 Unlinkability**

8655 **I.4.2.1 Component rationale and application notes**

8656 This component ensures that users cannot link different operations in the system and thereby
8657 obtain information.

8658 **I.4.2.2 Operations**

8659 In FPR_UNL.1.1, the PP, PP-Module, functional package or ST author **should** specify the set of
8660 users and/or subjects against which the TSF must provide protection.

8661 EXAMPLE 1

8662 Even if the PP, PP-Module, functional package or ST author specifies a single user or subject role, the TSF must not
8663 only provide protection against each individual user or subject but must protect with respect to cooperating users
8664 and/or subjects.

8665 EXAMPLE 2

8666 A set of users **could** be a group of users which can operate under the same role or can all use the same process(es).

8667 In FPR_UNL.1.1, the PP, PP-Module, functional package or ST author **should** identify the list of
8668 operations which **should** be subjected to the unlinkability requirement.

8669 EXAMPLE 3

8670 "Sending email".

8671 In FPR_UNL.1.1, the PP, PP-Module, functional package or ST author **should** select the
8672 relationships that **should** be obscured. The selection allows either the user identity or an
8673 assignment of relations to be specified.

8674 In FPR_UNL.1.1, the PP, PP-Module, functional package or ST author **should** identify the list of
8675 relations which **should** be protected against.

8676 EXAMPLE

8677 "Originate from the same IP address".

8678 **I.5 Unobservability (FPR_UNO)**

8679 **I.5.1 User application notes**

8680 Unobservability ensures that a user **may** use a resource or service without others, especially
8681 third parties, being able to observe that the resource or service is being used.

8682 Unobservability approaches the user identity from a different direction than the previous
8683 families Anonymity, Pseudonymity and Unlinkability. In this case, the intent is to hide the use of
8684 a resource or service, rather than to hide the user's identity.

8685 A number of techniques **can** be applied to implement unobservability.

8686 EXAMPLE

8687 Examples of techniques to provide unobservability are:

8688 a) Allocation of information impacting unobservability: Unobservability relevant information (such as.
8689 information that describes that an operation occurred) can be allocated in several locations within the TOE.
8690 The information might be allocated to a single randomly chosen part of the TOE such that an attacker does
8691 not know which part of the TOE should be attacked. An alternative system might distribute the information
8692 such that no single part of the TOE has sufficient information that, if circumvented, the privacy of the user
8693 would be compromised. This technique is explicitly addressed in FPR_UNO.2 Allocation of information
8694 impacting unobservability.

- 8695 b) Broadcast: When information is broadcast (such as Internet and Radio frequencies, including Ethernet,
8696 Bluetooth, WiFi and Near-field communication bands), users cannot determine who actually received and
8697 used that information. This technique is especially useful when information should reach receivers which
8698 have to fear a stigma for being interested in that information (such as sensitive medical information).
- 8699 c) Cryptographic protection and message padding: People observing a message stream might obtain
8700 information from the fact that a message is transferred and from attributes on that message. By traffic
8701 padding, message padding and encrypting the message stream, the transmission of a message and its
8702 attributes can be protected.

8703 Sometimes, users **should** not see the use of a resource, but an authorized user must be allowed
8704 to see the use of the resource in order to perform his duties. In such cases, the FPR_UNO.4
8705 Authorized user observability **could** be used, which provides the capability for one or more
8706 authorized users to see the usage.

8707 This family makes use of the concept “parts of the TOE”. This is considered any part of the TOE
8708 that is either physically or logically separated from other parts of the TOE.

8709 Unobservability of communications **may** be an important factor in many areas, such as the
8710 enforcement of constitutional rights, organizational policies, or in defense related applications.

8711 **I.5.2 FPR_UNO.1 Unobservability**

8712 **I.5.2.1 Component rationale and application notes**

8713 This component requires that the use of a function or resource cannot be observed by
8714 unauthorized users.

8715 **I.5.2.2 Operations**

8716 In FPR_UNO.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
8717 users and/or subjects against which the TSF must provide protection.

8718 EXAMPLE 1

8719 Even if the PP, PP-Module, functional package or ST author specifies a single user or subject role, the TSF must not
8720 only provide protection against each individual user or subject but must protect with respect to cooperating users
8721 and/or subjects.

8722 EXAMPLE 2

8723 A set of users **could** be a group of users which **can** operate under the same role or **can** all use the same process(es).

8724 In FPR_UNO.1.1, the PP, PP-Module, functional package or ST author **should** identify the list of
8725 operations that are subjected to the unobservability requirement. Other users/subjects will
8726 then not be able to observe the operations on a covered object in the specified list.

8727 EXAMPLE 3

8728 Reading and writing to the object.

8729 In FPR_UNO.1.1, the PP, PP-Module, functional package or ST author **should** identify the list of
8730 objects which are covered by the unobservability requirement.

8731 EXAMPLE 4

8732 A specific mail server or ftp site.

8733 In FPR_UNO.1.1, the PP, PP-Module, functional package or ST author **should** specify the set of
8734 protected users and/or subjects whose unobservability information will be protected.

8735 EXAMPLE 5

8736 “Users accessing the system through the internet”.

8737 **I.5.3 FPR_UNO.2 Allocation of information impacting unobservability**

8738 **I.5.3.1 Component rationale and application notes**

8739 This component requires that the use of a function or resource cannot be observed by specified
8740 users or subjects. Furthermore, this component specifies that information related to the privacy

8741 of the user is distributed within the TOE such that attackers might not know which part of the
8742 TOE to target, or they need to attack multiple parts of the TOE.

8743 An example of the use of this component is the use of a randomly allocated node to provide a
8744 function. In such a case the component might require that the privacy related information **shall**
8745 only be available to one identified part of the TOE and will not be communicated outside this
8746 part of the TOE.

8747 EXAMPLE

8748 A more complex example can be found in some “voting algorithms”. Several parts of the TOE will be involved in the
8749 service, but no individual part of the TOE will be able to violate the policy. So, a person may cast a vote (or not)
8750 without the TOE being able to determine whether a vote has been cast and what the vote happened to be (unless the
8751 vote was unanimous).

8752 I.5.3.2 Operations

8753 In FPR_UNO.2.1, the PP, PP-Module, functional package or ST author **should** specify the list of
8754 users and/or subjects against which the TSF must provide protection. For example, even if the
8755 PP, PP-Module, functional package or ST author specifies a single user or subject role, the TSF
8756 must not only provide protection against each individual user or subject but must protect with
8757 respect to cooperating users and/or subjects.

8758 EXAMPLE 1

8759 A set of users **could** be a group of users which **can** operate under the same role or **can** all use the same process(es).

8760 In FPR_UNO.2.1, the PP, PP-Module, functional package or ST author **should** identify the list of
8761 operations that are subjected to the unobservability requirement. Other users/subjects will
8762 then not be able to observe the operations on a covered object in the specified list

8763 EXAMPLE 2

8764 Reading and writing to the object.

8765 In FPR_UNO.2.1, the PP, PP-Module, functional package or ST author **should** identify the list of
8766 objects which are covered by the unobservability requirement. An example **could** be a specific
8767 mail server or ftp site.

8768 In FPR_UNO.2.1, the PP, PP-Module, functional package or ST author **should** specify the set of
8769 protected users and/or subjects whose unobservability information will be protected.

8770 EXAMPLE 3

8771 “Users accessing the system through the internet”.

8772 In FPR_UNO.2.2, the PP, PP-Module, functional package or ST author **should** identify which
8773 privacy related information **should** be distributed in a controlled manner.

8774 EXAMPLE 4

8775 This information **could** include: IP address of subject, IP address of object, time, used encryption keys.

8776 In FPR_UNO.2.2, the PP, PP-Module, functional package or ST author **should** specify the
8777 conditions to which the dissemination of the information **should** adhere. These conditions
8778 **should** be maintained throughout the lifetime of the privacy related information of each
8779 instance.

8780 EXAMPLE 5

8781 Examples of these conditions **could** be:

8782 — “the information shall only be present at a single separated part of the TOE and shall not be communicated
8783 outside this part of the TOE.”,

8784 — “the information shall only reside in a single separated part of the TOE, but shall be moved to another part
8785 of the TOE periodically”;

8786 — “the information shall be distributed between the different parts of the TOE such that compromise of any 5
8787 separated parts of the TOE will not compromise the security policy”.

8788 **I.5.4 FPR_UNO.3 Unobservability without soliciting information**

8789 **I.5.4.1 Component rationale and application notes**

8790 This component is used to require that the TSF does not try to obtain information that might
8791 compromise unobservability when provided specific services. Therefore, the TSF will not solicit
8792 (i.e. try to obtain from other entities) any information that might be used to compromise
8793 unobservability.

8794 **I.5.4.2 Operations**

8795 In FPR_UNO.3.1, the PP, PP-Module, functional package or ST author **should** identify the list of
8796 services which are subject to the unobservability requirement.

8797 EXAMPLE 1

8798 "The accessing of job descriptions".

8799 In FPR_UNO.3.1, the PP, PP-Module, functional package or ST author **should** identify the list of
8800 subjects from which privacy related information **should** be protected when the specified
8801 services are provided.

8802 In FPR_UNO.3.1, the PP, PP-Module, functional package or ST author **should** specify the privacy
8803 related information that will be protected from the specified subjects.

8804 EXAMPLE 2

8805 Examples of privacy related information include the identity of the subject that used a service and the quantity of a
8806 service that has been used such as memory resource utilization.

8807 **I.5.5 FPR_UNO.4 Authorized user observability**

8808 **I.5.5.1 Component rationale and application notes**

8809 This component is used to require that there will be one or more authorized users with the
8810 rights to view the resource utilization. Without this component, this review is allowed, but not
8811 mandated.

8812 **I.5.5.2 Operations**

8813 In FPR_UNO.4.1, the PP, PP-Module, functional package or ST author **should** specify the set of
8814 authorized users for which the TSF must provide the capability to observe the resource
8815 utilization. A set of authorized users, for example, **could** be a group of authorized users which
8816 **can** operate under the same role or **can** all use the same process(es).

8817 In FPR_UNO.4.1, the PP, PP-Module, functional package or ST author **should** specify the set of
8818 resources and/or services that the authorized user must be able to observe.

Annex J (normative)

Class FPT: Protection of the TSF- application notes

8823 J.1 General information

8824 This class contains families of functional requirements that relate to the integrity and
8825 management of the mechanisms that constitute the TSF and to the integrity of TSF data. In some
8826 sense, families in this class **may** appear to duplicate components in the FDP: User data
8827 protection class; they **may** even be implemented using the same mechanisms. However, FDP:
8828 User data protection focuses on user data protection, while FPT: Protection of the TSF focuses
8829 on TSF data protection. In fact, components from the FPT: Protection of the TSF class are
8830 necessary to provide requirements that the SFPs in the TOE cannot be tampered with or
8831 bypassed.

8832 From the point of view of this class, regarding to the TSF there are three significant elements:

- 8833 a) The TSF's implementation, which executes and implements the mechanisms that
8834 enforce the SFRs.
- 8835 b) The TSF's data, which are the administrative databases that guide the enforcement
8836 of the SFRs.
- 8837 c) The external entities that the TSF **may** interact with in order to enforce the SFRs.

8838 All of the families in the FPT: Protection of the TSF class **can** be related to these areas, and fall
8839 into the following groupings:

- 8840 a) TOE emanation (FPT_EMS), which addresses potential leakage of information from
8841 the TOE via emanations.
- 8842 b) Trusted recovery (FPT_RCV), Fail secure (FPT_FLS), and Internal TOE TSF data
8843 replication consistency (FPT_TRC), which address the behaviour of the TSF when
8844 failure occurs and immediately after.
- 8845 c) TSF initialization (FPT_INI), which addresses the initialization of the TOE into a
8846 correct and secure operational state.
- 8847 d) Internal TOE TSF data transfer (FPT_ITT), which addresses protection of TSF data
8848 when it is transmitted between physically-separated parts of the TOE.
- 8849 e) TSF physical protection (FPT_PHP), which provides an authorized user with the
8850 ability to detect external attacks on the parts of the TOE that comprise the TSF.
- 8851 f) Availability of exported TSF data (FPT_ITA), Confidentiality of exported TSF data
8852 (FPT_ITC), Integrity of exported TSF data (FPT_ITI), which address the protection
8853 and availability of TSF data between the TSF and another trusted IT product.
- 8854 g) Replay detection (FPT_RPL), which addresses the replay of various types of
8855 information and/or operations.
- 8856 h) State synchrony protocol (FPT_SSP), which addresses the synchronization of states,
8857 based upon TSF data, between different parts of a distributed TSF.
- 8858 i) Time stamps (FPT_STM), which addresses reliable timing.
- 8859 j) Inter-TSF TSF data consistency (FPT_TDC), which addresses the consistency of TSF
8860 data shared between the TSF and another trusted IT product.
- 8861 k) Testing of external entities (FPT_TEE) and TSF self-test (FPT_TST), which provide
8862 an authorized user with the ability to verify the correct operation of the external

8863 entities interacting with the TSF to enforce the SFRs, and the integrity of the TSF
8864 data and TSF itself.

8865 **J.2 FPT_EMS TOE emanation**

8866 **J.2.1 User application notes**

8867 This family defines the requirements for the TOE to be able to prevent or mitigate attacks
8868 against data stored in and used by the TOE where the attack is based on external observable
8869 physical phenomena of the TOE.

8870 EXAMPLE

8871 Examples of such attacks are analysis of TOE's electromagnetic radiation, simple power analysis (SPA), differential
8872 power analysis (DPA), timing attacks, etc.

8873 FPT_EMS.1.1 Limit of Emissions requires the TOE to not emit intelligible emissions enabling
8874 access to TSF data or user data.

8875 **J.2.2 FPT_EMS.1 TOE emanation**

8876 **J.2.2.1 Component rationale and application notes**

8877 **FPT_EMS.1.1 Table** found as part of the FPT_EMS.1.1 Limit of Emissions element shall be
8878 completed by the PP, PP-Module, functional package or ST author. Each row, which can be
8879 identified using the "Identifier", provides a set of assignments for completing the SFR, allowing
8880 the PP, PP-Module, functional package or ST author to specify the requirements for TOE
8881 emanation protection for various different combinations of emissions, interfaces, TSF data and
8882 user data.

8883 EXAMPLE

8884 Types of emission can include audio frequencies and radio frequencies.

8885 Types of interfaces can include physical ports, I.C. boundaries, and electronic components.

8886 **J.3 Fail secure (FPT_FLS)**

8887 **J.3.1 User application notes**

8888 The requirements of this family ensure that the TOE will always enforce its SFRs in the event of
8889 certain types of failures in the TSF.

8890 **J.3.2 FPT_FLS.1 Failure with preservation of secure state**

8891 **J.3.2.1 Component rationale and application notes**

8892 The term "secure state" refers to a state in which the TSF data are consistent and the TSF
8893 continues correct enforcement of the SFRs.

8894 Although it is desirable to audit situations in which failure with preservation of secure state
8895 occurs, it is not possible in all situations. The PP, PP-Module, functional package or ST author
8896 **should** specify those situations in which audit is desired and feasible.

8897 Failures in the TSF **may** include "hard" failures, which indicate an equipment malfunction and
8898 which **may** require maintenance, service, or repair of the TSF. Failures in the TSF **may** also
8899 include recoverable "soft" failures, which **may** only require initialization or resetting of the TSF.

8900 **J.3.2.2 Operations**

8901 In FPT_FLS.1.1, the PP, PP-Module, functional package or ST author **should** list the types of
8902 failures in the TSF for which the TSF **should** "fail secure," that is, **should** preserve a secure state
8903 and continue to correctly enforce the SFRs.

8904 **J.4 TSF initialization (FPT_INI)**

8905 **J.4.1 User application notes**

8906 This family defines the functional requirements for the initialization of the TSF. By a dedicated
8907 function of the TOE that ensures that the initialization of the TSF results in a correct and secure
8908 operational state. This can cover code/data that are stored and executed from non-modifiable
8909 memory at boot time, the immutable root-of-trust, and other one-time programmable (OTP)
8910 values such as versions and identifiers.

8911 **J.4.2 FPT_INI.1 TSF initialization**

8912 **J.4.2.1 Operations**

8913 In FPT_INI.1.2 the PP, PP-Module, functional package or ST author should list the properties and
8914 the elements to which they apply, using the assignment table format in the element.

8915 EXAMPLE

8916 Properties could include authenticity, integrity, correct version and elements to which the properties apply could
8917 include TSF or user firmware, software or data.

8918 In FPT_INI.1.3 the PP, PP-Module, functional package or ST author uses the selections and
8919 assignments to describe the behaviour of the TOE initialization function in the case that errors
8920 or other failures are encountered during the initialization.

8921 FPT_INI.1.4 the PP, PP-Module, functional package or ST author uses the assignment to describe
8922 the methods by which the TOE initialization function interacts with the TSF.

8923 **J.5 Availability of exported TSF data (FPT_ITA)**

8924 **J.5.1 User application notes**

8925 This family defines the rules for the prevention of loss of availability of TSF data moving
8926 between the TSF and another trusted IT product. This data **could** be TSF critical data such as
8927 passwords, keys, audit data, or TSF executable code.

8928 This family is used in a distributed context where the TSF is providing TSF data to another
8929 trusted IT product. The TSF **can** only take the measures at its site and cannot be held
8930 responsible for the TSF at the other trusted IT product.

8931 If there are different availability metrics for different types of TSF data, then this component
8932 **should** be iterated for each unique pairing of metrics and types of TSF data.

8933 **J.5.2 FPT_ITA.1 Inter-TSF availability within a defined availability metric**

8934 **J.5.2.1 Operations**

8935 In FPT_ITA.1.1, the PP, PP-Module, functional package or ST author **should** specify the types of
8936 TSF data that are subject to the availability metric.

8937 In FPT_ITA.1.1, the PP, PP-Module, functional package or ST **should** specify the availability
8938 metric for the applicable TSF data.

8939 In FPT_ITA.1.1, the PP, PP-Module, functional package or ST author **should** specify the
8940 conditions under which availability must be ensured.

8941 EXAMPLE

8942 There must be a connection between the TOE and another trusted IT product.

8943 **J.6 Confidentiality of exported TSF data (FPT_ITC)**

8944 **J.6.1 User application notes**

8945 This family defines the rules for the protection from unauthorized disclosure of TSF data
8946 moving between the TSF and another trusted IT product.

8947 EXAMPLE

8948 Examples of this data are TSF critical data such as passwords, keys, audit data, or TSF executable code.

8949 This family is used in a distributed context where the TSF is providing TSF data to another
8950 trusted IT product. The TSF **can** only take the measures at its site and cannot be held
8951 responsible for the behaviour of the other trusted IT product.

8952 **J.6.2 Evaluator notes**

8953 Confidentiality of TSF Data during transmission is necessary to protect such information from
8954 disclosure.

8955 EXAMPLE

8956 Some possible implementations that **could** provide confidentiality include the use of cryptographic algorithms as
8957 well as spread spectrum techniques.

8958 **J.6.3 FPT_ITC.1 Inter-TSF confidentiality during transmission**

8959 **J.6.3.1 Component rationale and application notes**

8960 This component is used when it is necessary to make the requirement for confidentiality of TSF
8961 data when being transmitted from the TSF to another trusted IT product.

8962 **J.7 Integrity of exported TSF data (FPT_ITI)**

8963 **J.7.1 User application notes**

8964 This family defines the rules for the protection, from unauthorized modification, of TSF data
8965 during transmission between the TSF and another trusted IT product.

8966 EXAMPLE

8967 Examples of this data are TSF critical data such as passwords, keys, audit data, or TSF executable code.

8968 This family is used in a distributed context where the TSF is exchanging TSF data with another
8969 trusted IT product. Note that a requirement that addresses modification, detection, or recovery
8970 at another trusted IT product cannot be specified, as the mechanisms that another trusted IT
8971 product will use to protect its data cannot be determined in advance. For this reason, these
8972 requirements are expressed in terms of the "TSF providing a capability" which another trusted
8973 IT product **can** use.

8974 **J.7.1.1 Evaluator notes**

8975 In the FPT_ITI.2 component some possible means of satisfying this requirement involves the
8976 use of cryptographic functions or some form of checksum.

8977 **J.7.2 FPT_ITI.1 Inter-TSF detection of modification**

8978 **J.7.2.1 Component rationale and application notes**

8979 This component **should** be used in situations where it is sufficient to detect when data have
8980 been modified. An example of such a situation is one in which another trusted IT product **can**
8981 request the TOE's TSF to retransmit data when modification has been detected or respond to
8982 such types of request.

8983 The desired strength of modification detection is based upon a specified modification metric
8984 that is a function of the algorithm used, which **may** range from a weak checksum and parity
8985 mechanisms that **may** fail to detect multiple bit changes, to more complicated cryptographic
8986 checksum approaches.

8987 **J.7.2.2 Operations**

8988 In FPT_ITI.1.1, the PP, PP-Module, functional package or ST **should** specify the modification
8989 metric that the detection mechanism must satisfy. This modification metric **shall** specify the
8990 desired strength of the modification detection.

8991 In FPT_ITI.1.2, the PP, PP-Module, functional package or ST **should** specify the actions to be
8992 taken if a modification of TSF data has been detected. An example of an action is: “ignore the
8993 TSF data and request the originating trusted product to send the TSF data again”.

8994 **J.7.3 FPT_ITI.2 Inter-TSF detection and correction of modification**

8995 **J.7.3.1 Component rationale and application notes**

8996 This component **should** be used in situations where it is necessary to detect or correct
8997 modifications of TSF critical data.

8998 The desired strength of modification detection is based upon a specified modification metric
8999 that is a function of the algorithm used, which **may** range from a checksum and parity
9000 mechanisms that **may** fail to detect multiple bit changes, to more complicated cryptographic
9001 checksum approaches. The metric that needs to be defined **can** either refer to the attacks it will
9002 resist or to mechanisms that are well known in the public literature.

9003 EXAMPLE

9004 Attack reference: “only 1 in 1000 random messages will be accepted”.

9005 Well known mechanism: “the strength must be conformant to the strength offered by Secure Hash Algorithm”.

9006 The approach taken to correct modification might be done through some form of error
9007 correcting checksum.

9008 **J.7.3.2 Operations**

9009 In FPT_ITI.2.1, the PP, PP-Module, functional package or ST **should** specify the modification
9010 metric that the detection mechanism must satisfy. This modification metric **shall** specify the
9011 desired strength of the modification detection.

9012 In FPT_ITI.2.2, the PP, PP-Module, functional package or ST **should** specify the actions to be
9013 taken if a modification of TSF data has been detected.

9014 EXAMPLE

9015 An example of an action is: “ignore the TSF data and request the originating trusted product to send the TSF data
9016 again”.

9017 In FPT_ITI.2.3, the PP, PP-Module, functional package or ST author **should** define the types of
9018 modification from which the TSF **should** be capable of recovering.

9019 **J.8 Internal TOE TSF data transfer (FPT_ITT)**

9020 **J.8.1 User application notes**

9021 This family provides requirements that address protection of TSF data when it is transferred
9022 between separate parts of a TOE across an internal channel.

9023 The determination of the degree of separation (i.e., physical, or logical) that would make
9024 application of this family useful depends on the intended environment of use. In a hostile
9025 environment, there **may** be risks arising from transfers between parts of the TOE separated by
9026 only a system bus or an inter-process communications channel. In more benign environments,
9027 the transfers **may** be across more traditional network media.

9028 **J.8.2 Evaluator notes**

9029 One practical mechanism available to a TSF to provide this protection is cryptographically-
9030 based.

9031 **J.8.3 FPT_ITT.1 Basic internal TSF data transfer protection**

9032 **J.8.3.1 Operations**

9033 In FPT_ITT.1.1, the PP, PP-Module, functional package or ST author **should** specify the desired
9034 type of protection to be provided from the choices: disclosure, modification.

9035 **J.8.4 FPT_ITT.2 TSF data transfer separation**9036 **J.8.4.1 Component rationale and application notes**

9037 One of the ways to achieve separation of TSF data based on SFP-relevant attributes is through
9038 the use of separate logical or physical channels.

9039 **J.8.4.2 Operations**

9040 In FPT_ITT.2.1, the PP, PP-Module, functional package or ST author **should** specify the desired
9041 type of protection to be provided from the choices: disclosure, modification.

9042 **J.8.5 FPT_ITT.3 TSF data integrity monitoring**9043 **J.8.5.1 Operations**

9044 In FPT_ITT.3.1, the PP, PP-Module, functional package or ST author **should** specify the desired
9045 type of modification that the TSF **shall** be able to detect. The PP, PP-Module, functional package
9046 or ST author **should** select from: modification of data, substitution of data, re-ordering of data,
9047 deletion of data, or any other integrity errors.

9048 In FPT_ITT.3.1, if the PP, PP-Module, functional package or ST author chooses the latter
9049 selection noted in the preceding paragraph, then the author **should** also specify what those
9050 other integrity errors are that the TSF **should** be capable of detecting.

9051 In FPT_ITT.3.2, the PP, PP-Module, functional package or ST author **should** specify the action to
9052 be taken when an integrity error is identified.

9053 **J.9 TSF physical protection (FPT_PHP)**9054 **J.9.1 User application notes**

9055 TSF physical protection components refer to restrictions on unauthorized physical access to the
9056 TSF, and to the deterrence of, and resistance to, unauthorized physical modification, or
9057 substitution of the TSF.

9058 The requirements in this family ensure that the TSF is protected from physical tampering and
9059 interference. Satisfying the requirements of these components results in the TSF being
9060 packaged and used in such a manner that physical tampering is detectable, or resistance to
9061 physical tampering is measurable based on defined work factors. Without these components,
9062 the protection functions of a TSF lose their effectiveness in environments where physical
9063 damage cannot be prevented. This component also provides requirements regarding how the
9064 TSF must respond to physical tampering attempts.

9065 **EXAMPLE 1**

9066 Examples of physical tampering scenarios include mechanical attack, radiation, changing the temperature.

9067 It is acceptable for the functions that are available to an authorized user for detecting physical
9068 tampering to be available only in an off-line or maintenance mode. Controls **should** be in place
9069 to limit access during such modes to authorized users. As the TSF **may** not be “operational”
9070 during those modes, it **may** not be able to provide normal enforcement for authorized user
9071 access. The physical implementation of a TOE might consist of several structures. This set of
9072 “elements” as a whole must protect (protect, notify and resist) the TSF from physical tampering.
9073 This does not mean that all devices must provide these features, but the complete physical
9074 construct as a whole **should**.

9075 **EXAMPLE 2**

9076 Examples of structures include an outer shielding, cards, and chips.

9077 Although there is only minimal auditing associating with these components, this is solely
 9078 because there is the potential that the detection and alarm mechanisms **may** be implemented
 9079 completely in hardware, below the level of interaction with an audit subsystem. Nevertheless, a
 9080 PP, PP-Module, functional package or ST author **may** determine that for a particular anticipated
 9081 threat environment, there is a need to audit physical tampering. If this is the case, the PP, PP-
 9082 Module, functional package or ST author **should** include appropriate requirements in the list of
 9083 audit events.

9084 NOTE Inclusion of these requirements **may** have implications on the hardware design and its interface to the
 9085 software.

9086 EXAMPLE 3

9087 Examples of a hardware-based detection system is one based on breaking a circuit and lighting a light emitting diode
 9088 (LED) if the circuit is broken when a button is pressed by the authorized user.

9089 **J.9.2 FPT_PHP.1 Passive detection of physical attack**

9090 **J.9.2.1 Component rationale and application notes**

9091 FPT_PHP.1 Passive detection of physical attack **should** be used when threats from unauthorized
 9092 physical tampering with parts of the TOE are not countered by procedural methods. It
 9093 addresses the threat of undetected physical tampering with the TSF. Typically, an authorized
 9094 user would be given the function to verify whether tampering took place. As written, this
 9095 component simply provides a TSF capability to detect tampering. Specification of management
 9096 functions in FMT_LIM.1 **should** be considered to specify who **can** make use of that capability,
 9097 and how they **can** make use of that capability. If this is done by non-IT mechanisms such as
 9098 physical inspection. management functions are not required.

9099 **J.9.3 FPT_PHP.2 Notification of physical attack**

9100 **J.9.3.1 Component rationale and application notes**

9101 FPT_PHP.2 Notification of physical attack **should** be used when threats from unauthorized
 9102 physical tampering with parts of the TOE are not countered by procedural methods, and it is
 9103 required that designated individuals be notified of physical tampering. It addresses the threat
 9104 that physical tampering with TSF elements, although detected, **may** not be noticed. Specification
 9105 of management functions in FMT_MOF.1 Management of security functions behaviour **should** be
 9106 considered to specify who **can** make use of that capability, and how they **can** make use of that
 9107 capability.

9108 **J.9.3.2 Operations**

9109 In FPT_PHP.2.3, the PP, PP-Module, functional package or ST author **should** provide a list of TSF
 9110 devices/elements for which active detection of physical tampering is required.

9111 In FPT_PHP.2.3, the PP, PP-Module, functional package or ST author **should** designate a user or
 9112 role that is to be notified when tampering is detected. The type of user or role **may** vary
 9113 depending on the particular security administration component (from the FMT_LIM.1 family)
 9114 included in the PP, PP-Module, functional package or ST.

9115 **J.9.4 FPT_PHP.3 Resistance to physical attack**

9116 **J.9.4.1 Component rationale and application notes**

9117 For some forms of tampering, it is necessary that the TSF not only detects the tampering, but
 9118 actually resists it or delays the attacker.

9119 This component **should** be used when TSF devices and TSF elements are expected to operate in
 9120 an environment where a physical tampering of the internals of a TSF device or TSF element
 9121 itself is a threat.

9122 EXAMPLE

9123 Physical tampering includes observation, analysis, or modification.

9124 **J.9.4.2 Operations**

9125 In FPT_PHP.3.1, the PP, PP-Module, functional package or ST author **should** specify tampering
 9126 scenarios to a list of TSF devices/elements for which the TSF **should** resist physical tampering.
 9127 This list **may** be applied to a defined subset of the TSF physical devices and elements based on
 9128 considerations such as technology limitations and relative physical exposure of the device. Such
 9129 sub setting **should** be clearly defined and justified. Furthermore, the TSF **should** automatically
 9130 respond to physical tampering. The automatic response **should** be such that the policy of the
 9131 device is preserved.

9132 EXAMPLE

9133 An example of policy protection:

9134 with a confidentiality policy, it would be acceptable to physically disable the device so that the protected information
 9135 **may** not be retrieved.

9136 In FPT_PHP.3.1, the PP, PP-Module, functional package or ST author **should** specify the list of
 9137 TSF devices/elements for which the TSF **should** resist physical tampering in the scenarios that
 9138 have been identified.

9139 **J.10 Trusted recovery (FPT_RCV)**9140 **J.10.1 User application notes**

9141 The requirements of this family ensure that the TSF **can** determine that the TOE is started-up
 9142 without protection compromise and **can** recover without protection compromise after
 9143 discontinuity of operations. This family is important because the start-up state of the TSF
 9144 determines the protection of subsequent states.

9145 Recovery components reconstruct the TSF secure states, or prevent transitions to insecure
 9146 states, as a direct response to occurrences of expected failures, discontinuity of operation or
 9147 start-up.

9148 EXAMPLE

9149 Failures that must be generally anticipated include the following:

- 9150 a) Unmaskable action failures that always result in a system crash (such as persistent inconsistency of critical
 9151 system tables, uncontrolled transfers within the TSF code caused by transient failures of hardware or
 9152 firmware, power failures, processor failures, communication failures).
- 9153 b) Media failures causing part or all of the media representing the TSF objects to become inaccessible or
 9154 corrupt (such as parity errors, disk head crash, persistent read/write failure caused by misaligned disk
 9155 heads, worn-out magnetic coating, dust on the disk surface, loss of Internet connection).
- 9156 c) Discontinuity of operation caused by erroneous administrative action or lack of timely administrative
 9157 action (such as unexpected shutdowns by turning off power, ignoring the exhaustion of critical resources,
 9158 inadequate installed configuration).

9159 NOTE Recovery may be from either a complete or partial failure scenario. Although a complete failure might
 9160 occur in a monolithic operating system, it is less likely to occur in a distributed environment. In such environments,
 9161 subsystems may fail, but other portions remain operational. Further, critical components may be redundant (disk
 9162 mirroring, alternative routes), and checkpoints may be available. Thus, recovery is expressed in terms of recovery to
 9163 a secure state.

9164 There are different interactions between Trusted recovery (FPT_RCV) and TSF self-test
 9165 (FPT_TST) components to be considered when selecting Trusted recovery (FPT_RCV):

- 9166 a) The need for trusted recovery **may** be indicated through the results of TSF self-
 9167 testing, where the results of the self-tests indicate that the TSF is in an insecure
 9168 state and return to a secure state or entrance in maintenance mode is required.
- 9169 b) A failure, as discussed above, **may** be identified by an administrator. Either the
 9170 administrator **may** perform the actions to return the TOE to a secure state and then
 9171 invoke TSF self-tests to confirm that the secure state has been achieved. Or, the TSF
 9172 self-tests **may** be invoked to complete the recovery process.

- 9173 c) A combination of a. and b. above, where the need for trusted recovery is indicated
 9174 through the results of TSF self-testing, the administrator performs the actions to
 9175 return the TOE to a secure state and then invokes TSF self-tests to confirm that the
 9176 secure state has been achieved.
- 9177 d) Self-tests detect a failure/service discontinuity, then either automated recovery or
 9178 entrance to a maintenance mode.

9179 This family identifies a maintenance mode. In this maintenance mode, normal operation might
 9180 be impossible or severely restricted, as otherwise insecure situations might occur. Typically,
 9181 only authorized users **should** be allowed access to this mode but the real details of who **can**
 9182 access this mode is a function of FMT: Security management. If FMT: Security management does
 9183 not put any controls on who **can** access this mode, then it **may** be acceptable to allow any user
 9184 to restore the system if the TOE enters such a state. However, in practice, this is probably not
 9185 desirable as the user restoring the system has an opportunity to configure the TOE in such a
 9186 way as to violate the SFRs.

9187 Mechanisms designed to detect exceptional conditions during operation fall under TSF self-test
 9188 (FPT_TST), Fail secure (FPT_FLS), and other areas that address the concept of “Software Safety.”
 9189 It is likely that the use of one of these families will be required to support the adoption of
 9190 Trusted recovery (FPT_RCV). This is to ensure that the TOE will be able to detect when recovery
 9191 is required.

9192 Throughout this family, the phrase “secure state” is used. This refers to some state in which the
 9193 TOE has consistent TSF data and a TSF that **can** correctly enforce the policy. This state **may** be
 9194 the initial “boot” of a clean system, or it might be some checkpointed state.

9195 Following recovery, it **may** be necessary to confirm that the secure state has been achieved
 9196 through self-testing of the TSF. However, if the recovery is performed in a manner such that
 9197 only a secure state **can** be achieved, else recovery fails, then the dependency to the FPT_TST.1
 9198 TSF self-testing component **may** be argued away.

9199 **J.10.1.1 Evaluator notes**

9200 In FPT_RCV.1, it is acceptable for the functions that are available to an authorized user for
 9201 trusted recovery to be available only in a maintenance mode. Controls **should** be in place to
 9202 limit access during maintenance to authorized users.

9203 In FPT_RCV.2 It is acceptable for the functions that are available to an authorized user for
 9204 trusted recovery to be available only in a maintenance mode. Controls **should** be in place to
 9205 limit access during maintenance to authorized users.

9206 For FPT_RCV.2.1, it is the responsibility of the developer of the TSF to determine the set of
 9207 recoverable failures and service discontinuities.

9208 It is assumed that the robustness of the automated recovery mechanisms will be verified.

9209 In FPT_RCV.3 It is acceptable for the functions that are available to an authorized user for
 9210 trusted recovery to be available only in a maintenance mode. Controls **should** be in place to
 9211 limit access during maintenance to authorized users.

9212 It is assumed that the evaluators will verify the robustness of the automated recovery
 9213 mechanisms.

9214 **J.10.2 FPT_RCV.1 Manual recovery**

9215 **J.10.2.1 Component rationale and application notes**

9216 In the hierarchy of the trusted recovery family, recovery that requires only manual intervention
 9217 is the least desirable, for it precludes the use of the system in an unattended fashion.

9218 This component is intended for use in TOEs that do not require unattended recovery to a secure
 9219 state. The requirements of this component reduce the threat of protection compromise

9220 resulting from an attended TOE returning to an insecure state after recovery from a failure or
9221 other discontinuity.

9222 **J.10.2.2 Operations**

9223 In FPT_RCV.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
9224 failures or service discontinuities following which the TOE will enter a maintenance mode.

9225 EXAMPLE

9226 Power failure, audit storage exhaustion, any failure or discontinuity.

9227 **J.10.3 FPT_RCV.2 Automated recovery**

9228 **J.10.3.1 Component rationale and application notes**

9229 Automated recovery is considered to be more useful than manual recovery, as it allows the
9230 machine to operate in an unattended fashion.

9231 The component FPT_RCV.2 Automated recovery extends the feature coverage of FPT_RCV.1
9232 Manual recovery by requiring that there be at least one automated method of recovery from
9233 failure or service discontinuity. It addresses the threat of protection compromise resulting from
9234 an unattended TOE returning to an insecure state after recovery from a failure or other
9235 discontinuity.

9236 **J.10.3.2 Operations**

9237 In FPT_RCV.2.1, the PP, PP-Module, functional package or ST author **should** specify the list of
9238 failures or service discontinuities following which the TOE will need to enter a maintenance
9239 mode.

9240 EXAMPLE

9241 Power failure, audit storage exhaustion.

9242 In FPT_RCV.2.2, the PP, PP-Module, functional package or ST author **should** specify the list of
9243 failures or other discontinuities for which automated recovery must be possible.

9244 **J.10.4 FPT_RCV.3 Automated recovery without undue loss**

9245 **J.10.4.1 Component rationale and application notes**

9246 Automated recovery is considered to be more useful than manual recovery, but it runs the risk
9247 of losing a substantial number of objects. Preventing undue loss of objects provides additional
9248 utility to the recovery effort.

9249 The component FPT_RCV.3 Automated recovery without undue loss extends the feature
9250 coverage of FPT_RCV.2 Automated recovery by requiring that there not be undue loss of TSF
9251 data or objects under the control of the TSF. At FPT_RCV.2 Automated recovery, the automated
9252 recovery mechanisms **could** conceivably recover by deleting all objects and returning the TSF to
9253 a known secure state. This type of drastic automated recovery is precluded in FPT_RCV.3
9254 Automated recovery without undue loss.

9255 This component addresses the threat of protection compromise resulting from an unattended
9256 TOE returning to an insecure state after recovery from a failure or other discontinuity with a
9257 large loss of TSF data or objects under the control of the TSF.

9258 **J.10.4.2 Operations**

9259 In FPT_RCV.3.1, the PP, PP-Module, functional package or ST author **should** specify the list of
9260 failures or service discontinuities following which the TOE will need to enter a maintenance
9261 mode.

9262 EXAMPLE

9263 Power failure, audit storage exhaustion.

9264 In FPT_RCV.3.2, the PP, PP-Module, functional package or ST author **should** specify the list of
9265 failures or other discontinuities for which automated recovery must be possible.

9266 In FPT_RCV.3.3, the PP, PP-Module, functional package or ST author **should** provide a
9267 quantification for the amount of loss of TSF data or objects that is acceptable.

9268 **J.10.5 FPT_RCV.4 Function recovery**

9269 **J.10.5.1 Component rationale and application notes**

9270 Function recovery requires that if there **should** be some failure in the TSF, that certain functions
9271 in the TSF **should** either complete successfully or recover to a secure state.

9272 **J.10.5.2 Operations**

9273 In FPT_RCV.4.1, the PP, PP-Module, functional package or ST author **should** specify a list the
9274 functions and failure scenarios. In the event that any of the identified failure scenarios happen,
9275 the functions that have been specified must either complete successfully or recover to a
9276 consistent and secure state.

9277 **J.11 Replay detection (FPT_RPL)**

9278 **J.11.1 User application notes**

9279 This family addresses detection of replay for various types of entities and subsequent actions to
9280 correct.

9281 **J.11.2 FPT_RPL.1 Replay detection**

9282 **J.11.2.1 Component rationale and application notes**

9283 The entities included here are those that can be involved in replay detection.

9284 EXAMPLE

9285 Messages, service requests, service responses, or sessions.

9286 **J.11.2.2 Operations**

9287 In FPT_RPL.1.1, the PP, PP-Module, functional package or ST author **should** provide a list of
9288 identified entities for which detection of replay **should** be possible.

9289 EXAMPLE

9290 Messages, service requests, service responses, and user sessions.

9291 In FPT_RPL.1.2, the PP, PP-Module, functional package or ST author **should** specify the list of
9292 actions to be taken by the TSF when replay is detected. The potential set of actions that **can** be
9293 taken includes: ignoring the replayed entity, requesting confirmation of the entity from the
9294 identified source, and terminating the subject from which the re-played entity originated.

9295 **J.12 State synchrony protocol (FPT_SSP)**

9296 **J.12.1 User application notes**

9297 Distributed TOEs **may** give rise to greater complexity than monolithic TOEs through the
9298 potential for differences in state between parts of the TOE, and through delays in
9299 communication. In most cases, synchronization of state between distributed functions involves
9300 an exchange protocol, not a simple action. When malice exists in the distributed environment of
9301 these protocols, more complex defensive protocols are required.

9302 State synchrony protocol (FPT_SSP) establishes the requirement for certain critical functions of
9303 the TSF to use a trusted protocol. State synchrony protocol (FPT_SSP) ensures that two
9304 distributed parts of the TOE, such as hosts, have synchronized their states after a security-
9305 relevant action.

9306 Some states **may** never be synchronized, or the transaction cost **may** be too high for practical
9307 use.

9308 EXAMPLE 1

9309 Encryption key revocation is an example, where knowing the state after the revocation action is initiated **can** never
9310 be known. Either the action was taken and acknowledgment cannot be sent, or the message was ignored by hostile
9311 communication partners and the revocation never occurred.

9312 Indeterminacy is unique to distributed TOEs. Indeterminacy and state synchrony are related,
9313 and the same solution **may** apply. It is futile to design for indeterminate states; the PP, PP-
9314 Module, functional package or ST author **should** express other requirements in such cases.

9315 EXAMPLE 2

9316 Raise an alarm, audit the event.

9317 **J.12.2 FPT_SSP.1 Simple trusted acknowledgement**

9318 **J.12.2.1 Component rationale and application notes**

9319 In this component, the TSF must supply an acknowledgement to another part of the TSF when
9320 requested. This acknowledgement **should** indicate that one part of a distributed TOE
9321 successfully received an unmodified transmission from a different part of the distributed TOE.

9322 **J.12.3 FPT_SSP.2 Mutual trusted acknowledgement**

9323 **J.12.3.1 Component rationale and application notes**

9324 In this component, in addition to the TSF being able to provide an acknowledgement for the
9325 receipt of a data transmission, the TSF must comply with a request from another part of the TSF
9326 for an acknowledgement to the acknowledgement.

9327 EXAMPLE

9328 The local TSF transmits some data to a remote part of the TSF. The remote part of the TSF acknowledges the
9329 successful receipt of the data and requests that the sending TSF confirm that it receives the acknowledgement. This
9330 mechanism provides additional confidence that both parts of the TSF involved in the data transmission know that the
9331 transmission completed successfully.

9332 **J.13 Time stamps (FPT_STM)**

9333 **J.13.1 User application notes**

9334 This family addresses requirements for a reliable time stamp function within a TOE.

9335 It is the responsibility of the PP, PP-Module, functional package or ST author to clarify the
9336 meaning of the phrase “reliable time stamp”, and to indicate where the responsibility lies in
9337 determining the acceptance of trust.

9338 **J.13.2 FPT_STM.1 Reliable time stamps**

9339 **J.13.2.1 Component rationale and application notes**

9340 Some possible uses of this component include providing reliable time stamps for the purposes
9341 of audit as well as for security attribute expiration.

9342 **J.14 Inter-TSF TSF data consistency (FPT_TDC)**

9343 **J.14.1 User application notes**

9344 In a distributed or composite environment, a TOE **may** need to exchange TSF data with another
9345 trusted IT Product.

9346 EXAMPLE

9347 the SFP-attributes associated with data, audit information, identification information.

9348 This family defines the requirements for sharing and consistent interpretation of these
9349 attributes between the TSF of the TOE and that of a different trusted IT Product.

9350 The components in this family are intended to provide requirements for automated support for
9351 TSF data consistency when such data is transmitted between the TSF of the TOE and another
9352 trusted IT Product. It is also possible that wholly procedural means **could** be used to produce
9353 security attribute consistency, but they are not provided for here.

9354 This family is different from FDP_ETC and FDP_ITC, as those two families are concerned only
9355 with resolving the security attributes between the TSF and its import/export medium.

9356 If the integrity of the TSF data is of concern, requirements **should** be chosen from the Integrity
9357 of exported TSF data (FPT_ITI) family. These components specify requirements for the TSF to
9358 be able to detect or detect and correct modifications to TSF data in transit.

9359 **J.14.2 FPT_TDC.1 Inter-TSF basic TSF data consistency**

9360 **J.14.2.1 Component rationale and application notes**

9361 The TSF is responsible for maintaining the consistency of TSF data used by or associated with
9362 the specified function and that are common between two or more trusted systems.

9363 EXAMPLE

9364 The TSF data of two different systems **may** have different conventions internally. For the TSF data to be used
9365 properly (such as to afford the user data the same protection as within the TOE) by the receiving trusted IT product,
9366 the TOE and the other trusted IT product must use a pre-established protocol to exchange TSF data.

9367 **J.14.2.2 Operations**

9368 In FPT_TDC.1.1, the PP, PP-Module, functional package or ST author **should** define the list of TSF
9369 data types, for which the TSF **shall** provide the capability to consistently interpret, when shared
9370 between the TSF and another trusted IT product.

9371 In FPT_TDC.1.2, the PP, PP-Module, functional package or ST **should** assign the list of
9372 interpretation rules to be applied by the TSF.

9373 **J.15 Testing of external entities (FPT_TEE)**

9374 **J.15.1 User application notes**

9375 This family defines requirements for the testing of one or more external entities by the TSF.
9376 These external entities are not human users, and they **can** include combinations of software
9377 and/or hardware interacting with the TOE.

9378 EXAMPLE

9379 Examples of the types of tests that **may** be run are:

- 9380 a) tests for the presence of a firewall, and possibly whether it is correctly configured;
- 9381 b) tests of some of the properties of the operating system that an application TOE runs on;
- 9382 c) tests of some of the properties of the IC that a smart card OS TOE runs on (such as the random number
9383 generator).

9384 Note The external entity **may** "lie" about the test results, either on purpose or because it is not working
9385 correctly.

9386 These tests **can** be carried out either in some maintenance state, at start-up, on-line, or
9387 continuously. The actions to be taken by the TOE as the result of testing are defined also in this
9388 family.

9389 **J.15.2 Evaluator notes**

9390 The tests of external entities **should** be sufficient to test all of the characteristics of them upon
9391 which the TSF relies.

9392 For FPT_TEE.1 Testing of external entities, It is acceptable for the functions for periodic testing
 9393 to be available only in an off-line or maintenance mode. Controls **should** be in place to limit
 9394 access, during maintenance, to authorized users.

9395 **J.15.3 FPT_TEE.1 Testing of external entities**

9396 **J.15.3.1 Component rationale and application notes**

9397 This component is not intended to be applied to human users.

9398 This component provides support for the periodic testing of properties related to external
 9399 entities upon which the TSF's operation depends, by requiring the ability to periodically invoke
 9400 testing functions.

9401 The PP, PP-Module, functional package or ST author **may** refine the requirement to state
 9402 whether the function **should** be available in off-line, on-line or maintenance mode.

9403 **J.15.3.2 Operations**

9404 In FPT_TEE.1.1, the PP, PP-Module, functional package or ST author **should** specify when the
 9405 TSF will run the testing of external entities, during initial start-up, periodically during normal
 9406 operation, at the request of an authorized user, or under other conditions. If the tests are run
 9407 often, then the end users **should** have more confidence that the TOE is operating correctly than
 9408 if the tests are run less frequently. However, this need for confidence that the TOE is operating
 9409 correctly must be balanced with the potential impact on the availability of the TOE, as often
 9410 times, the testing of external entities **may** delay the normal operation of a TOE.

9411 In FPT_TEE.1.1, the PP, PP-Module, functional package or ST author **should** specify the
 9412 properties of the external entities to be checked by the tests.

9413 EXAMPLE 1

9414 Examples of these properties **may** include configuration or availability properties of a directory server supporting
 9415 some access control part of the TSF.

9416 In FPT_TEE.1.1, the PP, PP-Module, functional package or ST author **should**, if other conditions
 9417 are selected, specify the frequency with which the testing of external entities will be run.

9418 EXAMPLE 2

9419 An example of this other frequency or condition **may** be to run the tests each time a user requests to initiate a session
 9420 with the TOE. For instance, this **could** be the case of testing a directory server before its interaction with the TSF
 9421 during the user authentication process.

9422 In FPT_TEE.1.2, the PP, PP-Module, functional package or ST author **should** specify what are the
 9423 action(s) that the TSF **shall** perform when the testing fails.

9424 EXAMPLE 3

9425 Examples of these action(s), illustrated by a directory server instance, **may** include to connect to an alternative
 9426 available server or otherwise to look for a backup server.

9427 **J.16 Internal TOE TSF data replication consistency (FPT_TRC)**

9428 **J.16.1 User application notes**

9429 The requirements of this family are needed to ensure the consistency of TSF data when such
 9430 data is replicated internal to the TOE. Such data **may** become inconsistent if an internal channel
 9431 between parts of the TOE becomes inoperative. If the TOE is internally structured as a network
 9432 of parts of the TOE, this **can** occur when parts become disabled, network connections are
 9433 broken, and so on.

9434 The method of ensuring consistency is not specified in this component. It **could** be attained
 9435 through a form of transaction logging (where appropriate transactions are "rolled back" to a
 9436 site upon reconnection); it **could** be updating the replicated data through a synchronization

9437 protocol. If a particular protocol is necessary for a PP, PP-Module, functional package or ST, it
9438 **can** be specified through refinement.

9439 It **can** be impossible to synchronize some states, or the cost of such synchronization **can** be too
9440 high.

9441 EXAMPLE

9442 Examples of this situation are communication channel and encryption key revocations.

9443 Indeterminate states **can** also occur; if a specific behaviour is desired, it **should** be specified via
9444 refinement.

9445 **J.16.2 FPT_TRC.1 Internal TSF consistency**

9446 **J.16.2.1 Operations**

9447 In FPT_TRC.1.2, the PP, PP-Module, functional package or ST author **should** specify the list of
9448 functions dependent on TSF data replication consistency.

9449 **J.17 TSF self-test (FPT_TST)**

9450 **J.17.1 User application notes**

9451 The family defines the requirements for the self-testing of the TSF with respect to some
9452 expected correct operation.

9453 EXAMPLE

9454 Examples are interfaces to enforcement functions, and sample arithmetical operations on critical parts of the TOE.

9455 These tests **can** be carried out at start-up, periodically, at the request of an authorized user, or
9456 when other conditions are met. The actions to be taken by the TOE as the result of self-testing
9457 are defined in other families.

9458 The requirements of this family are also needed to detect the corruption of TSF data and TSF
9459 itself (i.e. TSF executable code or TSF hardware component) by various failures that do not
9460 necessarily stop the TOE's operation (which would be handled by other families). These checks
9461 must be performed because these failures **may** not necessarily be prevented. Such failures **can**
9462 occur either because of unforeseen failure modes or associated oversights in the design of
9463 hardware, firmware, or software, or because of malicious corruption of the TSF due to
9464 inadequate logical and/or physical protection.

9465 In addition, use of this component may, with appropriate conditions, help to prevent
9466 inappropriate or damaging TSF changes being applied to an operational TOE as the result of
9467 maintenance activities.

9468 The term "correct operation of the TSF" refers primarily to the operation of the TSF and the
9469 integrity of the TSF data.

9470 **J.17.2 Evaluator notes**

9471 For FPT_TST.1 TSF testing, it is acceptable for the functions that are available to the authorized
9472 user for periodic testing to be available only in an off-line or maintenance mode. Controls
9473 **should** be in place to limit access during these modes to authorized users.

9474 **J.17.3 FPT_TST.1 TSF testing**

9475 **J.17.3.1 Component rationale and application notes**

9476 This component provides support for the testing of the critical functions of the TSF's operation
9477 by requiring the ability to invoke testing functions and check the integrity of TSF data and
9478 executable code.

9479 **J.17.3.2 Operations**

- 9480 In FPT_TST.1.1, the PP, PP-Module, functional package or ST author **should** specify when the
9481 TSF will execute the TSF test; during initial start-up, periodically during normal operation, at
9482 the request of an authorized user, at other conditions. In the case of the latter option, the PP, PP-
9483 Module, functional package or ST author **should** also assign what those conditions are via the
9484 following assignment.
- 9485 In FPT_TST.1.1, the PP, PP-Module, functional package or ST author **should** specify whether the
9486 self-tests are to be carried out to demonstrate the correct operation of the entire TSF, or of only
9487 specified parts of TSF.
- 9488 In FPT_TST.1.1, the PP, PP-Module, functional package or ST author **should**, if selected, specify
9489 the conditions under which the self-test **should** take place.
- 9490 In FPT_TST.1.1, the PP, PP-Module, functional package or ST author **should**, if selected, specify
9491 the list of parts of the TSF that will be subject to TSF self-testing.
- 9492 In FPT_TST.1., the PP, PP-Module, functional package or ST author **should** specify whether data
9493 integrity is to be verified for all TSF data, or only for selected data.
- 9494 In FPT_TST.1., the PP, PP-Module, functional package or ST author **should**, if selected, specify
9495 the list of TSF data that will be verified for integrity.
- 9496 In FPT_TST.1., the PP, PP-Module, functional package or ST author **should** specify whether TSF
9497 integrity is to be verified for all TSF, or only for selected TSF.
- 9498 In FPT_TST.1., the PP, PP-Module, functional package or ST author **should**, if selected, specify
9499 the list of TSF that will be verified for integrity.

Annex K (normative)

Class FRU: Resource utilization- application notes

9504 **K.1 General information**

9505 This class provides three families that support the availability of required resources such as
9506 processing capability and/or storage capacity. The family Fault Tolerance provides protection
9507 against unavailability of capabilities caused by failure of the TOE. The family Priority of Service
9508 ensures that the resources will be allocated to the more important or time-critical tasks and
9509 cannot be monopolized by lower priority tasks. The family Resource Allocation provides limits
9510 on the use of available resources, therefore preventing users from monopolizing the resources.

9511 **K.2 Fault tolerance (FRU_FLT)**

9512 **K.2.1 User application notes**

9513 This family provides requirements for the availability of capabilities even in the case of failures.

9514 EXAMPLE 1

9515 Examples of such failures are power failure, hardware failure, or software error.

9516 In case of these errors, if so specified, the TOE will maintain the specified capabilities.

9517 EXAMPLE 2

9518 The PP, PP-Module, functional package or ST author **could** specify that a TOE used in a nuclear plant will continue the
9519 operation of the shut-down procedure in the case of power-failure or communication-failure

9520 Because the TOE **can** only continue its correct operation if the SFRs are enforced, there is a
9521 requirement that the system must remain in a secure state after a failure. This capability is
9522 provided by FPT_FLS.1 Failure with preservation of secure state.

9523 The mechanisms to provide fault tolerance **could** be active or passive. In case of an active
9524 mechanism, specific functions are in place that are activated in case the error occurs. For
9525 example, a fire alarm is an active mechanism: the TSF will detect the fire and **can** take action
9526 such as switching operation to a backup. In a passive scheme, the architecture of the TOE is
9527 capable of handling the error. For example, the use of a majority voting scheme with multiple
9528 processors is a passive solution; failure of one processor will not disrupt the operation of the
9529 TOE (although it needs to be detected to allow correction).

9530 For this family, it does not matter whether the failure has been initiated accidentally (such as
9531 flooding or unplugging the wrong device) or intentionally (such as monopolizing).

9532 **K.2.2 FRU_FLT.1 Degraded fault tolerance**

9533 **K.2.2.1 Component rationale and application notes**

9534 This component is intended to specify which capabilities the TOE will still provide after a failure
9535 of the system. Since it would be difficult to describe all specific failures, categories of failures
9536 **may** be specified.

9537 EXAMPLE

9538 Examples of general failures are flooding of the computer room, short term power interruption, breakdown of a CPU
9539 or host, software failure, or buffer overflow.

9540 **K.2.2.2 Operations**

9541 In FRU_FLT.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
9542 TOE capabilities the TOE will maintain during and after a specified failure.

9543 In FRU_FLT.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
 9544 types of failures against which the TOE has to be explicitly protected. If a failure in this list
 9545 occurs, the TOE will be able to continue its operation.

9546 **K.2.3 FRU_FLT.2 Limited fault tolerance**

9547 **K.2.3.1 Component rationale and application notes**

9548 This component is intended to specify against what type of failures the TOE must be resistant.
 9549 Since it would be difficult to describe all specific failures, categories of failures **may** be specified.

9550 EXAMPLE

9551 Examples of general failures are flooding of the computer room, short term power interruption, breakdown of a CPU
 9552 or host, software failure, or overflow of buffer.

9553 **K.2.3.2 Operations**

9554 In FRU_FLT.2.1, the PP, PP-Module, functional package or ST author **should** specify the list of
 9555 types of failures against which the TOE has to be explicitly protected. If a failure in this list
 9556 occurs, the TOE will be able to continue its operation.

9557 **K.3 Priority of service (FRU_PRS)**

9558 **K.3.1 User application notes**

9559 The requirements of this family allow the TSF to control the use of resources under the control
 9560 of the TSF by users and subjects such that high priority activities under the control of the TSF
 9561 will always be accomplished without interference or delay due to low priority activities. In
 9562 other words, time critical tasks will not be delayed by tasks that are less time critical.

9563 This family **could** be applicable to several types of resources.

9564 EXAMPLE

9565 Processing capacity, and communication channel capacity.

9566 The Priority of Service mechanism might be passive or active. In a passive Priority of Service
 9567 system, the system will select the task with the highest priority when given a choice between
 9568 two waiting applications. While using passive Priority of Service mechanisms, when a low
 9569 priority task is running, it cannot be interrupted by a high priority task. While using an active
 9570 Priority of Service mechanisms, lower priority tasks might be interrupted by new high priority
 9571 tasks.

9572 The audit requirement states that all reasons for rejection **should** be audited. It is left to the
 9573 developer to argue that an operation is not rejected but delayed.

9574 **K.3.2 FRU_PRS.1 Limited priority of service**

9575 **K.3.2.1 Component rationale and application notes**

9576 This component defines priorities for a subject, and the resources for which this priority will be
 9577 used. If some subject attempts to act on a resource controlled by the Priority of Service
 9578 requirements, the access and/or time of access will be dependent on the subject's priority, the
 9579 priority of the currently acting subject, and the priority of the subjects still in the queue.

9580 **K.3.2.2 Operations**

9581 In FRU_PRS.1.2, the PP, PP-Module, functional package or ST author **should** specify the list of
 9582 controlled resources for which the TSF enforces priority of service

9583 EXAMPLE

9584 Resources such as processes, disk space, memory, bandwidth.

9585 **K.3.3 FRU_PRS.2 Full priority of service**

9586 **K.3.3.1 Component rationale and application notes**

9587 This component defines priorities for a subject. All shareable resources under the control of the
9588 TSF will be subjected to the Priority of Service mechanism. If some subject attempts to take
9589 action on a shareable TSF resource, the access and/or time of access will be dependent on the
9590 subject's priority, the priority of the currently acting subject, and the priority of the subjects still
9591 in the queue.

9592 **K.4 Resource allocation (FRU_RSA)**

9593 **K.4.1 User application notes**

9594 The requirements of this family allow the TSF to control the use of resources under the control
9595 of the TSF by users and subjects such that unauthorized denial of service will not take place by
9596 means of monopolization of resources by other users or subjects.

9597 Resource allocation rules allow the creation of quotas or other means of defining limits on the
9598 amount of resource space or time that **may** be allocated on behalf of a specific user or subjects.

9599 EXAMPLE 1

9600 These rules may, for example:

- 9601 — Provide for object quotas that constrain the number and/or size of objects a specific user may allocate;
- 9602 — Control the allocation/deallocation of preassigned resource units where these units are under the control of
9603 the TSF.

9604 In general, these functions will be implemented through the use of attributes assigned to users
9605 and resources.

9606 The objective of these components is to ensure a certain amount of fairness among the users
9607 and subjects.

9608 EXAMPLE 2

9609 A single user **should** not allocate all the available space

9610 Since resource allocation often goes beyond the lifespan of a subject (i.e. files often exist longer
9611 than the applications that generated them), and multiple instantiations of subjects by the same
9612 user **should** not negatively affect other users too much, the components allow that the
9613 allocation limits are related to the users. In some situations, the resources are allocated by a
9614 subject.

9615 EXAMPLE 3

9616 Main memory or CPU cycles.

9617 In those instances, the components allow that the resource allocation be on the level of subjects.

9618 This family imposes requirements on resource allocation, not on the use of the resource itself.
9619 The audit requirements therefore, as stated, also apply to the allocation of the resource, not to
9620 the use of the resource.

9621 **K.4.2 FRU_RSA.1 Maximum quotas**

9622 **K.4.2.1 Component rationale and application notes**

9623 This component provides requirements for quota mechanisms that apply to only a specified set
9624 of the shareable resources in the TOE. The requirements allow the quotas to be associated with
9625 a user, possibly assigned to groups of users or subjects as applicable to the TOE.

9626 **K.4.2.2 Operations**

9627 In FRU_RSA.1.1, the PP, PP-Module, functional package or ST author **should** specify the list of
9628 controlled resources for which maximum resource allocation limits are required.

9629 EXAMPLE

- 9630 Examples of controlled resources include processes, disk space, memory, and bandwidth.
- 9631 If all resources under the control of the TSF need to be included, the words “all TSF resources”
9632 **may** be specified.
- 9633 In FRU_RSA.1.1, the PP, PP-Module, functional package or ST author **should** select whether the
9634 maximum quotas apply to individual users, to a defined group of users, or subjects or any
9635 combination of these.
- 9636 In FRU_RSA.1.1, the PP, PP-Module, functional package or ST author **should** select whether the
9637 maximum quotas are applicable to any given time (simultaneously), or over a specific time
9638 interval.
- 9639 **K.4.3 FRU_RSA.2 Minimum and maximum quotas**
- 9640 **K.4.3.1 Component rationale and application notes**
- 9641 This component provides requirements for quota mechanisms that apply to a specified set of
9642 the shareable resources in the TOE. The requirements allow the quotas to be associated with a
9643 user, or possibly assigned to groups of users as applicable to the TOE.
- 9644 **K.4.3.2 Operations**
- 9645 In FRU_RSA.2.1, the PP, PP-Module, functional package or ST author **should** specify the
9646 controlled resources for which maximum and minimum resource allocation limits are required.
- 9647 If all resources under the control of the TSF need to be included, the words “all TSF resources”
9648 **can** be specified.
- 9649 In FRU_RSA.2.2, the PP, PP-Module, functional package or ST author specifies the controlled
9650 resources for which a minimum allocation limit needs to be set.
- 9651 If all resources under the control of the TSF need to be included the words “all TSF resources”
9652 **can** be specified.
- 9653 EXAMPLE
- 9654 Examples of controlled resources include processes, disk space, memory, and bandwidth.
- 9655 In FRU_RSA.2.1, the PP, PP-Module, functional package or ST author **should** select whether the
9656 maximum quotas apply to individual users, to a defined group of users, or subjects or any
9657 combination of these.
- 9658 In FRU_RSA.2.1, the PP, PP-Module, functional package or ST author **should** select whether the
9659 maximum quotas are applicable to any given time (simultaneously), or over a specific time
9660 interval.
- 9661 In FRU_RSA.2.2, the PP, PP-Module, functional package or ST author selects whether the
9662 minimum quotas apply to individual users, to a defined group of users, or subjects or any
9663 combination of these.
- 9664 In FRU_RSA.2.2, the PP, PP-Module, functional package or ST author selects whether the
9665 minimum quotas are applicable to any given time (simultaneously), or over a specific time
9666 interval.

Annex L (normative)

Class FTA: TOE access- application notes

9667
9668
9669
9670

9671 **L.1 General information**

9672 The establishment of a user's session typically consists of the creation of one or more subjects
9673 that perform operations in the TOE on behalf of the user. At the end of the session
9674 establishment procedure, provided the TOE access requirements are satisfied, the created
9675 subjects bear the attributes determined by the identification and authentication functions. This
9676 family specifies functional requirements for controlling the establishment of a user's session.

9677 A user session is defined as the period starting at the time of the identification/authentication,
9678 or if more appropriate, the start of an interaction between the user and the system, up to the
9679 moment that all subjects (resources and attributes) related to that session have been
9680 deallocated.

9681 **L.2 Limitation on scope of selectable attributes (FTA_LSA)**

9682 **L.2.1 User application notes**

9683 This family defines requirements that will limit the session security attributes a user **may** select,
9684 and the subjects to which a user **may** be bound, based on: the method of access; the location or
9685 port of access; and/or the time.

9686 EXAMPLE 1

9687 Time-of-day, day-of-week.

9688 This family provides the capability for a PP, PP-Module, functional package or ST author to
9689 specify requirements for the TSF to place limits on the domain of an authorized user's security
9690 attributes based on an environmental condition.

9691 EXAMPLE 2

9692 A user **could** be allowed to establish a "secret session" during normal business hours but outside those hours the
9693 same user **could** be constrained to only establishing "unclassified sessions".

9694 The identification of relevant constraints on the domain of selectable attributes **may** be
9695 achieved through the use of the selection operation. These constraints **may** be applied on an
9696 attribute-by-attribute basis. When there exists a need to specify constraints on multiple
9697 attributes this component will have to be replicated for each attribute.

9698 EXAMPLE 3

9699 Examples of attributes that **could** be used to limit the session security attributes are:

- 9700 — The method of access can be used to specify in which type of environment the user will be operating (such
9701 as file transfer protocol, terminal, vtam).
- 9702 — The location of access can be used to constrain the domain of a user's selectable attributes based on a user's
9703 location or port of access. This capability is of particular use in environments where dial-up facilities or
9704 network facilities are available.
- 9705 — The time of access can be used to constrain the domain of a user's selectable attributes. For example, ranges
9706 may be based upon time-of-day, day-of-week, or calendar dates. This constraint provides some operational
9707 protection against user actions that **could** occur at a time where proper monitoring or where proper
9708 procedural measures may not be in place.

9709 **L.2.2 FTA_LSA.1 Limitation on scope of selectable attributes**

9710 **L.2.2.1 Operations**

9711 In FTA_LSA.1.1, the PP, PP-Module, functional package or ST author specifies the set of session
9712 security attributes that are to be constrained.

9713 EXAMPLE 1

9714 Examples of these session security attributes are user clearance level, integrity level and roles.

9715 In FTA_LSA.1.1, the PP, PP-Module, functional package or ST author specifies the set of
9716 attributes that **can** be used to determine the scope of the session security attributes.

9717 EXAMPLE 2

9718 Examples of such attributes are user identity, originating location, time of access, and method of access.

9719 **L.3 Limitation on multiple concurrent sessions (FTA_MCS)**

9720 **L.3.1 User application notes**

9721 This family defines how many sessions a user **may** have at the same time (concurrent sessions).
9722 This number of concurrent sessions **may** either be set for a group of users or for each individual
9723 user.

9724 **L.3.2 FTA_MCS.1 Basic limitation on multiple concurrent sessions**

9725 **L.3.2.1 Component rationale and application notes**

9726 This component allows the system to limit the number of sessions in order to effectively use the
9727 resources of the TOE.

9728 **L.3.2.2 Operations**

9729 In FTA_MCS.1.2, the PP, PP-Module, functional package or ST author specifies the default
9730 number of maximum concurrent sessions to be used.

9731 **L.3.3 FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions**

9732 **L.3.3.1 Component rationale and application notes**

9733 This component provides additional capabilities over those of FTA_MCS.1 Basic limitation on
9734 multiple concurrent sessions, by allowing further constraints to be placed on the number of
9735 concurrent sessions that users are able to invoke. These constraints are in terms of a user's
9736 security attributes, such as a user's identity, or membership of a role.

9737 **L.3.3.2 Operations**

9738 In FTA_MCS.2.1, the PP, PP-Module, functional package or ST author specifies the rules that
9739 determine the maximum number of concurrent sessions.

9740 EXAMPLE

9741 An example of a rule is "maximum number of concurrent sessions is one if the user has a classification level of
9742 "secret" and five otherwise".

9743 In FTA_MCS.2.2, the PP, PP-Module, functional package or ST author specifies the default
9744 number of maximum concurrent sessions to be used.

9745 **L.4 Session locking and termination (FTA_SSL)**

9746 **L.4.1 User application notes**

9747 This family defines requirements for the TSF to provide the capability for TSF-initiated and
9748 user-initiated locking, unlocking, and termination of interactive sessions.

9749 When a user is directly interacting with subjects in the TOE (interactive session), the user's
9750 terminal is vulnerable if left unattended. This family provides requirements for the TSF to
9751 disable (lock) the terminal or terminate the session after a specified period of inactivity, and for
9752 the user to initiate the disabling (locking) of the terminal or terminate the session. To reactivate

9753 the terminal, an event specified by the PP, PP-Module, functional package or ST author, such as
9754 the user re-authentication must occur.

9755 A user is considered inactive, if he/she has not provided any stimulus to the TOE for a specified
9756 period of time.

9757 PP, PP-Module, functional package or ST authors consider whether FTP_TRP.1 Trusted path
9758 **should** be included. In that case, the function “session locking” **must** be included in the
9759 operation in FTP_TRP.1 Trusted path.

9760 **L.4.2 FTA_SSL.1 TSF-initiated session locking**

9761 **L.4.2.1 Component rationale and application notes**

9762 FTA_SSL.1 TSF-initiated session locking, provides the capability for the TSF to lock an active
9763 user session after a specified period of time. Locking a terminal would prevent any further
9764 interaction with an existing active session through the use of the locked terminal.

9765 If display devices are overwritten, the replacement contents need not be static (i.e. “screen
9766 savers” are permitted).

9767 This component allows the PP, PP-Module, functional package or ST author to specify what
9768 events will unlock the session. These events **may** be related to the terminal, the user, or time.

9769 EXAMPLE

9770 Examples of events include

9771 — Terminal related: a fixed set of keystrokes to unlock the session.

9772 — User related: reauthentication.

9773 — Time related: after 15 minutes.

9774 **L.4.2.2 Operations**

9775 In FTA_SSL.1.1, the PP, PP-Module, functional package or ST author specifies the interval of user
9776 inactivity that will trigger the locking of an interactive session. If so desired the PP, PP-Module,
9777 functional package or ST author **could**, through the assignment, specify that the time interval is
9778 left to the authorized administrator or the user. The management functions in the FMT class **can**
9779 specify the capability to modify this time interval, making it the default value.

9780 In FTA_SSL.1.2, the PP, PP-Module, functional package or ST author specifies the event(s) that
9781 **should** occur before the session is unlocked.

9782 EXAMPLE

9783 Examples of such an event are: “user re-authentication” or “user enters unlock key-sequence”.

9784 **L.4.3 FTA_SSL.2 User-initiated locking**

9785 **L.4.3.1 Component rationale and application notes**

9786 FTA_SSL.2 User-initiated locking, provides the capability for an authorized user to lock and
9787 unlock his/her own interactive session. This would provide authorized users with the ability to
9788 effectively block further use of their active sessions without having to terminate the active
9789 session.

9790 If devices are overwritten, the replacement contents need not be static (i.e. “screen savers” are
9791 permitted).

9792 **L.4.3.2 Operations**

9793 In FTA_SSL.2.2, the PP, PP-Module, functional package or ST author specifies the event(s) that
9794 **must** occur before the session is unlocked.

9795 EXAMPLE

9796 Examples of such an event are: “user re-authentication”, or “user enters unlock key-sequence”.

9797 **L.4.4 FTA_SSL.3 TSF-initiated termination**

9798 **L.4.4.1 Component rationale and application notes**

9799 FTA_SSL.3 TSF-initiated termination, requires that the TSF terminate an interactive user
9800 session after a period of inactivity.

9801 The PP, PP-Module, functional package or ST author **should** be aware that a session **may**
9802 continue after the user terminated his/her activity. This requirement would terminate this
9803 background subject after a period of inactivity of the user without regard to the status of the
9804 subject.

9805 EXAMPLE

9806 An example of a continuing session after a user terminated activity is background processing.

9807 **L.4.4.2 Operations**

9808 In FTA_SSL.3.1, the PP, PP-Module, functional package or ST author specifies the interval of user
9809 inactivity that will trigger the termination of an interactive session. If so desired, the PP, PP-
9810 Module, functional package or ST author **could**, through the assignment, specify that the interval
9811 is left to the authorized administrator or the user. The management functions in the FMT class
9812 **can** specify the capability to modify this time interval, making it the default value.

9813 **L.4.5 FTA_SSL.4 User-initiated termination**

9814 **L.4.5.1 Component rationale and application notes**

9815 FTA_SSL.4 User-initiated termination, provides the capability for an authorized user to
9816 terminate his/her interactive session.

9817 The PP, PP-Module, functional package or ST author **should** be aware that a session **could**
9818 continue after the user terminated his/her activity.

9819 EXAMPLE

9820 An example of a continuing session after a user terminated activity is background processing.

9821 This requirement would allow the user to terminate this background subject without regard to
9822 the status of the subject.

9823 **L.5 TOE access banners (FTA_TAB)**

9824 **L.5.1 User application notes**

9825 Prior to identification and authentication, TOE access requirements provide the ability for the
9826 TOE to display an advisory warning message to potential users pertaining to appropriate use of
9827 the TOE.

9828 **L.5.2 FTA_TAB.1 Default TOE access banners**

9829 **L.5.2.1 Component rationale and application notes**

9830 This component requires that there is an advisory warning regarding the unauthorized use of
9831 the TOE. A PP, PP-Module, functional package or ST author **could** refine the requirement to
9832 include a default banner.

9833 **L.6 TOE access history (FTA_TAH)**

9834 **L.6.1 User application notes**

9835 This family defines requirements for the TSF to display to users, upon successful session
9836 establishment to the TOE, a history of unsuccessful attempts to access the account. This history
9837 **could** include the date, time, means of access, and port of the last successful access to the TOE,

9838 as well as the number of unsuccessful attempts to access the TOE since the last successful
9839 access by the identified user.

9840 **L.6.2 FTA_TAH.1 TOE access history**

9841 **L.6.2.1 Component rationale and application notes**

9842 This family **can** provide authorized users with information that **could** indicate the possible
9843 misuse of their user account.

9844 This component requests that the user is presented with the information. The user **should** be
9845 able to review the information but is not forced to do so.

9846 EXAMPLE

9847 A user might create scripts that ignore this information and start other processes.

9848 **L.6.2.2 Operations**

9849 In FTA_TAH.1.1, the PP, PP-Module, functional package or ST author selects the security
9850 attributes of the last successful session establishment that will be shown at the user interface.
9851 The items are: date, time, method of access, and/or location.

9852 In FTA_TAH.1.2, the PP, PP-Module, functional package or ST author selects the security
9853 attributes of the last unsuccessful session establishment that will be shown at the user
9854 interface. The items are: date, time, method of access, and/or location.

9855 EXAMPLE

9856 Method of access: ftp.

9857 Location: terminal 50.

9858 **L.7 TOE session establishment (FTA_TSE)**

9859 **L.7.1 User application notes**

9860 This family defines requirements to deny a user permission to establish a session with the TOE
9861 based on attributes such as the location or port of access, the user's security attribute, ranges of
9862 time or combinations of parameters.

9863 EXAMPLE 1

9864 Security attribute: identity, clearance level, integrity level, membership in a role.

9865 Ranges of time: time-of-day, day-of-week, calendar dates.

9866 This family provides the capability for the PP, PP-Module, functional package or ST author to
9867 specify requirements for the TOE to place constraints on the ability of an authorized user to
9868 establish a session with the TOE. The identification of relevant constraints **can** be achieved
9869 through the use of the selection operation.

9870 EXAMPLE 2

9871 Examples of attributes that **could** be used to specify the session establishment constraints are:

- 9872 a) The location of access can be used to constrain the ability of a user to establish an active session with the
9873 TOE, based on the user's location or port of access. This capability is of particular use in environments
9874 where dial-up facilities or network facilities are available.
- 9875 b) The user's security attributes can be used to place constraints on the ability of a user to establish an active
9876 session with the TOE. For example, these attributes would provide the capability to deny session
9877 establishment based on any of the following:
- 9878 — a user's identity;
 - 9879 — a user's clearance level;
 - 9880 — a user's integrity level; and
 - 9881 — a user's membership in a role.

9882 This capability is particularly relevant in situations where authorization or login may take place at a different
9883 location from where TOE access checks are performed.

9884 c) The time of access can be used to constrain the ability of a user to establish an active session with the TOE
9885 based on ranges of time. For example, ranges may be based upon time-of-day, day-of-week, or calendar
9886 dates. This constraint provides some operational protection against actions that **could** occur at a time
9887 where proper monitoring or where proper procedural measures may not be in place.

9888 **L.7.2 FTA_TSE.1 TOE session establishment**

9889 **L.7.2.1 Operations**

9890 In FTA_TSE.1.1, the PP, PP-Module, functional package or ST author specifies the attributes that
9891 **can** be used to restrict the session establishment.

9892 EXAMPLE

9893 Examples of possible attributes are user identity, originating location (such as no remote terminals), time of access
9894 (such as outside hours), or method of access (such as telnet).

Annex M (normative)

Class FTP: Trusted path/channels- application notes

9899 **M.1 General information**

9900 Users often need to perform functions through direct interaction with the TSF. A trusted path
9901 provides confidence that a user is communicating directly with the TSF whenever it is invoked.
9902 A user's response via the trusted path guarantees that untrusted applications cannot intercept
9903 or modify the user's response. Similarly, trusted channels are one approach for secure
9904 communication between the TSF and another trusted IT product.

9905 Absence of a trusted path **may** allow breaches of accountability or access control in
9906 environments where untrusted applications are used. These applications **can** intercept user-
9907 private information, such as passwords, and use it to impersonate other users. As a
9908 consequence, responsibility for any system actions cannot be reliably assigned to an
9909 accountable entity. Also, these applications **could** output erroneous information on an
9910 unsuspecting user's display, resulting in subsequent user actions that **could** be erroneous and
9911 **could** lead to a security breach.

9912 **M.2 Inter-TSF trusted channel (FTP_ITC)**

9913 **M.2.1 User application notes**

9914 This family defines the rules for the creation of a trusted channel connection that goes between
9915 the TSF and another trusted IT product for the performance of security critical operations
9916 between the products.

9917 EXAMPLE

9918 An example of such a security critical operation is the updating of the TSF authentication database by the transfer of
9919 data from a trusted product whose function is the collection of audit data.

9920 **M.2.2 FTP_ITC.1 Inter-TSF trusted channel**

9921 **M.2.2.1 Component rationale and application notes**

9922 This component is used when a trusted communication channel between the TSF and another
9923 trusted IT product is required.

9924 **M.2.2.2 Operations**

9925 In FTP_ITC.1.2, the PP, PP-Module, functional package or ST author must specify whether the
9926 local TSF, another trusted IT product, or both **shall** have the capability to initiate the trusted
9927 channel.

9928 In FTP_ITC.1.3, the PP, PP-Module, functional package or ST author specifies the functions for
9929 which a trusted channel is required.

9930 EXAMPLE

9931 Examples of these functions **may** include transfer of user, subject, and/or object security attributes and ensuring
9932 consistency of TSF data.

9933 **M.3 Trusted channel protocol (FTP_PRO)**

9934 **M.3.1 User application notes**

9935 This family defines the rules for the creation of a trusted channel connection that goes between
9936 the TSF and another trusted IT product for the protection of data transfers. In contrast with
9937 FTP_ITC or FTP_TRP, FTP_PRO is concerned with security details of the protocol used for a

9938 channel and provides a focus for protocol properties that might otherwise be split between a
 9939 larger number of separate SFRs. It can improve clarity of a PP/ST by highlighting mechanisms
 9940 within the protocol that may then be linked to cryptographic functions described in other SFRs
 9941 (such as FCS_COP.1).

9942 The components of FTP_PRO are not hierarchical but are intended to be used together to
 9943 separately specify different aspects of a trusted channel, such as its confidentiality and integrity
 9944 protection features.

9945 There is no dependency from FTP_PRO.2 to FTP_PRO.3 because any mechanisms for security of
 9946 the shared secret establishment will be part of the mechanism described in FTP_PRO.2 itself.

9947 In cases where some cryptographic operations used in the trusted channel protocol are
 9948 performed outside the TOE, FTP_PRO.2 and/or FTP_PRO.3 might be omitted from an ST, and the
 9949 ST author would then need to supply a rationale for the unsatisfied dependencies between
 9950 FTP_PRO components.

9951 Separate iterations of the relevant FTP_PRO components may be used for different channels
 9952 where the completion of the SFR needs to be different for each channel. In general, each
 9953 separate iteration will need to include all three components with its own dependencies'
 9954 rationale.

9955 **M.3.2 FTP_PRO.1 Trusted channel protocol**

9956 **M.3.2.1 Component rationale and application notes**

9957 Where values used in the completion of FTP_PRO operations have dependencies between
 9958 different SFR elements, these need to be made clear in the instantiation of the SFR.

9959 EXAMPLE

9960 A table could be given in which the columns represent the relevant selections and assignments, and the rows define
 9961 the valid combination of completion values.

9962 **M.3.2.2 Operations**

9963 In FTP_PRO.1.1, if selected, the PP, PP-Module, functional package or ST author **should** specify a
 9964 trusted channel protocol and the defined protocol roles.

9965 EXAMPLE 1

9966 Examples of "defined protocol roles" would be 'client' or 'server' (TLS), 'initiator' or 'responder' (IKEv2/IPsec), 'Trust
 9967 Center' (ZigBee) or 'Key Distribution Centre' (Kerberos).

9968 In FTP_PRO.1.2 the first assignment is intended to state rules for when the trusted channel is
 9969 required to be used by the TOE, such as mandating its use for communications with an audit
 9970 server. This assignment may take the value 'None specified' (also with 'None specified' in the
 9971 second assignment) if no specific uses of the channel are mandated for the TOE.

9972 In FTP_PRO.1.5 the assignment is intended to state rules related to implementation of the
 9973 protocol(s). It may take the value 'None specified' if no rules are required, or if the standards
 9974 referenced in other elements of the SFR include the relevant rules and no specific evaluator
 9975 check is required for the context in which the SFR is being used.

9976 EXAMPLE 2

9977 Rules include those for maximum packet sizes or rekey intervals

9978 In FTP_PRO.1.6 the assignment is intended to state rules related to negotiable aspects of the
 9979 protocol, when intending to narrow the options provided by the TOE compared to the standard
 9980 that defines the protocol.

9981 EXAMPLE 3

9982 Specification of acceptable ciphersuites or acceptable older protocol versions.

9983 The assignment **may** take the value 'None specified' if no rules are required. Where the
 9984 assignment is completed with a list then that list specifies the only configurations permitted –

9985 any other configuration would be a violation of the SFR. This element may be used to specify
9986 mandatory supported configurations without limiting the TOE to using these configurations by,
9987 for example, listing the required configurations with “(support required)” after each entry in
9988 the list and then including a final element which states that any other configuration permitted
9989 by the standard is allowed.

9990 In FTP_PRO.1.3 the PP, PP-Module, functional package or ST author selects which entity is
9991 allowed to initiate the establishment of the trusted channel.

9992 **M.3.3 FTP_PRO.2 Trusted channel establishment**

9993 **M.3.3.1 Component rationale and application notes**

9994 In FTP_PRO.2, the ‘list of rules for carrying out the authentication’ may be used to limit available
9995 parameters for the authentication mechanisms.

9996 EXAMPLE

9997 Rules might be stated for the format (e.g. FQDN or IP address, use of wildcards) or prioritization of identifiers when
9998 alternative sources of an identifier are available in the authentication data exchanged.

9999 **M.3.3.2 Operations**

10000 In FTP_PRO.2.2 the selection indicating the direction of the authentication should be chosen.

10001 In FTP_PRO.2.1 The PP, PP-Module, functional package or ST author provides a list of key
10002 establishment mechanisms.

10003 In FTP_PRO.2.2 the assignments include providing a list of authentication mechanisms used
10004 during the authentication and a list of rules used during the authentication.

10005 **M.3.4 FTP_PRO.3 Trusted channel data protection**

10006 **M.3.4.1 Component rationale and application notes**

10007 The FTP_PRO.3 component addresses protection (confidentiality and integrity) of data in
10008 transit through a trusted channel.

10009 **M.3.4.2 Operations**

10010 The PP, PP-Module, functional package or ST author selects the attacks that are mitigated by the
10011 TSF.

10012 The PP, PP-Module, functional package or ST author completes the assignment by specifying
10013 lists of encryption and integrity protection mechanisms.

10014 EXAMPLE

10015 Examples of integrity protection mechanism include protection of contents and file-system permissions of system
10016 files and directories; protection of processes against code injection, and protection against unsigned kernel
10017 extensions.

10018 **M.4 Trusted path (FTP_TRP)**

10019 **M.4.1 User application notes**

10020 This family defines the requirements to establish and maintain trusted communication to or
10021 from users and the TSF. A trusted path **may** be required for any security-relevant interaction.
10022 Trusted path exchanges **may** be initiated by a user during an interaction with the TSF, or the
10023 TSF **may** establish communication with the user via a trusted path.

10024 **M.4.2 FTP_TRP.1 Trusted path**

10025 **M.4.2.1 Component rationale and application notes**

10026 This component is used when trusted communication between a user and the TSF is required,
10027 either for initial authentication purposes only or for additional specified user operations.

10028 **M.4.2.2 Operations**

10029 In FTP_TRP.1.1, the PP, PP-Module, functional package or ST author specifies whether the
10030 trusted path must be extended to remote and/or local users.

10031 In FTP_TRP.1.1, the PP, PP-Module, functional package or ST author specifies whether the
10032 trusted path shall protect the data from modification, disclosure, and/or other types of integrity
10033 or confidentiality violation.

10034 In FTP_TRP.1.1, if selected, the PP, PP-Module, functional package or ST author identifies any
10035 additional types of integrity or confidentiality violation against which the trusted path shall
10036 protect the data.

10037 In FTP_TRP.1.2, the PP, PP-Module, functional package or ST author specifies whether the TSF,
10038 local users, and/or remote users are able to initiate the trusted path.

10039 In FTP_TRP.1.3, the PP, PP-Module, functional package or ST author specifies whether the
10040 trusted path is to be used for initial user authentication and/or for other specified services.

10041 In FTP_TRP.1.3, if selected, the PP, PP-Module, functional package or ST author identifies other
10042 services for which trusted path is required, if any.