

COMMITTEE DRAFT ISO/IEC 2nd CD 15408-2, revision		Reference document: SC 27 N18804	
Date: 2019-01-07		Supersedes document N18701	
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 technology - Security techniques Secretariat: Germany (DIN)	Circulated to P- and O-members, and to technical committees and organizations in liaison for comments by: 2019-03-05 Please submit your comments via the online balloting application by the due date indicated.		
ISO/IEC 2nd 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).
2nd CD Consideration In accordance with Recommendation 14 (see SC 27 N18820 = WG 3 N1610) of the 57th SC 27/WG 3 meeting held in Gjøvik, Norway, 2018-09-30/10-04 the hereby attached document is being circulated for a 8-week 2nd CD letter ballot closing by <b style="text-align: center;">2019-03-05 Medium: http://isotc.iso.org/livelink/livelink/open/jtc1sc27 No. of pages: 2 + 288			

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
20

ISO/IEC JTC 1/SC 27 N18804
ISO/IEC JTC 1/SC 27/WG 3 N1650
Date: 2018-12-21
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.

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.....	xx
35	Introduction.....	xxi
36	1 Scope.....	1
37	2 Normative references	1
38	3 Terms and Definitions	1
39	4 Overview.....	1
40	4.1 <i>Organization of this document.....</i>	<i>2</i>
41	5 Functional requirements paradigm.....	2
42	6 Security functional components	6
43	6.1 <i>Overview.....</i>	<i>6</i>
44	6.1.1 Class structure.....	6
45	6.1.2 Family structure.....	7
46	6.1.3 Component structure	8
47	6.2 <i>Component catalogue.....</i>	<i>10</i>
48	6.2.1 Component changes highlighting.....	11
49	7 Class FAU: Security audit.....	12
50	7.1 <i>Class description.....</i>	<i>12</i>
51	7.2 <i>Security audit automatic response (FAU_ARP).....</i>	<i>12</i>
52	7.2.1 Family behaviour.....	12
53	7.2.2 Components leveling and description.....	12
54	7.2.3 Management of FAU_ARP.1.....	13
55	7.2.4 Audit of FAU_ARP.1.....	13
56	7.2.5 FAU_ARP.1 Security alarms.....	13
57	7.3 <i>Security audit data generation (FAU_GEN).....</i>	<i>13</i>
58	7.3.1 Family behaviour.....	13
59	7.3.2 Components leveling and description.....	13
60	7.3.3 Management of FAU_GEN.1, FAU_GEN.2.....	14
61	7.3.4 Audit of FAU_GEN.1, FAU_GEN.2	14
62	7.3.5 FAU_GEN.1 Audit data generation	14
63	7.3.6 FAU_GEN.2 User identity association.....	14
64	7.4 <i>Security audit analysis (FAU_SAA).....</i>	<i>14</i>
65	7.4.1 Family behaviour.....	14
66	7.4.2 Components leveling and description.....	15
67	7.4.3 Management of FAU_SAA.1	15
68	7.4.4 Management of FAU_SAA.2	15
69	7.4.5 Management of FAU_SAA.3	15

70	7.4.6	Management of FAU_SAA.4	15
71	7.4.7	Audit of FAU_SAA.1, FAU_SAA.2, FAU_SAA.3, FAU_SAA.4	16
72	7.4.8	FAU_SAA.1 Potential violation analysis.....	16
73	7.4.9	FAU_SAA.2 Profile based anomaly detection.....	16
74	7.4.10	FAU_SAA.3 Simple attack heuristics.....	17
75	7.4.11	FAU_SAA.4 Complex attack heuristics.....	17
76	7.5	<i>Security audit review (FAU_SAR)</i>	17
77	7.5.1	Family behaviour.....	17
78	7.5.2	Components leveling and description.....	18
79	7.5.3	Management of FAU_SAR.1	18
80	7.5.4	Management of FAU_SAR.2, FAU_SAR.3.....	18
81	7.5.5	Audit of FAU_SAR.1	18
82	7.5.6	Audit of FAU_SAR.2	18
83	7.5.7	Audit of FAU_SAR.3	18
84	7.5.8	FAU_SAR.1 Audit review	18
85	7.5.9	FAU_SAR.2 Restricted audit review.....	19
86	7.5.10	FAU_SAR.3 Selectable audit review.....	19
87	7.6	<i>Security audit event selection (FAU_SEL)</i>	19
88	7.6.1	Family behaviour.....	19
89	7.6.2	Components leveling and description.....	19
90	7.6.3	Management of FAU_SEL.1	19
91	7.6.4	Audit of FAU_SEL.1.....	20
92	7.6.5	FAU_SEL.1 Selective audit.....	20
93	7.7	<i>Security audit data storage (FAU_STG)</i>	21
94	7.7.1	Family behaviour.....	21
95	7.7.2	Components leveling and description.....	21
96	7.7.3	Management of FAU_STG.1.....	21
97	7.7.4	Management of FAU_STG.2.....	21
98	7.7.5	Management of FAU_STG.3.....	21
99	7.7.6	Management of FAU_STG.4.....	21
100	7.7.7	Management of FAU_STG.5.....	22
101	7.7.8	Audit of FAU_STG.1	22
102	7.7.9	Audit of FAU_STG.2, FAU_STG.3.....	22
103	7.7.10	Audit of FAU_STG.4	22
104	7.7.11	Audit of FAU_STG.5	22
105	7.7.12	FAU_STG.1 Audit data storage location.....	22
106	7.7.13	FAU_STG.2 Protected audit data storage.....	22
107	7.7.14	FAU_STG.3 Guarantees of audit data availability.....	23
108	7.7.15	FAU_STG.4 Action in case of possible audit data loss.....	23
109	7.7.16	FAU_STG.5 Prevention of audit data loss.....	23

110	8	Class FCO: Communication	24
111	8.1	<i>Class description</i>	24
112	8.2	<i>Non-repudiation of origin (FCO_NRO)</i>	24
113	8.2.1	Family behaviour	24
114	8.2.2	Components leveling and description	24
115	8.2.3	Management of FCO_NRO.1, FCO_NRO.2	24
116	8.2.4	Audit of FCO_NRO.1.....	25
117	8.2.5	Audit of FCO_NRO.2.....	25
118	8.2.6	FCO_NRO.1 Selective proof of origin.....	25
119	8.2.7	FCO_NRO.2 Enforced proof of origin.....	25
120	8.3	<i>Non-repudiation of receipt (FCO_NRR)</i>	26
121	8.3.1	Family behaviour.....	26
122	8.3.2	Components leveling and description	26
123	8.3.3	Management of FCO_NRR.1, FCO_NRR.2.....	26
124	8.3.4	Audit of FCO_NRR.1.....	26
125	8.3.5	Audit of FCO_NRR.2.....	26
126	8.3.6	FCO_NRR.1 Selective proof of receipt.....	27
127	8.3.7	FCO_NRR.2 Enforced proof of receipt.....	27
128	9	Class FCS: Cryptographic support	28
129	9.1	<i>Class description</i>	28
130	9.2	<i>Cryptographic key management (FCS_CKM)</i>	28
131	9.2.1	Family behaviour	28
132	9.2.2	Components leveling and description	29
133	9.2.3	Management of FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.5, CKM.6.....	29
134	9.2.4	Audit of FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.5, CKM.6.....	29
135	9.2.5	FCS_CKM.1 Cryptographic key generation.....	29
136	9.2.6	FCS_CKM.2 Cryptographic key distribution.....	30
137	9.2.7	FCS_CKM.3 Cryptographic key access.....	30
138	9.2.8	FCS_CKM.4 Cryptographic key destruction.....	31
139	9.2.9	FCS_CKM.5 Cryptographic key derivation.....	31
140	9.2.10	FCS_CKM.6 Timing and event of cryptographic key destruction.....	31
141	9.3	<i>Cryptographic operation (FCS_COP)</i>	31
142	9.3.1	Family behaviour.....	31
143	9.3.2	Components leveling and description	32
144	9.3.3	Management of FCS_COP.1	32
145	9.3.4	Audit of FCS_COP.1.....	32
146	9.3.5	FCS_COP.1 Cryptographic operation.....	32
147	9.4	<i>Random bit generation (FCS_RBG)</i>	32
148	9.4.1	Family behaviour.....	32
149	9.4.2	Components leveling and description	33

150	9.4.3	Management of FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5, FCS_RBG.6	33
151	9.4.4	Audit of FCS_RBG.1, FCS_RBG.2.....	33
152	9.4.5	Audit of FCS_RBG.3, FCS_RBG.4, FCS_RBG.6, FCS_RBG.6	33
153	9.4.6	FCS_RBG.1 Random bit generation (RBG).....	34
154	9.4.7	FCS_RBG.2 Random bit generation (external seeding).....	34
155	9.4.8	FCS_RBG.3 Random bit generation (internal seeding – single source)	35
156	9.4.9	FCS_RBG.4 Random bit generation (internal seeding – multiple sources)	35
157	9.4.10	FCS_RBG.5 Random bit generation (combining entropy sources)	35
158	9.4.11	FCS_RBG.6 Random bit generation service.....	35
159	9.5	<i>Generation of random numbers (FCS_RNG)</i>	<i>36</i>
160	9.5.1	Family behaviour	36
161	9.5.2	Components leveling and description	36
162	9.5.3	Management of FCS_RNG.1.....	36
163	9.5.4	Audit of FCS_RNG.1	36
164	9.5.5	FCS_RNG.1 Random number generation	36
165	10	Class FDP: User data protection	37
166	10.1	<i>Class description</i>	<i>37</i>
167	10.2	<i>Access control policy (FDP_ACC)</i>	<i>38</i>
168	10.2.1	Family behaviour.....	38
169	10.2.2	Management of FDP_ACC.1, FDP_ACC.2.....	39
170	10.2.3	Audit of FDP_ACC.1, FDP_ACC.2.....	39
171	10.2.4	FDP_ACC.1 Subset access control.....	39
172	10.2.5	FDP_ACC.2 Complete access control	39
173	10.3	<i>Access control functions (FDP_ACF)</i>	<i>40</i>
174	10.3.1	Family behaviour	40
175	10.3.2	Components leveling and description.....	40
176	10.3.3	Management of FDP_ACF.1.....	40
177	10.3.4	Audit of FDP_ACF.1	40
178	10.3.5	FDP_ACF.1 Security attribute-based access control	40
179	10.4	<i>Data authentication (FDP_DAU)</i>	<i>41</i>
180	10.4.1	Family behaviour	41
181	10.4.2	Components leveling and description	41
182	10.4.3	Management of FDP_DAU.1, FDP_DAU.2	41
183	10.4.4	Audit of FDP_DAU.1	42
184	10.4.5	Audit of FDP_DAU.2	42
185	10.4.6	FDP_DAU.1 Basic Data Authentication	42
186	10.4.7	FDP_DAU.2 Data Authentication with Identity of Guarantor.....	42
187	10.5	<i>Export from the TOE (FDP_ETC).....</i>	<i>43</i>
188	10.5.1	Family behaviour.....	43
189	10.5.2	Components leveling and description.....	43

190	10.5.3	Management of FDP_ETC.1.....	43
191	10.5.4	Management of FDP_ETC.2.....	43
192	10.5.5	Audit of FDP_ETC.1, FDP_ETC.2.....	43
193	10.5.6	FDP_ETC.1 Export of user data without security attributes.....	43
194	10.5.7	FDP_ETC.2 Export of user data with security attributes.....	44
195	10.6	<i>Information flow control policy (FDP_IFC).....</i>	44
196	10.6.1	Family behaviour.....	44
197	10.6.2	Components leveling and description.....	44
198	10.6.3	Management of FDP_IFC.1, FDP_IFC.2.....	45
199	10.6.4	Audit of FDP_IFC.1, FDP_IFC.2.....	45
200	10.6.5	FDP_IFC.1 Subset information flow control.....	45
201	10.6.6	FDP_IFC.2 Complete information flow control.....	45
202	10.7	<i>Information flow control functions (FDP_IFF).....</i>	45
203	10.7.1	Family behaviour.....	45
204	10.7.2	Components leveling and description.....	46
205	10.7.3	Management of FDP_IFF.1, FDP_IFF.2.....	46
206	10.7.4	Management of FDP_IFF.3, FDP_IFF.4, FDP_IFF.5.....	46
207	10.7.5	Management of FDP_IFF.6.....	46
208	10.7.6	Audit of FDP_IFF.1, FDP_IFF.2, FDP_IFF.5.....	47
209	10.7.7	Audit of FDP_IFF.3, FDP_IFF.4, FDP_IFF.6.....	47
210	10.7.8	FDP_IFF.1 Simple security attributes.....	47
211	10.7.9	FDP_IFF.2 Hierarchical security attributes.....	48
212	10.7.10	FDP_IFF.3 Limited illicit information flows.....	49
213	10.7.11	FDP_IFF.4 Partial elimination of illicit information flows.....	49
214	10.7.12	FDP_IFF.5 No illicit information flows.....	49
215	10.7.13	FDP_IFF.6 Illicit information flow monitoring.....	49
216	10.8	<i>Information Retention Control (FDP_IRC).....</i>	50
217	10.8.1	Family behaviour.....	50
218	10.8.2	Components leveling and description.....	50
219	10.8.3	Management of FDP_IRC.1.....	50
220	10.8.4	Audit of FDP_IRC.1.....	51
221	10.8.5	FDP_IRC.1 Information retention control.....	51
222	10.9	<i>Import from outside of the TOE (FDP_ITC).....</i>	51
223	10.9.1	Family behaviour.....	51
224	10.9.2	Components leveling and description.....	51
225	10.9.3	Management of FDP_ITC.1, FDP_ITC.2.....	51
226	10.9.4	Audit of FDP_ITC.1, FDP_ITC.2.....	52
227	10.9.5	FDP_ITC.1 Import of user data without security attributes.....	52
228	10.9.6	FDP_ITC.2 Import of user data with security attributes.....	52
229	10.10	<i>Internal TOE transfer (FDP_ITT).....</i>	53

230	10.10.1	Family behaviour.....	53
231	10.10.2	Components leveling and description.....	53
232	10.10.3	Management of FDP_ITT.1, FDP_ITT.2.....	53
233	10.10.4	Management of FDP_ITT.3, FDP_ITT.4.....	53
234	10.10.5	Audit of FDP_ITT.1, FDP_ITT.2.....	54
235	10.10.6	Audit of FDP_ITT.3, FDP_ITT.4.....	54
236	10.10.7	FDP_ITT.1 Basic internal transfer protection.....	54
237	10.10.8	FDP_ITT.2 Transmission separation by attribute.....	54
238	10.10.9	FDP_ITT.3 Integrity monitoring.....	55
239	10.10.10	FDP_ITT.4 Attribute-based integrity monitoring.....	55
240	10.11	<i>Residual information protection (FDP_RIP).....</i>	<i>55</i>
241	10.11.1	Family behaviour.....	55
242	10.11.2	Components leveling and description.....	55
243	10.11.3	Management of FDP_RIP.1, FDP_RIP.2.....	56
244	10.11.4	Audit of FDP_RIP.1, FDP_RIP.2.....	56
245	10.11.5	FDP_RIP.1 Subset residual information protection.....	56
246	10.11.6	FDP_RIP.2 Full residual information protection.....	56
247	10.12	<i>Rollback (FDP_ROL).....</i>	<i>56</i>
248	10.12.1	Family behaviour.....	56
249	10.12.2	Components leveling and description.....	57
250	10.12.3	Management of FDP_ROL.1, FDP_ROL.2.....	57
251	10.12.4	Audit of FDP_ROL.1, FDP_ROL.2.....	57
252	10.12.5	FDP_ROL.1 Basic rollback.....	57
253	10.12.6	FDP_ROL.2 Advanced rollback.....	57
254	10.13	<i>Stored data confidentiality (FDP_SDC).....</i>	<i>58</i>
255	10.13.1	Family behaviour.....	58
256	10.13.2	Components leveling and description.....	58
257	10.13.3	Management of FDP_SDC.1, FDP_SDC.2.....	58
258	10.13.4	Audit of FDP_SDC.1, FDP_SDC.2.....	58
259	10.13.5	FDP_SDC.1 Stored data confidentiality.....	58
260	10.13.6	FDP_SDC.2 Stored data confidentiality with dedicated method.....	59
261	10.14	<i>Stored data integrity (FDP_SDI).....</i>	<i>59</i>
262	10.14.1	Family behaviour.....	59
263	10.14.2	Components leveling and description.....	59
264	10.14.3	Management of FDP_SDI.1.....	59
265	10.14.4	Management of FDP_SDI.2.....	59
266	10.14.5	Audit of FDP_SDI.1.....	60
267	10.14.6	Audit of FDP_SDI.2.....	60
268	10.14.7	FDP_SDI.1 Stored data integrity monitoring.....	60
269	10.14.8	FDP_SDI.2 Stored data integrity monitoring and action.....	60

270	10.15	<i>Inter-TSF user data confidentiality transfer protection (FDP_UCT)</i>	60
271	10.15.1	Family behaviour.....	60
272	10.15.2	Components leveling and description.....	61
273	10.15.3	Management of FDP_UCT.1.....	61
274	10.15.4	Audit of FDP_UCT.1.....	61
275	10.15.5	FDP_UCT.1 Basic data exchange confidentiality.....	61
276	10.16	<i>Inter-TSF user data integrity transfer protection (FDP_UIT)</i>	61
277	10.16.1	Family behaviour.....	61
278	10.16.2	Components leveling and description.....	62
279	10.16.3	Management of FDP_UIT.1, FDP_UIT.2, FDP_UIT.3.....	62
280	10.16.4	Audit of FDP_UIT.1.....	62
281	10.16.5	Audit of FDP_UIT.2, FDP_UIT.3.....	62
282	10.16.6	FDP_UIT.1 Data exchange integrity.....	63
283	10.16.7	FDP_UIT.2 Source data exchange recovery.....	63
284	10.16.8	FDP_UIT.3 Destination data exchange recovery.....	63
285	11	Class FIA: Identification and authentication	64
286	11.1	<i>Class description</i>	64
287	11.2	<i>Authentication failures (FIA_AFL)</i>	65
288	11.2.1	Family behaviour.....	65
289	11.2.2	Components leveling and description.....	65
290	11.2.3	Management of FIA_AFL.1.....	65
291	11.2.4	Audit of FIA_AFL.1.....	65
292	11.2.5	FIA_AFL.1 Authentication failure handling.....	65
293	11.3	<i>Authentication proof of identity (FIA_API)</i>	66
294	11.3.1	Family behaviour.....	66
295	11.3.2	Components leveling and description.....	66
296	11.3.3	Management of FIA_API.1.....	66
297	11.3.4	Audit of FIA_API.1.....	66
298	11.3.5	FIA_API.1 Authentication proof of identity.....	66
299	11.4	<i>User attribute definition (FIA_ATD)</i>	66
300	11.4.1	Family behaviour.....	66
301	11.4.2	Components leveling and description.....	66
302	11.4.3	Management of FIA_ATD.1.....	67
303	11.4.4	Audit of FIA_ATD.1.....	67
304	11.4.5	FIA_ATD.1 User attribute definition.....	67
305	11.5	<i>Specification of secrets (FIA_SOS)</i>	67
306	11.5.1	Family behaviour.....	67
307	11.5.2	Components leveling and description.....	67
308	11.5.3	Management of FIA_SOS.1.....	67
309	11.5.4	Management of FIA_SOS.2.....	67

310	11.5.5	Audit of FIA_SOS.1, FIA_SOS.2	68
311	11.5.6	FIA_SOS.1 Verification of secrets.....	68
312	11.5.7	FIA_SOS.2 TSF Generation of secrets	68
313	11.6	<i>User authentication (FIA_UAU)</i>	68
314	11.6.1	Family behaviour	68
315	11.6.2	Components leveling and description.....	69
316	11.6.3	Management of FIA_UAU.1	69
317	11.6.4	Management of FIA_UAU.2	69
318	11.6.5	Management of FIA_UAU.3, FIA_UAU.4, FIA_UAU.7	69
319	11.6.6	Management of FIA_UAU.5.....	70
320	11.6.7	Management of FIA_UAU.6.....	70
321	11.6.8	Management of FIA_UAU.7	70
322	11.6.9	Audit of FIA_UAU.1.....	70
323	11.6.10	Audit of FIA_UAU.2.....	70
324	11.6.11	Audit of FIA_UAU.3.....	70
325	11.6.12	Audit of FIA_UAU.4.....	70
326	11.6.13	Audit of FIA_UAU.5.....	70
327	11.6.14	Audit of FIA_UAU.6.....	70
328	11.6.15	Audit of FIA_UAU.7.....	71
329	11.6.16	FIA_UAU.1 Timing of authentication.....	71
330	11.6.17	FIA_UAU.2 User authentication before any action	71
331	11.6.18	FIA_UAU.3 Unforgeable authentication.....	71
332	11.6.19	FIA_UAU.4 Single-use authentication mechanisms.....	72
333	11.6.20	FIA_UAU.5 Multiple authentication mechanisms.....	72
334	11.6.21	FIA_UAU.6 Re-authenticating	72
335	11.6.22	FIA_UAU.7 Protected authentication feedback.....	72
336	11.7	<i>User identification (FIA_UID)</i>	73
337	11.7.1	Family behaviour	73
338	11.7.2	Components leveling and description.....	73
339	11.7.3	Management of FIA_UID.1	73
340	11.7.4	Management of FIA_UID.2	73
341	11.7.5	Audit of FIA_UID.1, FIA_UID.2.....	73
342	11.7.6	FIA_UID.1 Timing of identification.....	73
343	11.7.7	FIA_UID.2 User identification before any action	74
344	11.8	<i>User-subject binding (FIA_USB)</i>	74
345	11.8.1	Family behaviour	74
346	11.8.2	Components leveling and description.....	74
347	11.8.3	Management of FIA_USB.1.....	74
348	11.8.4	Audit of FIA_USB.1.....	74
349	11.8.5	FIA_USB.1 User-subject binding	74

350	12	Class FMT: Security management.....	76
351	12.1	<i>Class description.....</i>	76
352	12.2	<i>Limited capabilities and availability (FMT_LIM).....</i>	77
353	12.2.1	Family behaviour.....	77
354	12.2.2	Components leveling and description.....	77
355	12.2.3	Management of FMT_LIM.1, FMT_LIM.2.....	77
356	12.2.4	Audit of FMT_LIM.1.....	77
357	12.2.5	FMT_LIM.1 Limited capabilities.....	77
358	12.2.6	FMT_LIM.2 Limited availability.....	78
359	12.3	<i>Management of functions in TSF (FMT_MOF).....</i>	78
360	12.3.1	Family behaviour.....	78
361	12.3.2	Components leveling and description.....	78
362	12.3.3	Management of FMT_MOF.1.....	78
363	12.3.4	Audit of FMT_MOF.1.....	78
364	12.3.5	FMT_MOF.1 Management of security functions behaviour.....	78
365	12.4	<i>Management of security attributes (FMT_MSA).....</i>	79
366	12.4.1	Family behaviour.....	79
367	12.4.2	Components leveling and description.....	79
368	12.4.3	Management of FMT_MSA.1.....	79
369	12.4.4	Management of FMT_MSA.2.....	79
370	12.4.5	Management of FMT_MSA.3.....	79
371	12.4.6	Management of FMT_MSA.4.....	79
372	12.4.7	Audit of FMT_MSA.1.....	80
373	12.4.8	Audit of FMT_MSA.2.....	80
374	12.4.9	Audit of FMT_MSA.3.....	80
375	12.4.10	Audit of FMT_MSA.4.....	80
376	12.4.11	FMT_MSA.1 Management of security attributes.....	80
377	12.4.12	FMT_MSA.2 Secure security attributes.....	80
378	12.4.13	FMT_MSA.3 Static attribute initialization.....	81
379	12.4.14	FMT_MSA.4 Security attribute value inheritance.....	81
380	12.5	<i>Management of TSF data (FMT_MTD).....</i>	81
381	12.5.1	Family behaviour.....	81
382	12.5.2	Components leveling and description.....	81
383	12.5.3	Management of FMT_MTD.1.....	82
384	12.5.4	Management of FMT_MTD.2.....	82
385	12.5.5	Management of FMT_MTD.3.....	82
386	12.5.6	Audit of FMT_MTD.1.....	82
387	12.5.7	Audit of FMT_MTD.2.....	82
388	12.5.8	Audit of FMT_MTD.3.....	82
389	12.5.9	FMT_MTD.1 Management of TSF data.....	82

390	12.5.10	FMT_MTD.2 Management of limits on TSF data.....	83
391	12.5.11	FMT_MTD.3 Secure TSF data.....	83
392	12.6	<i>Revocation (FMT_REV)</i>	83
393	12.6.1	Family behaviour.....	83
394	12.6.2	Components leveling and description.....	83
395	12.6.3	Management of FMT_REV.1.....	83
396	12.6.4	Audit of FMT_REV.1.....	84
397	12.6.5	FMT_REV.1 Revocation.....	84
398	12.7	<i>Security attribute expiration (FMT_SAE)</i>	84
399	12.7.1	Family behaviour.....	84
400	12.7.2	Components leveling and description.....	84
401	12.7.3	Management of FMT_SAE.1.....	84
402	12.7.4	Audit of FMT_SAE.1.....	84
403	12.7.5	FMT_SAE.1 Time-limited authorization.....	85
404	12.8	<i>Specification of Management Functions (FMT_SMF)</i>	85
405	12.8.1	Family behaviour.....	85
406	12.8.2	Components leveling and description.....	85
407	12.8.3	Management of FMT_SMF.1.....	85
408	12.8.4	Audit of FMT_SMF.1.....	85
409	12.8.5	FMT_SMF.1 Specification of Management Functions.....	86
410	12.9	<i>Security management roles (FMT_SMR)</i>	86
411	12.9.1	Family behaviour.....	86
412	12.9.2	Components leveling and description.....	86
413	12.9.3	Management of FMT_SMR.1.....	86
414	12.9.4	Management of FMT_SMR.2.....	86
415	12.9.5	Management of FMT_SMR.3.....	86
416	12.9.6	Audit of FMT_SMR.1.....	86
417	12.9.7	Audit of FMT_SMR.2.....	87
418	12.9.8	Audit of FMT_SMR.3.....	87
419	12.9.9	FMT_SMR.1 Security roles.....	87
420	12.9.10	FMT_SMR.2 Restrictions on security roles.....	87
421	12.9.11	FMT_SMR.3 Assuming roles.....	87
422	13	Class FPR: Privacy	88
423	13.1	<i>Class description</i>	88
424	13.2	<i>Anonymity (FPR_ANO)</i>	88
425	13.2.1	Family behaviour.....	88
426	13.2.2	Components leveling and description.....	88
427	13.2.3	Management of FPR_ANO.1, FPR_ANO.2.....	89
428	13.2.4	Audit of FPR_ANO.1, FPR_ANO.2.....	89
429	13.2.5	FPR_ANO.1 Anonymity.....	89

430	13.2.6	FPR_ANO.2 Anonymity without soliciting information	89
431	13.3	<i>Pseudonymity (FPR_PSE)</i>	89
432	13.3.1	Family behaviour	89
433	13.3.2	Components leveling and description.....	89
434	13.3.3	Management of FPR_PSE.1, FPR_PSE.2, FPR_PSE.3.....	90
435	13.3.4	Audit of FPR_PSE.1, FPR_PSE.2, FPR_PSE.3	90
436	13.3.5	FPR_PSE.1 Pseudonymity	90
437	13.3.6	FPR_PSE.2 Reversible pseudonymity	90
438	13.3.7	FPR_PSE.3 Alias pseudonymity.....	91
439	13.4	<i>Unlinkability (FPR_UNL)</i>	91
440	13.4.1	Family behaviour.....	91
441	13.4.2	Components leveling and description.....	91
442	13.4.3	Management of FPR_UNL.1	92
443	13.4.4	Audit of FPR_UNL.1.....	92
444	13.4.5	FPR_UNL.1 Unlinkability of operations.....	92
445	13.5	<i>Unobservability (FPR_UNO)</i>	92
446	13.5.1	Family behaviour.....	92
447	13.5.2	Components leveling and description.....	92
448	13.5.3	Management of FPR_UNO.1, FPR_UNO.2	93
449	13.5.4	Management of FPR_UNO.3.....	93
450	13.5.5	Management of FPR_UNO.4.....	93
451	13.5.6	Audit of FPR_UNO.1, FPR_UNO.2	93
452	13.5.7	Audit of FPR_UNO.3	93
453	13.5.8	Audit of FPR_UNO.4	93
454	13.5.9	FPR_UNO.1 Unobservability.....	93
455	13.5.10	FPR_UNO.2 Allocation of information impacting unobservability.....	94
456	13.5.11	FPR_UNO.3 Unobservability without soliciting information	94
457	13.5.12	FPR_UNO.4 Authorized user observability	94
458	14	Class FPT: Protection of the TSF	95
459	14.1	<i>Class description</i>	95
460	14.2	<i>TOE emanation (FPT_EMS)</i>	96
461	14.2.1	Family behaviour	96
462	14.2.2	Components leveling and description.....	97
463	14.2.3	Management of FPT_EMS.1	97
464	14.2.4	Audit of FPT_EMS.1.....	97
465	14.2.5	FPT_EMS.1 Emanation of TSF and User data.....	97
466	14.3	<i>Fail secure (FPT_FLS)</i>	98
467	14.3.1	Family behaviour	98
468	14.3.2	Components leveling and description.....	98
469	14.3.3	Management of FPT_FLS.1.....	98

470	14.3.4	Audit of FPT_FLS.1	98
471	14.3.5	FPT_FLS.1 Failure with preservation of secure state	98
472	14.4	<i>TSF initialization (FPT_INI)</i>	99
473	14.4.1	Family behaviour	99
474	14.4.2	Components leveling and description	99
475	14.4.3	Management of FPT_INI.1.....	99
476	14.4.4	Audit of FPT_INI.1.....	99
477	14.4.5	FPT_INI.1 TSF initialization.....	99
478	14.5	<i>Availability of exported TSF data (FPT_ITA)</i>	100
479	14.5.1	Family behaviour	100
480	14.5.2	Components leveling and description	100
481	14.5.3	Management of FPT_ITA.1.....	100
482	14.5.4	Audit of FPT_ITA.1.....	100
483	14.5.5	FPT_ITA.1 Inter-TSF availability within a defined availability metric.....	100
484	14.6	<i>Confidentiality of exported TSF data (FPT_ITC)</i>	100
485	14.6.1	Family behaviour	100
486	14.6.2	Components leveling and description	100
487	14.6.3	Management of FPT_ITC.1.....	101
488	14.6.4	Audit of FPT_ITC.1.....	101
489	14.6.5	FPT_ITC.1 Inter-TSF confidentiality during transmission.....	101
490	14.7	<i>Integrity of exported TSF data (FPT_ITI)</i>	101
491	14.7.1	Family behaviour	101
492	14.7.2	Components leveling and description	101
493	14.7.3	Management of FPT_ITI.1.....	101
494	14.7.4	Management of FPT_ITI.2.....	102
495	14.7.5	Audit of FPT_ITI.1.....	102
496	14.7.6	Audit of FPT_ITI.2.....	102
497	14.7.7	FPT_ITI.1 Inter-TSF detection of modification	102
498	14.7.8	FPT_ITI.2 Inter-TSF detection and correction of modification	102
499	14.8	<i>Internal TOE TSF data transfer (FPT_ITT)</i>	103
500	14.8.1	Family behaviour	103
501	14.8.2	Components leveling and description	103
502	14.8.3	Management of FPT_ITT.1.....	103
503	14.8.4	Management of FPT_ITT.2.....	103
504	14.8.5	Management of FPT_ITT.3.....	103
505	14.8.6	Audit of FPT_ITT.1, FPT_ITT.2.....	104
506	14.8.7	Audit of FPT_ITT.3.....	104
507	14.8.8	FPT_ITT.1 Basic internal TSF data transfer protection	104
508	14.8.9	FPT_ITT.2 TSF data transfer separation	104
509	14.8.10	FPT_ITT.3 TSF data integrity monitoring.....	104

510	14.9	<i>TSF physical protection (FPT_PHP)</i>	105
511	14.9.1	Family behaviour.....	105
512	14.9.2	Components leveling and description.....	105
513	14.9.3	Management of FPT_PHP.1.....	105
514	14.9.4	Management of FPT_PHP.2.....	105
515	14.9.5	Management of FPT_PHP.3.....	106
516	14.9.6	Audit of FPT_PHP.1.....	106
517	14.9.7	Audit of FPT_PHP.2.....	106
518	14.9.8	Audit of FPT_PHP.3.....	106
519	14.9.9	FPT_PHP.1 Passive detection of physical attack.....	106
520	14.9.10	FPT_PHP.2 Notification of physical attack.....	106
521	14.9.11	FPT_PHP.3 Resistance to physical attack.....	107
522	14.10	<i>Trusted recovery (FPT_RCV)</i>	107
523	14.10.1	Family behaviour.....	107
524	14.10.2	Components leveling and description.....	107
525	14.10.3	Management of FPT_RCV.1.....	107
526	14.10.4	Management of FPT_RCV.2, FPT_RCV.3.....	108
527	14.10.5	Management of FPT_RCV.4.....	108
528	14.10.6	Audit of FPT_RCV.1, FPT_RCV.2, FPT_RCV.3.....	108
529	14.10.7	Audit of FPT_RCV.4.....	108
530	14.10.8	FPT_RCV.1 Manual recovery.....	108
531	14.10.9	FPT_RCV.2 Automated recovery.....	108
532	14.10.10	FPT_RCV.3 Automated recovery without undue loss.....	109
533	14.10.11	FPT_RCV.4 Function recovery.....	109
534	14.11	<i>Replay detection (FPT_RPL)</i>	109
535	14.11.1	Family behaviour.....	109
536	14.11.2	Components leveling and description.....	109
537	14.11.3	Management of FPT_RPL.1.....	110
538	14.11.4	Audit of FPT_RPL.1.....	110
539	14.11.5	FPT_RPL.1 Replay detection.....	110
540	14.12	<i>State synchrony protocol (FPT_SSP)</i>	110
541	14.12.1	Family behaviour.....	110
542	14.12.2	Components leveling and description.....	110
543	14.12.3	Management of FPT_SSP.1, FPT_SSP.2.....	111
544	14.12.4	Audit of FPT_SSP.1, FPT_SSP.2.....	111
545	14.12.5	FPT_SSP.1 Simple trusted acknowledgement.....	111
546	14.12.6	FPT_SSP.2 Mutual trusted acknowledgement.....	111
547	14.13	<i>Time stamps (FPT_STM)</i>	111
548	14.13.1	Family behaviour.....	111
549	14.13.2	Components leveling and description.....	111

550	14.13.3	Management of FPT_STM.1	112
551	14.13.4	Management of FPT_STM.2	112
552	14.13.5	Audit of FPT_STM.1	112
553	14.13.6	Audit of FPT_STM.2	112
554	14.13.7	FPT_STM.1 Reliable time stamps	112
555	14.13.8	FPT_STM.2 Time source	112
556	14.14	<i>Inter-TSF TSF data consistency (FPT_TDC)</i>	113
557	14.14.1	Family behaviour	113
558	14.14.2	Components leveling and description	113
559	14.14.3	Management of FPT_TDC.1	113
560	14.14.4	Audit of FPT_TDC.1	113
561	14.14.5	FPT_TDC.1 Inter-TSF basic TSF data consistency	113
562	14.15	<i>Testing of external entities (FPT_TEE)</i>	114
563	14.15.1	Family behaviour	114
564	14.15.2	Components leveling and description	114
565	14.15.3	Management of FPT_TEE.1	114
566	14.15.4	Audit of FPT_TEE.1	114
567	14.15.5	FPT_TEE.1 Testing of external entities	114
568	14.16	<i>Internal TOE TSF data replication consistency (FPT_TRC)</i>	114
569	14.16.1	Family behaviour	114
570	14.16.2	Components leveling and description	115
571	14.16.3	Management of FPT_TRC.1	115
572	14.16.4	Audit of FPT_TRC.1	115
573	14.16.5	FPT_TRC.1 Internal TSF consistency	115
574	14.17	<i>TSF self-test (FPT_TST)</i>	115
575	14.17.1	Family behaviour	115
576	14.17.2	Components leveling and description	116
577	14.17.3	Management of FPT_TST.1	116
578	14.17.4	Audit of FPT_TST.1	116
579	14.17.5	FPT_TST.1 TSF self-testing	116
580	15	Class FRU: Resource utilization	118
581	15.1	<i>Class description</i>	118
582	15.2	<i>Fault tolerance (FRU_FLT)</i>	118
583	15.2.1	Family behaviour	118
584	15.2.2	Components leveling and description	118
585	15.2.3	Management of FRU_FLT.1, FRU_FLT.2	119
586	15.2.4	Audit of FRU_FLT.1	119
587	15.2.5	Audit of FRU_FLT.2	119
588	15.2.6	FRU_FLT.1 Degraded fault tolerance	119
589	15.2.7	FRU_FLT.2 Limited fault tolerance	119

590	15.3	<i>Priority of service (FRU_PRS)</i>	119
591	15.3.1	Family behaviour.....	119
592	15.3.2	Components leveling and description.....	119
593	15.3.3	Management of FRU_PRS.1, FRU_PRS.2.....	120
594	15.3.4	Audit of FRU_PRS.1, FRU_PRS.2.....	120
595	15.3.5	FRU_PRS.1 Limited priority of service.....	120
596	15.3.6	FRU_PRS.2 Full priority of service.....	120
597	15.4	<i>Resource allocation (FRU_RSA)</i>	120
598	15.4.1	Family behaviour.....	120
599	15.4.2	Components leveling and description.....	121
600	15.4.3	Management of FRU_RSA.1.....	121
601	15.4.4	Management of FRU_RSA.2.....	121
602	15.4.5	Audit of FRU_RSA.1, FRU_RSA.2.....	121
603	15.4.6	FRU_RSA.1 Maximum quotas.....	121
604	15.4.7	FRU_RSA.2 Minimum and maximum quotas.....	121
605	16	Class FTA: TOE access	123
606	16.1	<i>Class description</i>	123
607	16.2	<i>Limitation on scope of selectable attributes (FTA_LSA)</i>	123
608	16.2.1	Family behaviour.....	123
609	16.2.2	Components leveling and description.....	123
610	16.2.3	Management of FTA_LSA.1.....	124
611	16.2.4	Audit of FTA_LSA.1.....	124
612	16.2.5	FTA_LSA.1 Limitation on scope of selectable attributes.....	124
613	16.3	<i>Limitation on multiple concurrent sessions (FTA_MCS)</i>	124
614	16.3.1	Family behaviour.....	124
615	16.3.2	Components leveling and description.....	124
616	16.3.3	Management of FTA_MCS.1.....	124
617	16.3.4	Management of FTA_MCS.2.....	125
618	16.3.5	Audit of FTA_MCS.1, FTA_MCS.2.....	125
619	16.3.6	FTA_MCS.1 Basic limitation on multiple concurrent sessions.....	125
620	16.3.7	FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions.....	125
621	16.4	<i>Session locking and termination (FTA_SSL)</i>	125
622	16.4.1	Family behaviour.....	125
623	16.4.2	Components leveling and description.....	126
624	16.4.3	Management of FTA_SSL.1.....	126
625	16.4.4	Management of FTA_SSL.2.....	126
626	16.4.5	Management of FTA_SSL.3.....	126
627	16.4.6	Management of FTA_SSL.4.....	126
628	16.4.7	Audit of FTA_SSL.1, FTA_SSL.2.....	126
629	16.4.8	Audit of FTA_SSL.3.....	127

630	16.4.9	Audit of FTA_SSL.4	127
631	16.4.10	FTA_SSL.1 TSF-initiated session locking	127
632	16.4.11	FTA_SSL.2 User-initiated locking	127
633	16.4.12	FTA_SSL.3 TSF-initiated termination.....	128
634	16.4.13	FTA_SSL.4 User-initiated termination.....	128
635	16.5	<i>TOE access banners (FTA_TAB)</i>	128
636	16.5.1	Family behaviour	128
637	16.5.2	Components leveling and description.....	128
638	16.5.3	Management of FTA_TAB.1	128
639	16.5.4	Audit of FTA_TAB.1	128
640	16.5.5	FTA_TAB.1 Default TOE access banners	128
641	16.6	<i>TOE access history (FTA_TAH)</i>	129
642	16.6.1	Family behaviour	129
643	16.6.2	Components leveling and description.....	129
644	16.6.3	Management of FTA_TAH.1.....	129
645	16.6.4	Audit of FTA_TAH.1.....	129
646	16.6.5	FTA_TAH.1 TOE access history	129
647	16.7	<i>TOE session establishment (FTA_TSE)</i>	130
648	16.7.1	Family behaviour.....	130
649	16.7.2	Components leveling and description.....	130
650	16.7.3	Management of FTA_TSE.1.....	130
651	16.7.4	Audit of FTA_TSE.1.....	130
652	16.7.5	FTA_TSE.1 TOE session establishment.....	130
653	17	Class FTP: Trusted path/channels.....	131
654	17.1	<i>Class description</i>	131
655	17.2	<i>Inter-TSF trusted channel (FTP_ITC)</i>	132
656	17.2.1	Family behaviour.....	132
657	17.2.2	Components leveling and description.....	132
658	17.2.3	Management of FTP_ITC.1.....	132
659	17.2.4	Audit of FTP_ITC.1.....	132
660	17.2.5	FTP_ITC.1 Inter-TSF trusted channel.....	132
661	17.3	<i>Secure channel (FTP_PRO)</i>	133
662	17.3.1	Family behavior	133
663	17.3.2	Components leveling and description.....	133
664	17.3.3	Management of FTP_PRO.1	133
665	17.3.4	Audit of FTP_PRO.1	133
666	17.3.5	FTP_PRO.1 Secure channel protocol	134
667	17.3.6	FTP_PRO.2 Secure channel establishment.....	134
668	17.3.7	FTP_PRO.3 Secure channel data protection.....	135
669	17.4	<i>Trusted path (FTP_TRP)</i>	135

670	17.4.1	Family behaviour.....	135
671	17.4.2	Components leveling and description.....	135
672	17.4.3	Management of FTP_TRP.1.....	136
673	17.4.4	Audit of FTP_TRP.1.....	136
674	17.4.5	FTP_TRP.1 Trusted path.....	136
675		Annex A (normative) Security functional requirements structure of the application notes.....	137
676		Annex B (informative) Dependency tables for security functional components.....	140
677		Annex C (normative) Class FAU: Security audit - application notes.....	151
678		Annex D (normative) Class FCO: Communication- application notes.....	164
679		Annex E (normative) Class FCS: Cryptographic support- application notes.....	169
680		Annex F (normative) Class FDP: User data protection- application notes.....	179
681		Annex G (normative) Class FIA: Identification and authentication- application notes.....	205
682		Annex H (normative) Class FMT: Security management- application notes.....	214
683		Annex I (normative) Class FPR: Privacy- application notes.....	223
684		Annex J (normative) Class FPT: Protection of the TSF- application notes.....	235
685		Annex K (normative) Class FRU: Resource utilization- application notes.....	252
686		Annex L (normative) Class FTA: TOE access- application notes.....	257
687		Annex M (normative) Class FTP: Trusted path/channels- application notes.....	263
688			

689 **Foreword**

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

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

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

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

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

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

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

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

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

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

- 722 — The document has been revised to comply with ISO/IEC Directives
 - 723 — Technical changes have been introduced:
 - 724 ○ New security functional components have been introduced
- 725

726 Introduction

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

733 Security functional components express security requirements intended to counter threats in the
734 assumed operating environment of the TOE and/or cover any identified organizational security
735 policies.

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

- 740 a) Consumers, who use this document when selecting components to express functional
741 requirements which satisfy the security objectives expressed in a PP or ST.
742 ISO/IEC 15408-1:20XX, Clause 6 provides more detailed information on the
743 relationship between security objectives and security requirements.
- 744 b) Developers, who respond to actual or perceived consumer security requirements in
745 constructing a TOE, **can** find a standardized method to understand those
746 requirements in this document. They **can** also use the contents of this document as a
747 basis for further defining the TOE security functionality and mechanisms that comply
748 with those requirements.
- 749 c) Evaluators, who use the functional requirements defined in this document in
750 verifying that the TOE functional requirements expressed in the PP or ST satisfy the
751 IT security objectives and that all dependencies are accounted for and shown to be
752 satisfied. Evaluators **shall** use this document to assist in determining whether a given
753 TOE satisfies stated requirements.

754

755 IT Security techniques — Evaluation criteria for IT security

756 — Part 2: Security functional components

757 1 Scope

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

761 2 Normative references

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

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

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

770 Editors' note

771 15408-3 has been added as a normative document since some of the component dependencies include
772 assurance components found in part 3.

773 3 Terms and Definitions

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

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

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

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

780 4 Overview

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

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

788 Since the understanding and needs of consumers **can** change, the functional components in this
789 document will need to be maintained. It is envisioned that some PP/ST authors **could** have
790 security needs not (yet) covered by the functional requirement components in this document.
791 In those cases, the PP/ST author **can** choose to consider using functional components and
792 requirements that are not given in this document. The concepts of extensibility are explained in
793 Annex D of ISO/IEC 15408-1:20XX.

794 **4.1 Organization of this document**

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

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

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

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

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

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

- 807 a) ISO/IEC 15408-1:20XX, Clause 3 defines the terms and definitions used in ISO/IEC
808 15408.
- 809 b) ISO/IEC 15408-1:20XX, Annex A provides further guidance on the structure for
810 security functional packages
- 811 c) ISO/IEC 15408-1:20XX, Annex B provides further guidance on the structure for
812 PPs.
- 813 d) ISO/IEC 15408-1:20XX, Annex C provides further guidance on the structure of PP-
814 Modules and PP-Configurations
- 815 e) ISO/IEC 15408-1:20XX, Annex D provides further guidance on the structure for STs

816 **5 Functional requirements paradigm**

817 This clause describes the paradigm used in the security functional components and the
818 derivation of security functional requirements. The key concepts discussed are highlighted in
819 bold/italics. This subclause is not intended to replace or supersede any of the terms found in
820 ISO/IEC 15408-1:20XX, Clause 3.

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

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

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

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

836 The TOE **may** be a monolithic product containing hardware, firmware, and software.
837 Alternatively, a TOE **may** be a distributed product that consists internally of multiple separated
838 parts. Each of these parts of the TOE provides a particular service for the TOE and is connected

839 to the other parts of the TOE through an **internal communication channel**. This channel **can**
840 be as small as a processor bus or **may** encompass a network internal to the TOE.

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

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

- 848 a) The SFRs of the other “trusted IT product” and the SFRs of the TOE have been
849 administratively coordinated and the other trusted IT product is assumed to
850 enforce its SFRs correctly (e. g. by being separately evaluated). Exchanges of
851 information in this situation are called **inter-TSF transfers**, as they are between
852 the TSFs of distinct trusted products.
- 853 b) The other IT product **may** not be trusted, it **may** be called an “untrusted IT
854 product”. Therefore, its SFRs are either unknown or their implementation is not
855 viewed as trustworthy. TSF mediated exchanges of information in this situation are
856 called **transfers outside of the TOE**, as there is no TSF (or its policy characteristics
857 are unknown) on the other IT product.

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

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

EXAMPLE 1

An example of a TOE device is a workstation.

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

EXAMPLE 2

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

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

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

EXAMPLE 3

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

880 TOEs contain **resources** that **may** be used for the processing and storing of information. The
 881 primary goal of the TSF is the complete and correct enforcement of the SFRs over the resources
 882 and information that the TOE controls.

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

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

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

EXAMPLE 4 UNIX processes

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

EXAMPLE 5 functions as might be found in client/server architectures

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

EXAMPLE 6 processes not acting on behalf of a user

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

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

EXAMPLE 7 An example of a subject is an inter-process communication
--

901

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

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

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

EXAMPLE 8

User data:

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

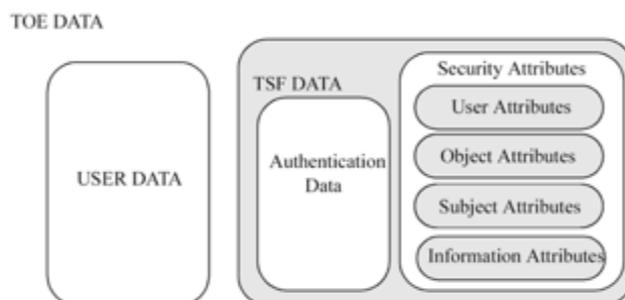
TSF data:

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

918

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

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



929

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

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

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

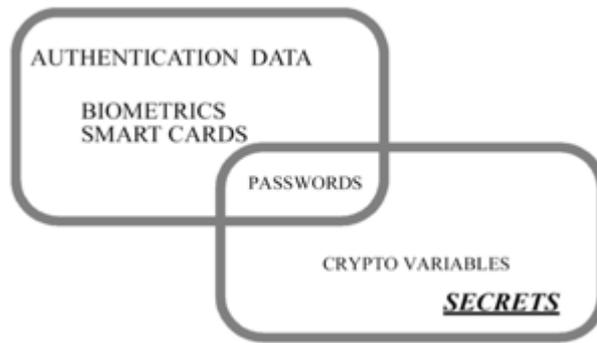
EXAMPLE 9

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

939

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

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



947
948 **Figure 2 — Relationship between “authentication data” and “secrets”**

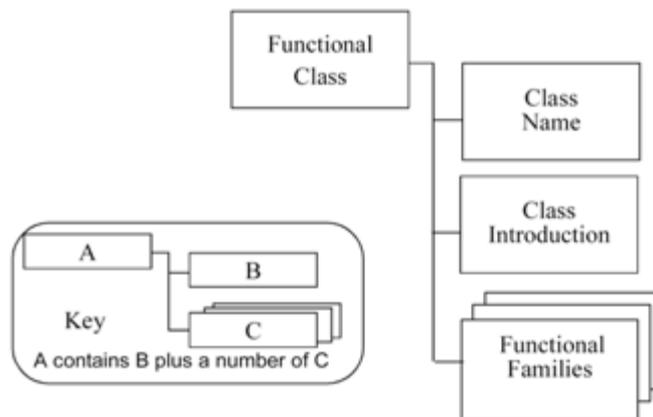
949 **6 Security functional components**

950 **6.1 Overview**

951 This clause defines the content and presentation of the functional requirements of this
952 document and provides guidance on the organization of the requirements for new, extended
953 components that *may* be included in an ST, PP, PP-Module, or security functional package. The
954 functional components and requirements are expressed in classes, families, and components.

955 **6.1.1 Class structure**

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



958
959 **Figure 3 — Functional class structure**

960 **Class name**

961 The class name subclause provides information necessary to identify and categorize a
962 functional class. Every functional class has a unique name. The categorical information consists
963 of a short name of three characters. The short name of the class is used in the specification of
964 the short names of the families of that class.

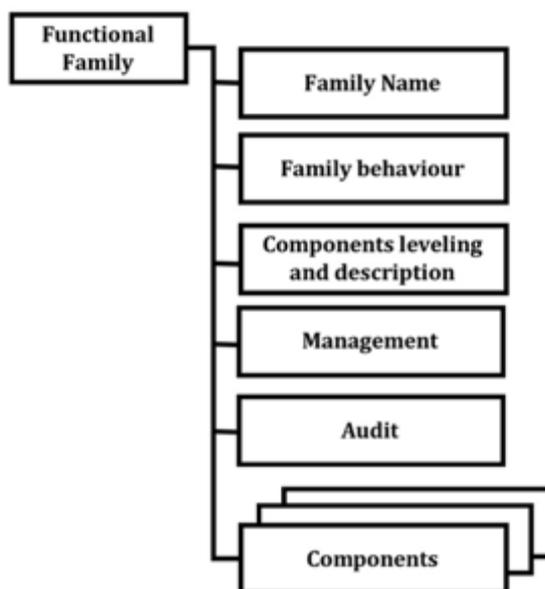
965 **Class introduction**

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

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

971 **6.1.2 Family structure**

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



973

974

Figure 4 — Functional family structure

975 **Family name**

976 The family name subclause provides categorical and descriptive information necessary to
 977 identify and categorize a functional family. Every functional family has a unique name. The
 978 categorical information consists of a short name of seven characters, with the first three
 979 identical to the short name of the class followed by an underscore and the short name of the
 980 family as follows, XXX_YYY. The unique short form of the family name provides the principal
 981 reference name for the security components.

982 **Family behaviour**

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

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

993 **Components leveling and description**

994 Functional families contain one or more components, any one of which **may** be selected for
 995 inclusion in STs, PPs, PP-Modules or security functional packages. The goal of this subclause is
 996 to provide information to users in selecting an appropriate functional component once the
 997 family has been identified as being a necessary or useful part of their security requirements.

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

1000 The relationships between components within a functional family **may** be hierarchical. A
1001 component is hierarchical to another if it offers more security.

1002 As explained in 6.2 the descriptions of the families provide a graphical overview of the
1003 hierarchy of the components in a family.

1004 **Management**

1005 The management clauses contain information for ST, PP, PP-Module, or security functional
1006 package authors to consider as management activities for a given component. The clauses
1007 reference components of the management class (FMT) and provide guidance regarding
1008 potential management activities that **may** be applied via operations to those components.

1009 An author **may** select the indicated management components or **may** include other
1010 management requirements not listed to detail management activities. As such the information
1011 **should** be considered informative.

1012 **Audit**

1013 The *audit* requirements contain auditable events for the authors to select, if requirements from
1014 the class FAU, are included in the ST, PP, PP-Module, or security functional package. These
1015 requirements include security relevant events in terms of the various levels of detail supported
1016 by the components of the Security audit data generation (FAU_GEN) family.

EXAMPLE 1

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

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

1017

1018 It **can** be observed that the categorization of auditable events is hierarchical.

EXAMPLE 2

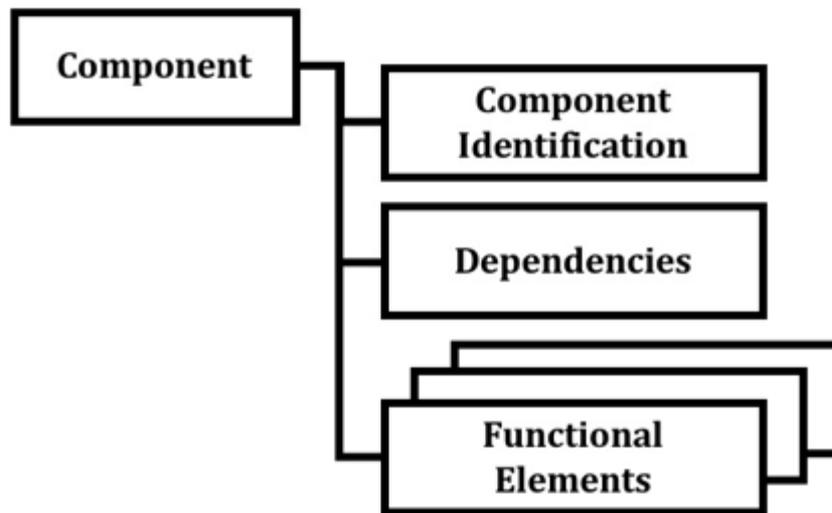
For example, when Basic Audit Generation is desired, all auditable events identified as being both Minimal and Basic are included in the PP/ST through the use of the appropriate assignment operation, except when the higher-level event simply provides more detail than the lower level event. When Detailed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

1019

1020 In the class FAU the rules governing the audit are explained in more detail.

1021 **6.1.3 Component structure**

1022 Figure 5 illustrates the functional component structure.



1023

1024

Figure 5 — Functional component structure

1025 **Component identification**

1026 The component identification subclause provides descriptive information necessary to identify,
1027 categorize, register, and cross-reference a component. The following is provided as part of
1028 every functional component:

1029 *A unique name.* The name reflects the purpose of the component.

1030 *A unique short name.* A unique short form of the functional component name. This short name
1031 serves as the principal reference name for the categorization, registration, and cross-
1032 referencing of the component. This short name reflects the class and family to which the
1033 component belongs and the component number within the family.

1034 *A hierarchical-to list.* A list of other components that this component is hierarchical to and for
1035 which this component **can** be used to satisfy dependencies to the listed components.

1036 **Functional elements**

1037 A set of elements is provided for each component. Each element is individually defined and is
1038 self-contained.

1039 **A functional element is a part of a security functional component that if further divided would not**
1040 **yield a meaningful evaluation result.** It is the smallest part of the taxonomy that is identified in
1041 ISO/IEC 15408-2.

1042 **Editors' Note**

1043 **Based on the definition of "3.68 element" in Part 1, the editors suggest to restate the above sentence**
1044 **highlighted with yellow as "A functional element is a part of a security functional component which is self-**
1045 **sufficient in detail such that further division would not improve the understandability and operability of**
1046 **the component and thus unnecessary."**

1047 **Unless comments are received on this topic, the editor's suggestion will be accepted and presented in the**
1048 **next draft.**

1049 When building packages, PPs and/or STs, it is not permitted to select only one or more
1050 elements from a component. The complete set of elements of a component must be selected for
1051 inclusion in a PP, PP-Module, security functional package or an ST.

1052 A unique short form of the functional element name is provided.

EXAMPLE

The component name FDP_IFF.4.2 reads as follows:

- F - functional requirement,
- DP - class “User data protection”,
- _IFF - family “Information flow control functions”,
- .4 - 4th component named “Partial elimination of illicit information flows”,
- .2 - 2nd element of the component.

1053

1054 **Dependencies**

1055 Dependencies among functional components arise when a component is not self-sufficient and
 1056 relies upon the functionality of, or interaction with, another component for its own proper
 1057 functioning.

1058 Each functional component provides a complete list of dependencies to other functional and
 1059 assurance components. Some components **may** list “No dependencies”. The components
 1060 depended upon **may** in turn have dependencies on other components. The list provided in the
 1061 components will be the direct dependencies. That is only references to the other functional
 1062 components that are required for this component to perform its job properly. The indirect
 1063 dependencies, that is the dependencies that result from the depended upon components **can** be
 1064 found in Annex A of this document. It is noted that in some cases the dependency is optional in
 1065 that a number of functional components are provided, where each one of them would be
 1066 sufficient to satisfy the dependency.

EXAMPLE

FDP_UIT.1 Data exchange integrity

1067 The dependency list identifies the minimum functional or assurance components needed to
 1068 satisfy the security requirements associated with an identified component. Components that
 1069 are hierarchical to the identified component **may** also be used to satisfy the dependency.

1070 The dependencies indicated in this document are normative and they **shall** be satisfied within a
 1071 package, PP or ST. In situations where the indicated dependencies are not applicable, the author
 1072 **shall** satisfy the dependency by providing a rationale why it is not applicable and **may** leave the
 1073 depended upon component from the package, PP or ST.

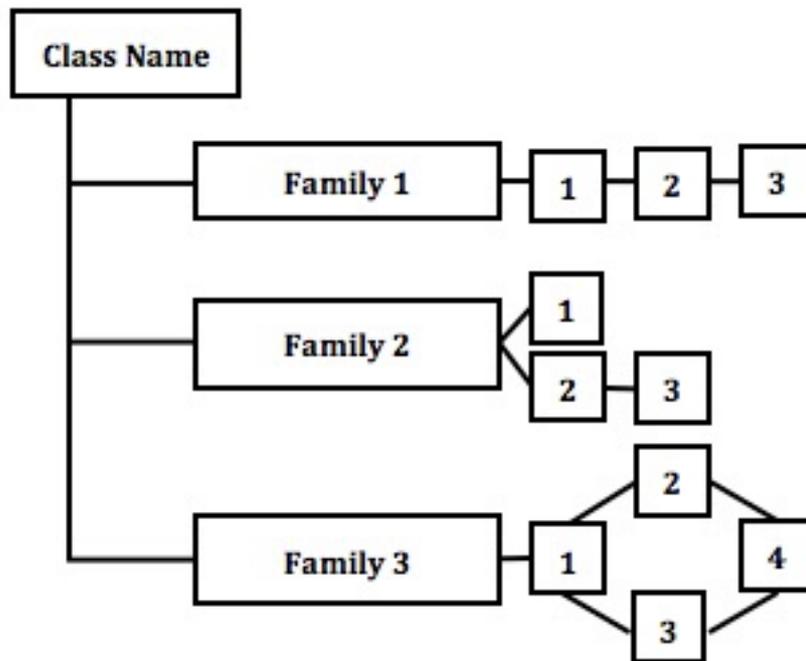
1074 **6.2 Component catalogue**

1075 The grouping of the components in this document does not reflect any formal taxonomy.

1076 This document contains classes of families and components, which are rough groupings on the
 1077 basis of related function or purpose, presented in alphabetic order. At the start of each class is
 1078 an informative diagram that indicates the taxonomy of each class, indicating the families in each
 1079 class and the components in each family. Figure 6 is a useful indicator of the hierarchical
 1080 relationship that **may** exist between components.

1081 In the description of the functional components, a subclause identifies the dependencies
 1082 between the component and any other components.

1083 In each class, a figure describing the family hierarchy similar to Figure 6 is provided. In Figure 6
 1084 the first family, Family 1, contains three hierarchical components, where component 2 and
 1085 component 3 **can** both be used to satisfy dependencies on component 1. Component 3 is
 1086 hierarchical to component 2 and **can** also be used to satisfy dependencies on component 2.



1087

1088

Figure 6 — Sample class decomposition diagram

1089 In Family 2 there are three components not all of which are hierarchical. Components 1 and 2
 1090 are hierarchical to no other components. Component 3 is hierarchical to component 2 and **can**
 1091 be used to satisfy dependencies on component 2, but not to satisfy dependencies on component
 1092 1.

1093 In Family 3, components 2, 3, and 4 are hierarchical to component 1. Components 2 and 3 are
 1094 both hierarchical to component 1, but non-comparable. Component 4 is hierarchical to both
 1095 component 2 and component 3.

1096 These diagrams are meant to complement the text of the families and make identification of the
 1097 relationships easier. They do not replace the “Hierarchical to:” note in each component that is
 1098 the mandatory claim of hierarchy for each component.

1099 **6.2.1 Component changes highlighting**

1100 The relationship between components within a family is highlighted using a **bolding**
 1101 convention. This bolding convention calls for the bolding of all new requirements. For
 1102 hierarchical components, requirements are bolded when they are enhanced or modified beyond
 1103 the requirements of the previous component. In addition, any new or enhanced permitted
 1104 operations beyond the previous component are also highlighted using **bold** type.

1105

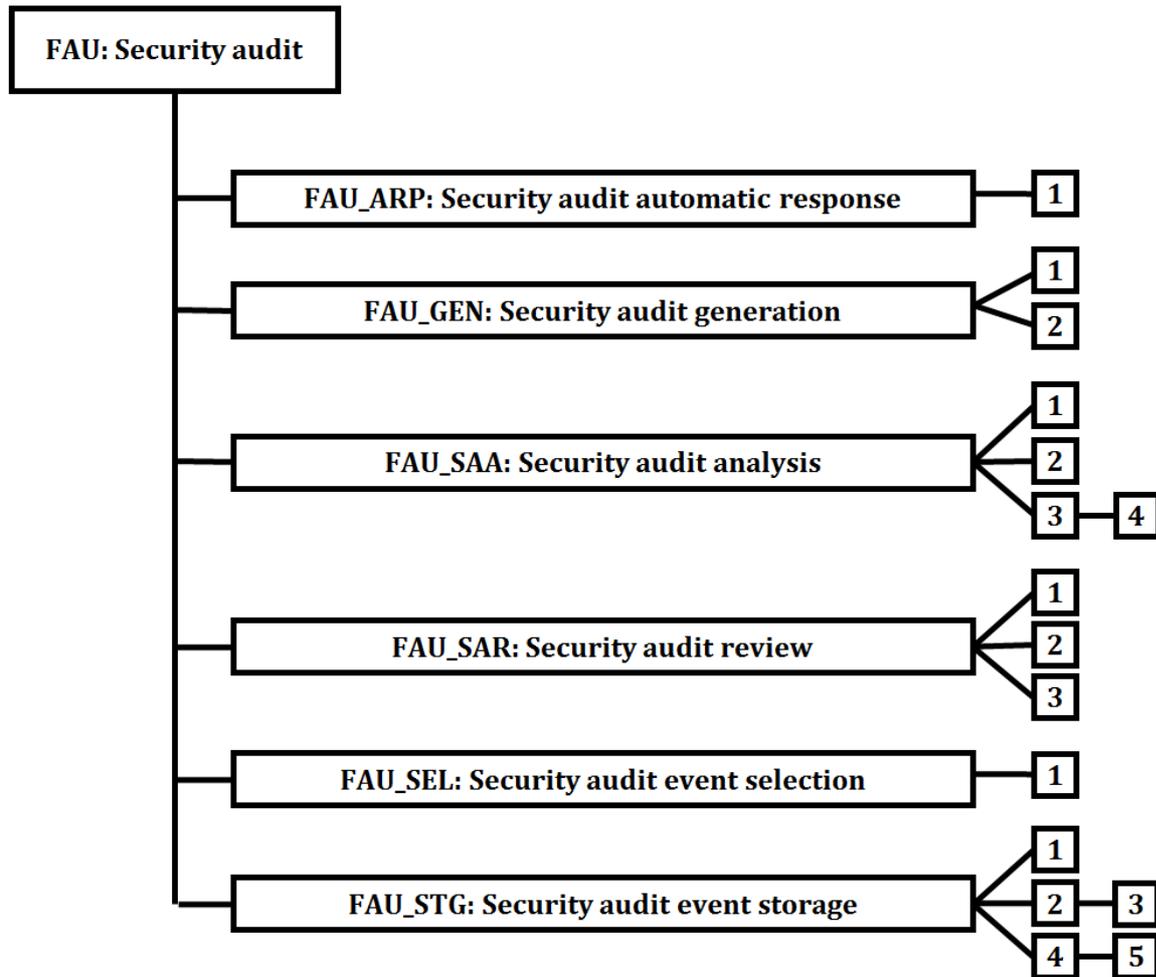
1106 **7 Class FAU: Security audit**

1107 **7.1 Class description**

1108 Security auditing involves recognizing, recording, storing, and analyzing information related to
 1109 security relevant activities (i.e. activities controlled by the TSF). The resulting audit records **can**
 1110 be examined to determine which security relevant activities took place and whom (which user)
 1111 is responsible for them.

1112 Figure 7 shows the decomposition of this class, it’s families and components. Elements are not
 1113 shown in the figure.

1114 Annex C provides explanatory information for this class and **should** be consulted when using
 1115 the components identified in this class.



1116
 1117 **Figure 7 — FAU: Security audit class decomposition**

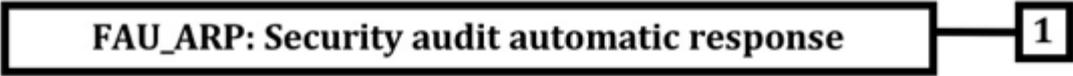
1118 **7.2 Security audit automatic response (FAU_ARP)**

1119 **7.2.1 Family behaviour**

1120 This family defines the response to be taken in case of detected events indicative of a potential
 1121 security violation.

1122 **7.2.2 Components leveling and description**

1123 Figure 8 shows the component leveling for this family.



FAU_ARP: Security audit automatic response

1

1124

1125

Figure 8 — FAU_ARP: Component leveling

1126 At FAU_ARP.1 Security alarms, the TSF **shall** take actions in case a potential security violation is
1127 detected.

7.2.3 Management of FAU_ARP.1

1129 The following actions **could** be considered for the management functions in FMT:

1130 a) the management (addition, removal, or modification) of actions.

7.2.4 Audit of FAU_ARP.1

1132 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1133 in the PP/ST:

1134 a) Minimal: Actions taken due to potential security violations.

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

1137 Hierarchical to: No other components.

1138 Dependencies: FAU_SAA.1 Potential violation analysis

FAU_ARP.1.1

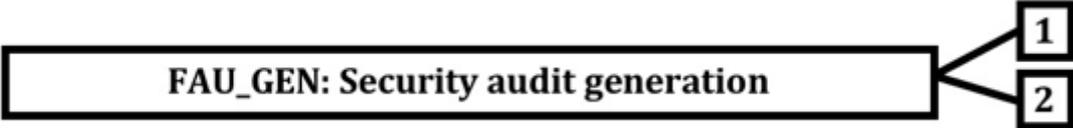
1140 **The TSF shall take [assignment: list of actions] upon detection of a potential security**
1141 **violation.**

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

1144 This family defines requirements for recording the occurrence of security relevant events that
1145 take place under TSF control. This family identifies the level of auditing, enumerates the types
1146 of events that **shall** be auditable by the TSF, and identifies the minimum set of audit-related
1147 information that **should** be provided within various audit record types.

7.3.2 Components leveling and description

1149 Figure 9 shows the component leveling for this family.



FAU_GEN: Security audit generation

1

2

1150

1151

Figure 9 — FAU_GEN: Component leveling

1152 FAU_GEN.1 Audit data generation defines the level of auditable events and specifies the list of
1153 data that **shall** be recorded in each record.

1154 At FAU_GEN.2 User identity association, the TSF **shall** associate auditable events to individual
1155 user identities.

1156 **7.3.3 Management of FAU_GEN.1, FAU_GEN.2**

1157 The following actions **could** be considered for the management functions in FMT:

1158 a) There are no management activities foreseen.

1159 **7.3.4 Audit of FAU_GEN.1, FAU_GEN.2**

1160 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1161 in the PP/ST:

1162 a) There are no auditable events foreseen.

1163 **7.3.5 FAU_GEN.1 Audit data generation**

1164 **Component relationships**

1165 Hierarchical to: No other components.

1166 Dependencies: FPT_STM.1 Reliable time stamps

1167 **FAU_GEN.1.1**

1168 The TSF **shall** be able to generate audit data of the following auditable events:

1169 a) **Start-up and shutdown of the audit functions;**

1170 b) **All auditable events for the [selection, choose one of: *minimum, basic,***
1171 ***detailed, not specified*] level of audit; and**

1172 c) **[assignment: other specifically defined auditable events].**

1173 **FAU_GEN.1.2**

1174 The TSF **shall** record within the audit data at least the following information:

1175 a) **Date and time of the auditable event, type of event, subject identity (if**
1176 **applicable), and the outcome (success or failure) of the event; and**

1177 b) **For each auditable event type, based on the auditable event definitions of the**
1178 **functional components included in the PP/ST, [assignment: *other audit***
1179 ***relevant information*].**

1180 **7.3.6 FAU_GEN.2 User identity association**

1181 **Component relationships**

1182 Hierarchical to: No other components.

1183 Dependencies: FAU_GEN.1 Audit data generation

1184 FIA_UID.1 Timing of identification

1185 **FAU_GEN.2.1**

1186 **For audit events resulting from actions of identified users, the TSF **shall** be able to**
1187 **associate each auditable event with the identity of the user that caused the event.**

1188 **7.4 Security audit analysis (FAU_SAA)**

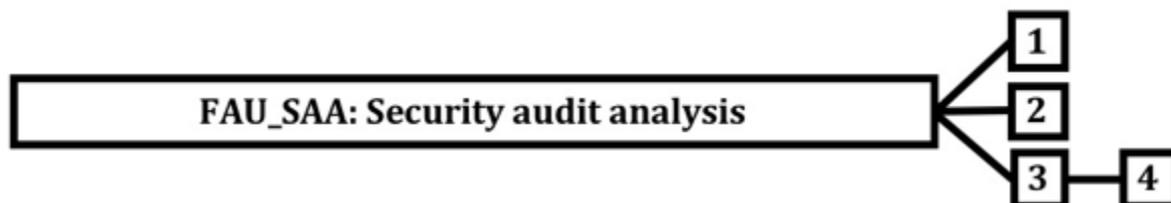
1189 **7.4.1 Family behaviour**

1190 This family defines requirements for automated means that analyze system activity and audit
1191 data looking for possible or real security violations. This analysis **may** work in support of
1192 intrusion detection, or automatic response to a potential security violation.

1193 The actions to be taken based on the detection **can** be specified using the Security audit
1194 automatic response (FAU_ARP) family as desired.

1195 7.4.2 Components leveling and description

1196 Figure 10 shows the component leveling for this family.



1197

1198 **Figure 10 — FAU_SAA: Component leveling**

1199 In FAU_SAA.1 Potential violation analysis, basic threshold detection on the basis of a fixed rule
1200 set is required.

1201 In FAU_SAA.2 Profile based anomaly detection, the TSF maintains individual profiles of system
1202 usage, where a profile represents the historical patterns of usage performed by members of the
1203 profile target group. A profile target group refers to a group of one or more individuals who
1204 interact with the TSF. Each member of a profile target group is assigned an individual suspicion
1205 rating that represents how well that member's current activity corresponds to the established
1206 patterns of usage represented in the profile. This analysis **can** be performed at runtime or
1207 during a post-collection batch-mode analysis.

1208 In FAU_SAA.3 Simple attack heuristics, the TSF **shall** be able to detect the occurrence of
1209 signature events that represent a significant threat to enforcement of the SFRs. This search for
1210 signature events **may** occur in real-time or during a post-collection batch-mode analysis.

1211 In FAU_SAA.4 Complex attack heuristics, the TSF **shall** be able to represent and detect multi-
1212 step intrusion scenarios. The TSF is able to compare system events (possibly performed by
1213 multiple individuals) against event sequences known to represent entire intrusion scenarios.
1214 The TSF **shall** be able to indicate when a signature event or event sequence is found that
1215 indicates a potential violation of the enforcement of the SFRs.

1216 7.4.3 Management of FAU_SAA.1

1217 The following actions **could** be considered for the management functions in FMT:

- 1218 a) Maintenance of the rules by (adding, modifying, deletion) of rules from the set of
1219 rules.

1220 7.4.4 Management of FAU_SAA.2

1221 The following actions **could** be considered for the management functions in FMT:

- 1222 a) Maintenance (deletion, modification, addition) of the group of users in the profile
1223 target group.

1224 7.4.5 Management of FAU_SAA.3

1225 The following actions **could** be considered for the management functions in FMT:

- 1226 a) Maintenance (deletion, modification, addition) of the subset of system events.

1227 7.4.6 Management of FAU_SAA.4

1228 The following actions **could** be considered for the management functions in FMT:

- 1229 a) Maintenance (deletion, modification, addition) of the subset of system events;

1230 b) Maintenance (deletion, modification, addition) of the set of sequences of system
1231 events.

1232 **7.4.7 Audit of FAU_SAA.1, FAU_SAA.2, FAU_SAA.3, FAU_SAA.4**

1233 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1234 in the PP/ST:

1235 a) Minimal: Enabling and disabling of any of the analysis mechanisms;

1236 b) Minimal: Automated responses performed by the tool.

1237 **7.4.8 FAU_SAA.1 Potential violation analysis**

1238 **Component relationships**

1239 Hierarchical to: No other components.

1240 Dependencies: FAU_GEN.1 Audit data generation

1241 **FAU_SAA.1.1**

1242 **The TSF shall be able to apply a set of rules in monitoring the audited events and based**
1243 **upon these rules indicate a potential violation of the enforcement of the SFRs.**

1244 **FAU_SAA.1.2**

1245 **The TSF shall enforce the following rules for monitoring audited events:**

1246 a) **Accumulation or combination of [assignment: *subset of defined auditable***
1247 ***events*] known to indicate a potential security violation;**

1248 b) **[assignment: *any other rules*].**

1249 **7.4.9 FAU_SAA.2 Profile based anomaly detection**

1250 **Component relationships**

1251 Hierarchical to: No other components.

1252 Dependencies: FIA_UID.1 Timing of identification

1253 **FAU_SAA.2.1**

1254 **The TSF shall be able to maintain profiles of system usage, where an individual profile**
1255 **represents the historical patterns of usage performed by the member(s) of [assignment:**
1256 ***the profile target group*].**

1257 **FAU_SAA.2.2**

1258 **The TSF shall be able to maintain a suspicion rating associated with each user whose**
1259 **activity is recorded in a profile, where the suspicion rating represents the degree to**
1260 **which the user's current activity is found inconsistent with the established patterns of**
1261 **usage represented in the profile.**

1262 **FAU_SAA.2.3**

1263 **The TSF shall be able to indicate a possible violation of the enforcement of the SFRs when**
1264 **a user's suspicion rating exceeds the following threshold conditions [assignment:**
1265 ***conditions under which anomalous activity is reported by the TSF*].**

1266 **7.4.10 FAU_SAA.3 Simple attack heuristics**1267 **Component relationships**

1268 Hierarchical to: No other components.

1269 Dependencies: No dependencies.

1270 **FAU_SAA.3.1**

1271 The TSF **shall** be able to maintain an internal representation of the following signature
 1272 events [assignment: *a subset of system events*] that **may** indicate a violation of the
 1273 enforcement of the SFRs.

1274 **FAU_SAA.3.2**

1275 The TSF **shall** be able to compare the signature events against the record of system
 1276 activity discernible from an examination of [assignment: *the information to be used to*
 1277 *determine system activity*].

1278 **FAU_SAA.3.3**

1279 The TSF **shall** be able to indicate a potential violation of the enforcement of the SFRs
 1280 when a system event is found to match a signature event that indicates a potential
 1281 violation of the enforcement of the SFRs.

1282 **7.4.11 FAU_SAA.4 Complex attack heuristics**1283 **Component relationships**

1284 Hierarchical to: FAU_SAA.3 Simple attack heuristics

1285 Dependencies: No dependencies.

1286 **FAU_SAA.4.1**

1287 The TSF **shall** be able to maintain an internal representation of the following **event sequences**
 1288 **of known intrusion scenarios** [assignment: *list of sequences of system events whose*
 1289 *occurrence are representative of known penetration scenarios*] and the following signature
 1290 events [assignment: *a subset of system events*] that **may** indicate a **potential** violation of the
 1291 enforcement of the SFRs.

1292 **FAU_SAA.4.2**

1293 The TSF **shall** be able to compare the signature events **and event sequences** against the record
 1294 of system activity discernible from an examination of [assignment: *the information to be used to*
 1295 *determine system activity*].

1296 **FAU_SAA.4.3**

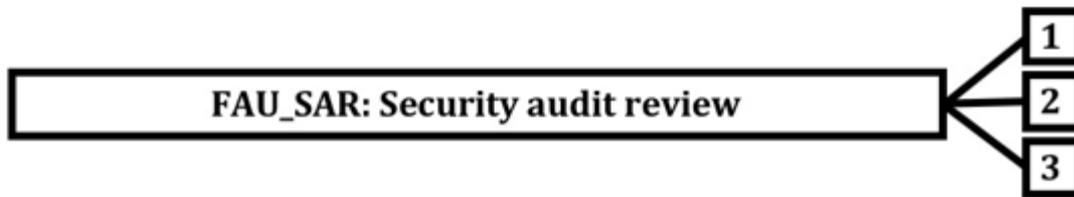
1297 The TSF **shall** be able to indicate a potential violation of the enforcement of the SFRs when
 1298 system **activity** is found to match a signature event **or event sequence** that indicates a
 1299 potential violation of the enforcement of the SFRs.

1300 **7.5 Security audit review (FAU_SAR)**1301 **7.5.1 Family behaviour**

1302 This family defines the requirements for tools that are made available to authorized users to
 1303 assist in the review of audit data.

1304 **7.5.2 Components leveling and description**

1305 Figure 11 shows the component leveling for this family.



1306

1307 **Figure 11 — FAU_SAR: Component leveling**

1308

1309 FAU_SAR.1 Audit review, provides the capability to read information from the audit data.

1310 FAU_SAR.2 Restricted audit review, requires that there are no other users except those that
1311 have been identified in FAU_SAR.1 Audit review that **can** read the information.

1312 FAU_SAR.3 Selectable audit review, requires audit review tools to select the audit data to be
1313 reviewed based on criteria.

1314 **7.5.3 Management of FAU_SAR.1**

1315 The following actions **could** be considered for the management functions in FMT:

- 1316 a) Maintenance (deletion, modification, addition) of the group of users with read
1317 access right to the audit records.

1318 **7.5.4 Management of FAU_SAR.2, FAU_SAR.3**

1319 The following actions **could** be considered for the management functions in FMT:

- 1320 a) There are no management activities foreseen.

1321 **7.5.5 Audit of FAU_SAR.1**

1322 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1323 in the PP/ST:

- 1324 a) Basic: Reading of information from the audit records.

1325 **7.5.6 Audit of FAU_SAR.2**

1326 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1327 in the PP/ST:

- 1328 a) Basic: Unsuccessful attempts to read information from the audit records.

1329 **7.5.7 Audit of FAU_SAR.3**

1330 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1331 in the PP/ST:

- 1332 a) Detailed: the parameters used for the viewing.

1333 **7.5.8 FAU_SAR.1 Audit review**

1334 **Component relationships**

1335 Hierarchical to: No other components.

1336 Dependencies: FAU_GEN.1 Audit data generation

1337 **FAU_SAR.1.1**

1338 The TSF **shall** provide [assignment: *authorized users*] with the capability to read
1339 [assignment: *list of audit information*] from the audit data.

1340 **FAU_SAR.1.2**

1341 The TSF **shall** provide the audit data in a manner suitable for the user to interpret the
1342 information.

1343 **7.5.9 FAU_SAR.2 Restricted audit review**1344 **Component relationships**

1345 Hierarchical to: No other components.

1346 Dependencies: FAU_SAR.1 Audit review

1347 **FAU_SAR.2.1**

1348 The TSF **shall** prohibit all users read access to the audit data, except those users that
1349 have been granted explicit read-access.

1350 **7.5.10 FAU_SAR.3 Selectable audit review**

1351 Hierarchical to: No other components.

1352 Dependencies: FAU_SAR.1 Audit review

1353 **FAU_SAR.3.1**

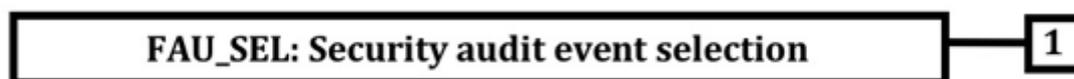
1354 The TSF **shall** provide the ability to apply [assignment: *methods of selection and/or*
1355 *ordering*] of audit data based on [assignment: *criteria with logical relations*].

1356 **7.6 Security audit event selection (FAU_SEL)**1357 **7.6.1 Family behaviour**

1358 This family defines requirements to select the set of events to be audited during TOE operation
1359 from the set of all auditable events.

1360 **7.6.2 Components leveling and description**

1361 Figure 12 shows the component leveling for this family.



1362

1363 **Figure 12 — FAU_SEL: Component leveling**

1364 FAU_SEL.1 Selective audit, requires the ability to select the set of events to be audited from the
1365 set of all auditable events, identified in FAU_GEN.1 Audit data generation, based upon attributes
1366 to be specified by the PP/ST author.

1367 **7.6.3 Management of FAU_SEL.1**

1368 The following actions **could** be considered for the management functions in FMT:

1369 a) Maintenance of the rights to view/modify the audit data.

1370 **7.6.4 Audit of FAU_SEL.1**

1371 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1372 in the PP/ST:

- 1373 a) Minimal: All modifications to the audit configuration that occur while the audit
1374 collection functions are operating.

1375 **7.6.5 FAU_SEL.1 Selective audit**

1376 **Component relationships**

1377 Hierarchical to: No other components.

1378 Dependencies: FAU_GEN.1 Audit data generation

1379 FMT_MTD.1 Management of TSF data

1380 **FAU_SEL.1.1**

1381 The TSF **shall** be able to select the set of events to be audited from the set of all auditable
1382 events based on the following attributes:

1383 a) [selection: *object identity, user identity, subject identity, host identity, event*
1384 *type*]

1385 b) [assignment: *list of additional attributes that audit selectivity is based upon*]

1386

1387 **7.7 Security audit data storage (FAU_STG)**1388 **7.7.1 Family behaviour**

1389 This family defines the requirements for the TSF to be able to create and maintain a secure
 1390 audit trail. Stored audit data refers to those data stored within an audit trail, and not to any
 1391 audit data that has been retrieved (to temporary storage) through selection.

1392 **7.7.2 Components leveling and description**

1393 Figure 13 shows the component leveling for this family.



1394

1395 **Figure 13 — FAU_STG: Component leveling**

1396 FAU_STG.1 Audit data storage location, requires that the storage location(s) for audit data be
 1397 specified

1398 FAU_STG.2 Protected audit data storage, requires that protections are placed on the audit data.
 1399 It will be protected from unauthorized deletion and/or modification.

1400 FAU_STG.3 Guarantees of audit data availability, specifies the guarantees that the TSF maintains
 1401 over the audit data given the occurrence of an undesired condition.

1402 FAU_STG.4 Action in case of possible audit data loss specifies actions to be taken if a threshold
 1403 on the stored audit data is exceeded.

1404 FAU_STG.5 Prevention of audit data loss specifies actions to be taken in the case that audit data
 1405 storage is full.

1406 **7.7.3 Management of FAU_STG.1**

1407 The following actions **could** be considered for the management functions in FMT:

1408 a) Maintenance of remote audit storage locations

1409 **7.7.4 Management of FAU_STG.2**

1410 The following actions **could** be considered for the management functions in FMT:

1411 a) There are no management activities foreseen.

1412 **7.7.5 Management of FAU_STG.3**

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

1414 a) Maintenance of the parameters that control the audit data storage capability.

1415 **7.7.6 Management of FAU_STG.4**

1416 The following actions **could** be considered for the management functions in FMT:

1417 a) Maintenance (deletion, modification, addition) of actions to be taken in case of
 1418 imminent audit data storage failure.

1419 **7.7.7 Management of FAU_STG.5**

1420 The following actions **could** be considered for the management functions in FMT:

- 1421 a) Maintenance (deletion, modification, addition) of actions to be taken in case of
1422 audit data storage failure.

1423 **7.7.8 Audit of FAU_STG.1**

1424 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1425 in the PP/ST:

- 1426 a) Basic: Changes in the location of remote audit data storage.

1427 **7.7.9 Audit of FAU_STG.2, FAU_STG.3**

1428 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1429 in the PP/ST:

- 1430 a) There are no auditable events foreseen.

1431 **7.7.10 Audit of FAU_STG.4**

1432 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1433 in the PP/ST:

- 1434 a) Basic: Actions taken due to exceeding of a threshold.

1435 **7.7.11 Audit of FAU_STG.5**

1436 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1437 in the PP/ST:

- 1438 a) Basic: Actions taken due to the audit data storage failure.

1439 **7.7.12 FAU_STG.1 Audit data storage location**

1440 **Component relationships**

1441	Hierarchical to:	No other components
1442	Dependencies:	FAU_GEN.1 Audit data generation
1443		FTP_ITC.1 Inter-TSF trusted channel

1444 **FAU_STG.1.1**

1445 **The TSF **shall** be able to store generated audit data on the [selection: *TOE itself, transmit*
1446 *the generated audit data to an external IT entity using a trusted channel according to*
1447 *FTP_ITC, [assignment: other storage location(s)].]***

1448 **7.7.13 FAU_STG.2 Protected audit data storage**

1449 **Component relationships**

1450	Hierarchical to:	No other components
1451	Dependencies:	FAU_GEN.1 Audit data generation

1452 **FAU_STG.2.1**

1453 **The TSF **shall** protect the stored audit data in the audit trail from unauthorized deletion.**

1454 **FAU_STG.2.2**

1455 **The TSF shall be able to [selection, choose one of: *prevent, detect*] unauthorized**
 1456 **modifications to the stored audit data in the audit trail.**

1457 **7.7.14 FAU_STG.3 Guarantees of audit data availability**1458 **Component relationships**

1459 Hierarchical to: FAU_STG.2 Protected audit data storage

1460 Dependencies: FAU_GEN.1 Audit data generation

1461 **FAU_STG.3.1**

1462 **The TSF shall ensure that [assignment: *metric for saving audit data*] stored audit data**
 1463 **will be maintained when the following conditions occur: [selection: *audit data storage***
 1464 ***exhaustion, failure, attack*].**

1465 **7.7.15 FAU_STG.4 Action in case of possible audit data loss**1466 **Component relationships**

1467 Hierarchical to: No other components

1468 Dependencies: FAU_STG.2 Protected audit data storage

1469 **FAU_STG.4.1**

1470 The TSF shall [assignment: *actions to be taken in case of possible audit data storage failure*]
 1471 if the audit data storage exceeds [assignment: *pre-defined limit*].

1472 **7.7.16 FAU_STG.5 Prevention of audit data loss**1473 **Component relationships**

1474 Hierarchical to: FAU_STG.4 Action in case of possible audit data loss

1475 Dependencies: FAU_STG.2 Protected audit data storage

1476 FAU_GEN.1 Audit data generation

1477 **FAU_STG.5.1**

1478 **The TSF shall [selection: *ignore audited events, "prevent audited events, except those***
 1479 ***taken by the authorized user with special rights", overwrite the oldest stored audit***
 1480 ***records*], [assignment: *other actions to be taken in case of audit storage failure and***
 1481 ***conditions for the actions*] if the audit data storage is full.**

1482

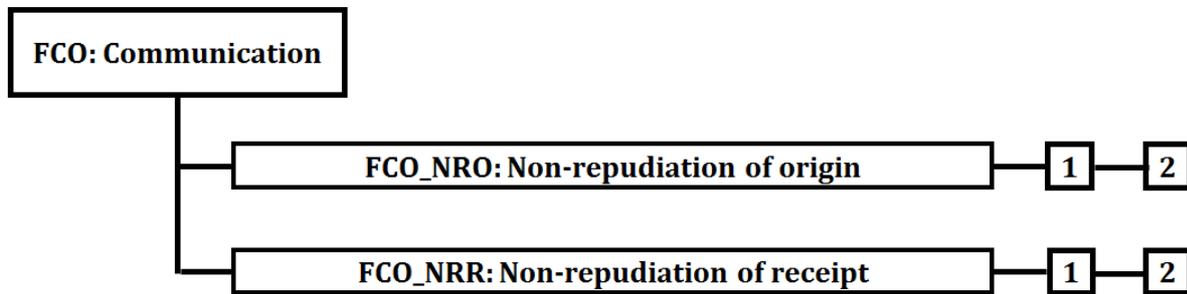
1483 **8 Class FCO: Communication**

1484 **8.1 Class description**

1485 This class provides two families specifically concerned with assuring the identity of a party
 1486 participating in a data exchange. These families are related to assuring the identity of the
 1487 originator of transmitted information (proof of origin) and assuring the identity of the recipient
 1488 of transmitted information (proof of receipt). These families ensure that an originator cannot
 1489 deny having sent the message, nor can the recipient deny having received it. Figure 14 shows
 1490 the decomposition of the class.

1491 Figure 14 shows the decomposition of this class, it's families and components. Elements are not
 1492 shown in the figure.

1493 Annex D provides explanatory information for this class and **should** be consulted when using
 1494 the components identified in this class.



1495
 1496 **Figure 14 — FCO: Communication class decomposition**

1497 **8.2 Non-repudiation of origin (FCO_NRO)**

1498 **8.2.1 Family behaviour**

1499 Non-repudiation of origin ensures that the originator of information cannot successfully deny
 1500 having sent the information. This family requires that the TSF provide a method to ensure that a
 1501 subject that receives information during a data exchange is provided with evidence of the origin
 1502 of the information. This evidence **can** then be verified by either this subject or other subjects.

1503 **8.2.2 Components leveling and description**

1504 Figure 15 shows the component leveling for this family.



1505
 1506 **Figure 15 — FCO_NRO: Component leveling**

1507 FCO_NRO.1 Selective proof of origin, requires the TSF to provide subjects with the capability to
 1508 request evidence of the origin of information.

1509 FCO_NRO.2 Enforced proof of origin, requires that the TSF always generate evidence of origin
 1510 for transmitted information.

1511 **8.2.3 Management of FCO_NRO.1, FCO_NRO.2**

1512 The following actions **could** be considered for the management functions in FMT:

- 1513 a) The management of changes to information types, fields, originator attributes and
- 1514 recipients of evidence.

1515 **8.2.4 Audit of FCO_NRO.1**

1516 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1517 in the PP/ST:

- 1518 a) Minimal: The identity of the user who requested that evidence of origin would be
1519 generated.
- 1520 b) Minimal: The invocation of the non-repudiation service.
- 1521 c) Basic: Identification of the information, the destination, and a copy of the evidence
1522 provided.
- 1523 d) Detailed: The identity of the user who requested a verification of the evidence.

1524 **8.2.5 Audit of FCO_NRO.2**

1525 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1526 in the PP/ST:

- 1527 a) Minimal: The invocation of the non-repudiation service.
- 1528 b) Basic: Identification of the information, the destination, and a copy of the evidence
1529 provided.
- 1530 c) Detailed: The identity of the user who requested a verification of the evidence.

1531 **8.2.6 FCO_NRO.1 Selective proof of origin**1532 **Component relationships**

1533 Hierarchical to: No other components.

1534 Dependencies: FIA_UID.1 Timing of identification

1535 **FCO_NRO.1.1**

1536 **The TSF shall be able to generate evidence of origin for transmitted [assignment: list of**
1537 **information types] at the request of the [selection: originator, recipient, [assignment: list**
1538 **of third parties]].**

1539 **FCO_NRO.1.2**

1540 **The TSF shall be able to relate the [assignment: list of attributes] of the originator of the**
1541 **information, and the [assignment: list of information fields] of the information to which**
1542 **the evidence applies.**

1543 **FCO_NRO.1.3**

1544 **The TSF shall provide a capability to verify the evidence of origin of information to**
1545 **[selection: originator, recipient, [assignment: list of third parties]] given [assignment:**
1546 **limitations on the evidence of origin].**

1547 **8.2.7 FCO_NRO.2 Enforced proof of origin**1548 **Component relationships**

1549 Hierarchical to: FCO_NRO.1 Selective proof of origin

1550 Dependencies: FIA_UID.1 Timing of identification

1551 **FCO_NRO.2.1**

1552 **The TSF shall enforce the generation of evidence of origin for transmitted [assignment: list of**
1553 **information types] at all times.**

1554 **FCO_NRO.2.2**

1555 The TSF **shall** be able to relate the [assignment: *list of attributes*] of the originator of the
 1556 information, and the [assignment: *list of information fields*] of the information to which the
 1557 evidence applies.

1558 **FCO_NRO.2.3**

1559 The TSF **shall** provide a capability to verify the evidence of origin of information to [selection:
 1560 *originator, recipient, [assignment: list of third parties]*] given [assignment: *limitations on the*
 1561 *evidence of origin*].

1562 **8.3 Non-repudiation of receipt (FCO_NRR)**

1563 **8.3.1 Family behaviour**

1564 Non-repudiation of receipt ensures that the recipient of information cannot successfully deny
 1565 receiving the information. This family requires that the TSF provide a method to ensure that a
 1566 subject that transmits information during a data exchange is provided with evidence of receipt
 1567 of the information. This evidence **can** then be verified by either this subject or other subjects.

1568 **8.3.2 Components leveling and description**

1569 Figure 16 shows the component leveling for this family.



1570

1571 **Figure 16 — FCO_NRR: Component leveling**

1572 FCO_NRR.1 Selective proof of receipt, requires the TSF to provide subjects with a capability to
 1573 request evidence of the receipt of information.

1574 FCO_NRR.2 Enforced proof of receipt, requires that the TSF always generate evidence of receipt
 1575 for received information.

1576 **8.3.3 Management of FCO_NRR.1, FCO_NRR.2**

1577 The following actions **could** be considered for the management functions in FMT:

- 1578 a) The management of changes to information types, fields, originator attributes and
 1579 third-party recipients of evidence.

1580 **8.3.4 Audit of FCO_NRR.1**

1581 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 1582 in the PP/ST:

- 1583 a) Minimal: The identity of the user who requested that evidence of receipt would be
 1584 generated.
- 1585 b) Minimal: The invocation of the non-repudiation service.
- 1586 c) Basic: Identification of the information, the destination, and a copy of the evidence
 1587 provided.
- 1588 d) Detailed: The identity of the user who requested a verification of the evidence.

1589 **8.3.5 Audit of FCO_NRR.2**

1590 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 1591 in the PP/ST:

- 1592 a) Minimal: The invocation of the non-repudiation service.
- 1593 b) Basic: Identification of the information, the destination, and a copy of the evidence
1594 provided.
- 1595 c) Detailed: The identity of the user who requested a verification of the evidence.

1596 8.3.6 FCO_NRR.1 Selective proof of receipt

1597 Component relationships

- 1598 Hierarchical to: No other components.
- 1599 Dependencies: FIA_UID.1 Timing of identification

1600 FCO_NRR.1.1

1601 The TSF **shall** be able to generate evidence of receipt for received [assignment: *list of*
1602 *information types*] at the request of the [selection: *originator, recipient, [assignment: list*
1603 *of third parties*]].

1604 FCO_NRR.1.2

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

1608 FCO_NRR.1.3

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

1612 8.3.7 FCO_NRR.2 Enforced proof of receipt

1613 Component relationships

- 1614 Hierarchical to: FCO_NRR.1 Selective proof of receipt
- 1615 Dependencies: FIA_UID.1 Timing of identification

1616 FCO_NRR.2.1

1617 The TSF **shall enforce the generation of** evidence of receipt for received [assignment: *list of*
1618 *information types*] at **all times**.

1619 FCO_NRR.2.2

1620 The TSF **shall** be able to relate the [assignment: *list of attributes*] of the recipient of the
1621 information, and the [assignment: *list of information fields*] of the information to which the
1622 evidence applies.

1623 FCO_NRR.2.3

1624 The TSF **shall** provide a capability to verify the evidence of receipt of information to [selection:
1625 *originator, recipient, [assignment: list of third parties]*] given [assignment: *limitations on the*
1626 *evidence of receipt*].

1627

1628 **9 Class FCS: Cryptographic support**

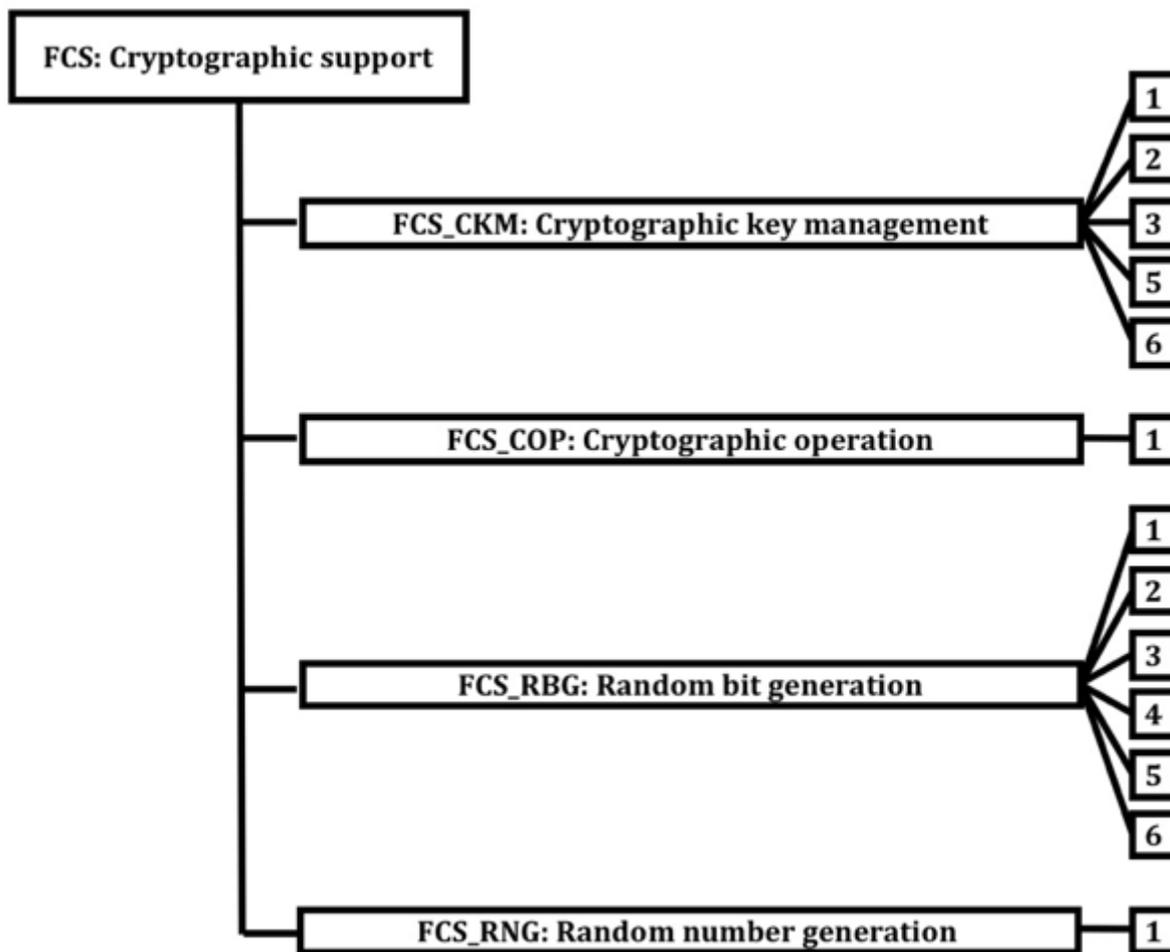
1629 **9.1 Class description**

1630 The TSF **may** employ cryptographic functionality to help satisfy several high-level security
 1631 objectives. These include (but are not limited to): identification and authentication, non-
 1632 repudiation, trusted path, trusted channel, and data separation. This class is used when the TOE
 1633 implements cryptographic functions, the implementation of which **could** be in hardware,
 1634 firmware and/or software.

1635 The FCS: Cryptographic support class is composed of four families.

1636 Figure 17 shows the decomposition of this class, it's families and components. Elements are not
 1637 shown in the figure.

1638 Annex E provides explanatory information for this class and **should** be consulted when using
 1639 the components identified in this class.



1640

1641 **Figure 17 — FCS: Cryptographic support class decomposition**

1642 **9.2 Cryptographic key management (FCS_CKM)**

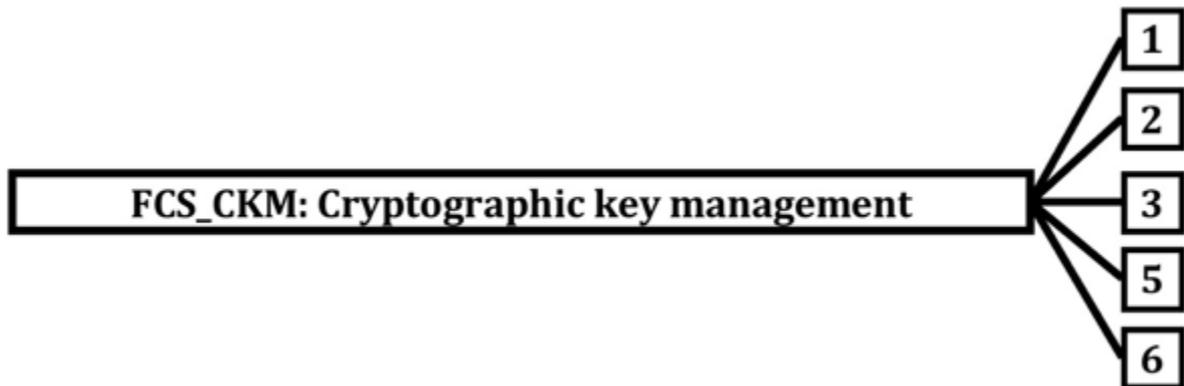
1643 **9.2.1 Family behaviour**

1644 Cryptographic keys must be managed throughout their life cycle. This family is intended to
 1645 support that lifecycle and consequently defines requirements for the following activities:
 1646 cryptographic key generation, cryptographic key derivation, cryptographic key distribution,
 1647 cryptographic key access and timing and event of cryptographic key destruction. This family

1648 **should** be included whenever there are functional requirements for the management of
 1649 cryptographic keys.

1650 9.2.2 Components leveling and description

1651 Figure 18 shows the component leveling for this family.



1652

1653 **Figure 18 — FCS_CKM: Component leveling**

1654 FCS_CKM.1 Cryptographic key generation, requires cryptographic keys to be generated in
 1655 accordance with a specified algorithm and key sizes which **can** be based on an assigned
 1656 standard.

1657 FCS_CKM.2 Cryptographic key distribution, requires cryptographic keys to be distributed in
 1658 accordance with a specified distribution method which **can** be based on an assigned standard.

1659 FCS_CKM.3 Cryptographic key access requires access to cryptographic keys to be performed in
 1660 accordance with a specified access method which **can** be based on an assigned standard.

1661 FCS_CKM.5 Cryptographic key derivation, requires that the methods, standards, and parameters
 1662 for key-derivation are specified.

1663 FCS_CKM.6 Timing and event of cryptographic key destruction, requires cryptographic keys to
 1664 be destroyed in accordance with specified destruction methods which **can** be based on an
 1665 assigned standard.

1666 NOTE Previous editions of this standard specified FCS_CKM.4 which has been deprecated in this edition of
 1667 ISO/IEC 15408-2. In order to preserve consistency when applying different editions of ISO/IEC 15408-2 the
 1668 component number has not been re-used.

1669 9.2.3 Management of FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.5, CKM.6

1670 The following actions **could** be considered for the management functions in FMT:

1671 a) There are no management activities foreseen.

1672 9.2.4 Audit of FCS_CKM.1, FCS_CKM.2, FCS_CKM.3, FCS_CKM.5, CKM.6

1673 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 1674 in the PP/ST:

1675 a) Minimal: Success and failure of the activity.

1676 b) Basic: The object attribute(s), and object value(s) excluding any sensitive
 1677 information

1678 9.2.5 FCS_CKM.1 Cryptographic key generation

1679 Component relationships

1680 Hierarchical to: No other components.

1721 **9.2.8 FCS_CKM.4 Cryptographic key destruction**

1722 The component has been deprecated. See FCS_CKM.6 Timing and event of cryptographic key
1723 destruction instead.

1724 **9.2.9 FCS_CKM.5 Cryptographic key derivation**1725 **Component relationships**

1726	Hierarchical to:	No other components.
1727	Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or
1728		FCS_COP.1 Cryptographic operation]
1729		FCS_CKM.6 Timing and event of cryptographic key
1730		destruction

1731 **FCS_CKM.5.1**

1732 **The TSF shall derive cryptographic keys [assignment: *key type*] from [selection: *input***
1733 ***parameters*] in accordance with a specified key derivation algorithm [selection: *key***
1734 ***derivation algorithm*] and specified cryptographic key sizes [selection: *list of key sizes***
1735 **that meet the following: [assignment: *list of standards*].**

1736 NOTE See E.2.5.1. for information on using this component.

1737 **9.2.10 FCS_CKM.6 Timing and event of cryptographic key destruction**1738 **Component relationships**

1739	Hierarchical to:	No other components
1740	Dependencies:	[FDP_ITC.1 Import of user data without security
1741		attributes, or
1742		FDP_ITC.2 Import of user data with security
1743		attributes, or
1744		FCS_CKM.1 Cryptographic key generation]

1745 **FCS_CKM.6.1**

1746 **The TSF shall destroy [assignment: *list of cryptographic keys (including keying material)*]**
1747 **when [selection: *no longer needed*, [assignment: *other circumstances for key or key***
1748 ***material destruction*]].**

1749 **FCS_CKM.6.2**

1750 **The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1**
1751 **in accordance with a specified cryptographic key destruction method [assignment:**
1752 ***cryptographic key destruction method*] that meets the following: [assignment: *list of***
1753 ***standards*].**

1754 **9.3 Cryptographic operation (FCS_COP)**1755 **9.3.1 Family behaviour**

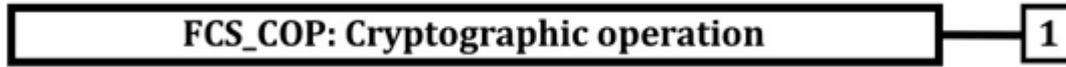
1756 In order for a cryptographic operation to function correctly, the operation must be performed
1757 in accordance with a specified algorithm and with a cryptographic key of a specified size. This
1758 family **should** be included whenever there are requirements for cryptographic operations to be
1759 performed.

1760 Typical cryptographic operations include data encryption and/or decryption, digital signature
1761 generation and/or verification, cryptographic checksum generation for integrity and/or

1762 verification of checksum, secure hash (message digest), cryptographic key encryption and/or
 1763 decryption, and cryptographic key agreement.

1764 **9.3.2 Components leveling and description**

1765 Figure 19 shows the component leveling for this family.



1766

1767 **Figure 19 — FCS_COP: Component leveling**

1768 FCS_COP.1 Cryptographic operation, requires a cryptographic operation to be performed in
 1769 accordance with a specified algorithm and with a cryptographic key of specified sizes. The
 1770 specified algorithm and cryptographic key sizes **can** be based on an assigned standard.

1771 **9.3.3 Management of FCS_COP.1**

1772 The following actions **could** be considered for the management functions in FCS:

- 1773 a) There are no management activities foreseen.

1774 **9.3.4 Audit of FCS_COP.1**

1775 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 1776 in the PP/ST:

- 1777 a) Minimal: Success and failure, and the type of cryptographic operation.
- 1778 b) Basic: Any applicable cryptographic mode(s) of operation, subject attributes and
 1779 object attributes.

1780 **9.3.5 FCS_COP.1 Cryptographic operation**

1781 **Component relationships**

1782	Hierarchical to:	No other components.
1783	Dependencies:	[FDP_ITC.1 Import of user data without security
1784		attributes, or
1785		FDP_ITC.2 Import of user data with security
1786		attributes, or
1787		FCS_CKM.1 Cryptographic key generation, or
1788		FCS_CKM.5 Cryptographic key derivation]
1789		FCS_CKM.3 Cryptographic key access

1790 **FCS_COP.1.1**

1791 **The TSF shall perform [assignment: *list of cryptographic operations*] in accordance with a**
 1792 **specified cryptographic algorithm [assignment: *cryptographic algorithm*] and**
 1793 **cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following:**
 1794 **[assignment: *list of standards*].**

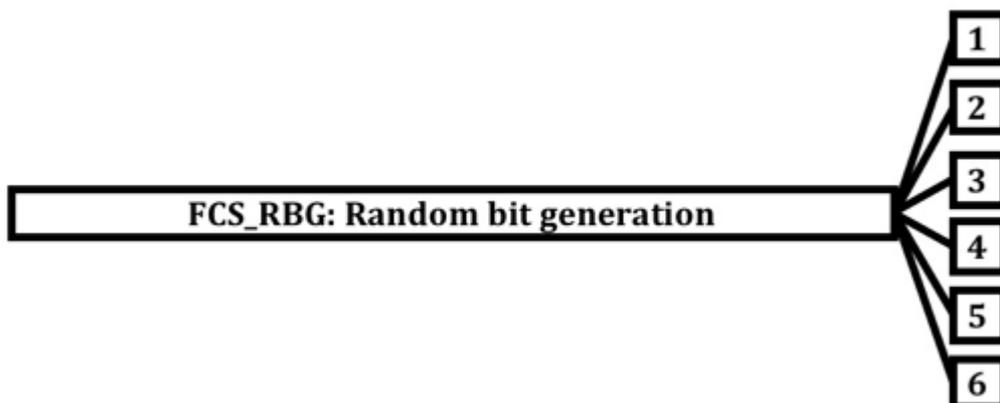
1795 **9.4 Random bit generation (FCS_RBG)**

1796 **9.4.1 Family behaviour**

1797 Components in this family address the requirements for random bit/number generation.

1798 **9.4.2 Components leveling and description**

1799 Figure 20 shows the component leveling for this family.



1800

1801 **Figure 20 — FCS_RBG: Component leveling**1802 FCS_RBG.1 Random bit generation (RBG) requires random bit generation to be performed in
1803 accordance with selected standards.1804 FCS_RBG.2 Random bit generation (external seeding) gives requirements for seeding by an
1805 external (outside the TOE) entropy source.1806 FCS_RBG.3 Random bit generation (internal seeding – single source) gives requirements for
1807 seeding using a TSF entropy source.1808 FCS_RBG.4 Random bit generation (internal seeding – multiple sources) gives requirements for
1809 seeding using multiple TSF entropy sources.1810 FCS_RBG.5 Random bit generation (combining entropy sources) gives requirements for
1811 combining multiple entropy sources (multiple internal sources, internal and external).1812 FCS_RBG.6 Random bit generation service requires random numbers to be supplied over an
1813 external interface as a service to other entities.1814 **9.4.3 Management of FCS_RBG.1, FCS_RBG.2, FCS_RBG.3, FCS_RBG.4, FCS_RBG.5,**
1815 **FCS_RBG.6**1816 The following actions **could** be considered for the management functions in FMT:

1817 a) There are no management activities foreseen.

1818 **9.4.4 Audit of FCS_RBG.1, FCS_RBG.2**1819 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1820 in the PP/ST:1821 a) Minimal: failure of the randomization process, failure to initialize or reseed (as
1822 supported by the technology)1823 **9.4.5 Audit of FCS_RBG.3, FCS_RBG.4, FCS_RBG.6, FCS_RBG.6**1824 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
1825 in the PP/ST:

1826 a) There are no auditable events foreseen.

1827 **9.4.6 FCS_RBG.1 Random bit generation (RBG)**

1828 **Component relationships**

1829 Hierarchical to: No other components

1830 Dependencies: [FCS_RBG.2 Random bit generation (external seeding), or

1831 FCS_RBG.3 Random bit generation (internal seeding – single source)]

1832

1833

1834 FPT_FLS.1 Failure with preservation of secure state

1835 FPT_TST.1 TSF self-testing

1836 **FCS_RBG.1.1**

1837 **The TSF shall perform deterministic random bit generation services using [assignment: RBG algorithm] in accordance with [assignment: list of standards] after initialization with a seed.**

1838

1839

1840 **FCS_RBG.1.2**

1841 ~~The TSF shall initialize and update the RBG state using a noise source as shown in the RBG State Update Table.~~

1842

1843 **The TSF shall initialize and update the RBG state using a noise source under a specified condition as shown in the RBG State Update Table.**

1844

1845 **Editors; Note:**

1846 Please see CD1 JP4 / 027 in regard to this change. However, please note that this change was accepted pending review by the CCDB Crypto WG. No review has been received yet from the CCDB.

1847

1848 If comments are not received during the CD2 commenting period then this change will be accepted in the next draft.

1849

1850 **Table 1 – 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, unstantiate+stantiate]	[selection: on demand; on the condition: [assignment: condition]; after [assignment: time]]	[assignment: list of standards]

1851

1852 **9.4.7 FCS_RBG.2 Random bit generation (external seeding)**

1853 **Component relationships**

1854 Hierarchical to: No other components.

1855 Dependencies: FCS_RBG.1 Random bit generation (RBG)

1856 **FCS_RBG.2.1**

1857 **The TSF shall be able to accept a minimum input of [assignment: *minimum input length***
 1858 ***greater than zero*] from an external interface for the purpose of seed generation.**

1859 **9.4.8 FCS_RBG.3 Random bit generation (internal seeding – single source)**1860 **Component relationships**

1861 Hierarchical to: No other components

1862 Dependencies: FCS_RBG.1 Random bit generation (RBG)

1863 **FCS_RBG.3.1**

1864 **The TSF shall be able to seed the RBG using a single [selection: *TSF software-based noise***
 1865 ***source, TSF hardware-based noise source*] with a minimum of [assignment: *number of***
 1866 ***bits*] bits of min-entropy.**

1867 **9.4.9 FCS_RBG.4 Random bit generation (internal seeding – multiple sources)**1868 **Component relationships**

1869 Hierarchical to: No other components

1870 Dependencies: FCS_RBG.1 Random bit generation (RBG)

1871 FCS_RBG.3 Random bit generation (internal seeding
 1872 – single source)

1873 **FCS_RBG.4.1**

1874 **The TSF shall be able to seed the RBG using [selection: [assignment: *number*] *TSF software-***
 1875 ***based noise source(s)*, [assignment: *number*] *TSF hardware-based noise source(s)*].**

1876 **9.4.10 FCS_RBG.5 Random bit generation (combining entropy sources)**1877 **Component relationships**

1878 Hierarchical to: No other components.

1879 Dependencies: FCS_RBG.1 Random bit generation (RBG)

1880 [FCS_RBG.2 Random bit generation (external
 1881 seeding), or

1882 FCS_RBG.3 Random bit generation (internal seeding
 1883 – single source)]

1884 **FCS_RBG.5.1 Combining entropy sources**

1885 **The TSF shall [assignment: *combining operation*] [selection: *TSF entropy source(s), TOE***
 1886 ***external entropy source(s)*] to create the entropy input into the derivation function as**
 1887 **defined in [assignment: *list of standards*], resulting in a minimum of [assignment:**
 1888 ***number of bits*] bits of min-entropy.**

1889 **9.4.11 FCS_RBG.6 Random bit generation service**1890 **Component relationships**

1891 Hierarchical to: No other components.

1892 Dependencies: FCS_RBG.1 Random bit generation (RBG)

1893 [FCS_RBG.2 Random bit generation (external
 1894 seeding), or
 1895 FCS_RBG.3 Random bit generation (internal seeding
 1896 – single source)]

1897 **FCS_RBG.6.1**

1898 **The TSF shall provide a [selection: *hardware, software, [assignment: other interface type]]***
 1899 **interface to make the RBG output, as specified in FCS_RBG.1 Random bit generation**
 1900 **(RBG), available as a service to entities outside of the TOE.**

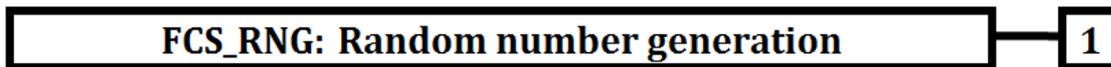
1901 **9.5 Generation of random numbers (FCS_RNG)**

1902 **9.5.1 Family behaviour**

1903 This family defines quality requirements for the generation of random numbers which are
 1904 intended to be use for cryptographic purposes.

1905 **9.5.2 Components leveling and description**

1906 Figure 21 shows the component leveling for this family.



1907

1908 **Figure 21 — FCS_RNG: Component leveling**

1909 FCS_RNG.1 Random number generation requires that random numbers meet a defined quality
 1910 metric.

1911 **9.5.3 Management of FCS_RNG.1**

1912 There are no management activities foreseen.

1913 **9.5.4 Audit of FCS_RNG.1**

1914 There are no actions defined to be auditable.

1915 **9.5.5 FCS_RNG.1 Random number generation**

1916 **Component relationships**

1917 Hierarchical to: No other components.

1918 Dependencies: No dependencies.

1919 **FCS_RNG.1.1**

1920 **The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid***
 1921 ***physical, hybrid deterministic*] random number generator that implements: [assignment:**
 1922 ***list of security capabilities*].**

1923 **FCS_RNG.1.2**

1924 **The TSF shall provide [selection: *bits, octets of bits, numbers* [assignment: *format of the***
 1925 ***numbers*]] that meet [assignment: *a defined quality metric*].**

1926

1927 **10 Class FDP: User data protection**

1928 **10.1 Class description**

1929 This class contains families specifying requirements related to protecting user data. FDP: User
 1930 data protection is split into four groups of families (listed below) that address user data within
 1931 a TOE, during import, export, and storage as well as security attributes directly related to user
 1932 data.

1933 The families in this class are organized into four groups:

1934 a) User data protection security function policies:

1935 — Access control policy (FDP_ACC); and

1936 — Information flow control policy (FDP_IFC).

1937 Components in these families permit the PP/ST author to name the user data
 1938 protection security function policies and define the scope of control of the policy,
 1939 necessary to address the security objectives. The names of these policies are meant
 1940 to be used throughout the remainder of the functional components that have an
 1941 operation that calls for an assignment or selection of an "access control SFP" or an
 1942 "information flow control SFP". The rules that define the functionality of the named
 1943 access control and information flow control SFPs will be defined in the Access
 1944 control functions (FDP_ACF) and Information flow control functions (FDP_IFF)
 1945 families (respectively).

1946 b) Forms of user data protection:

1947 — Access control functions (FDP_ACF);

1948 — Information flow control functions (FDP_IFF);

1949 — Internal TOE transfer (FDP_ITT);

1950 — Information Retention Control (FDP_IRC)

1951 — Residual information protection (FDP_RIP);

1952 — Rollback (FDP_ROL);

1953 — Stored data confidentiality (FDP_SDC); and

1954 — Stored data integrity (FDP_SDI).

1955 c) Off-line storage, import and export:

1956 — Data authentication (FDP_DAU);

1957 — Export from the TOE (FDP_ETC);

1958 — Import from outside of the TOE (FDP_ITC).

1959 Components in these families address the trustworthy transfer into or out of the
 1960 TOE.

1961 d) Inter-TSF communication:

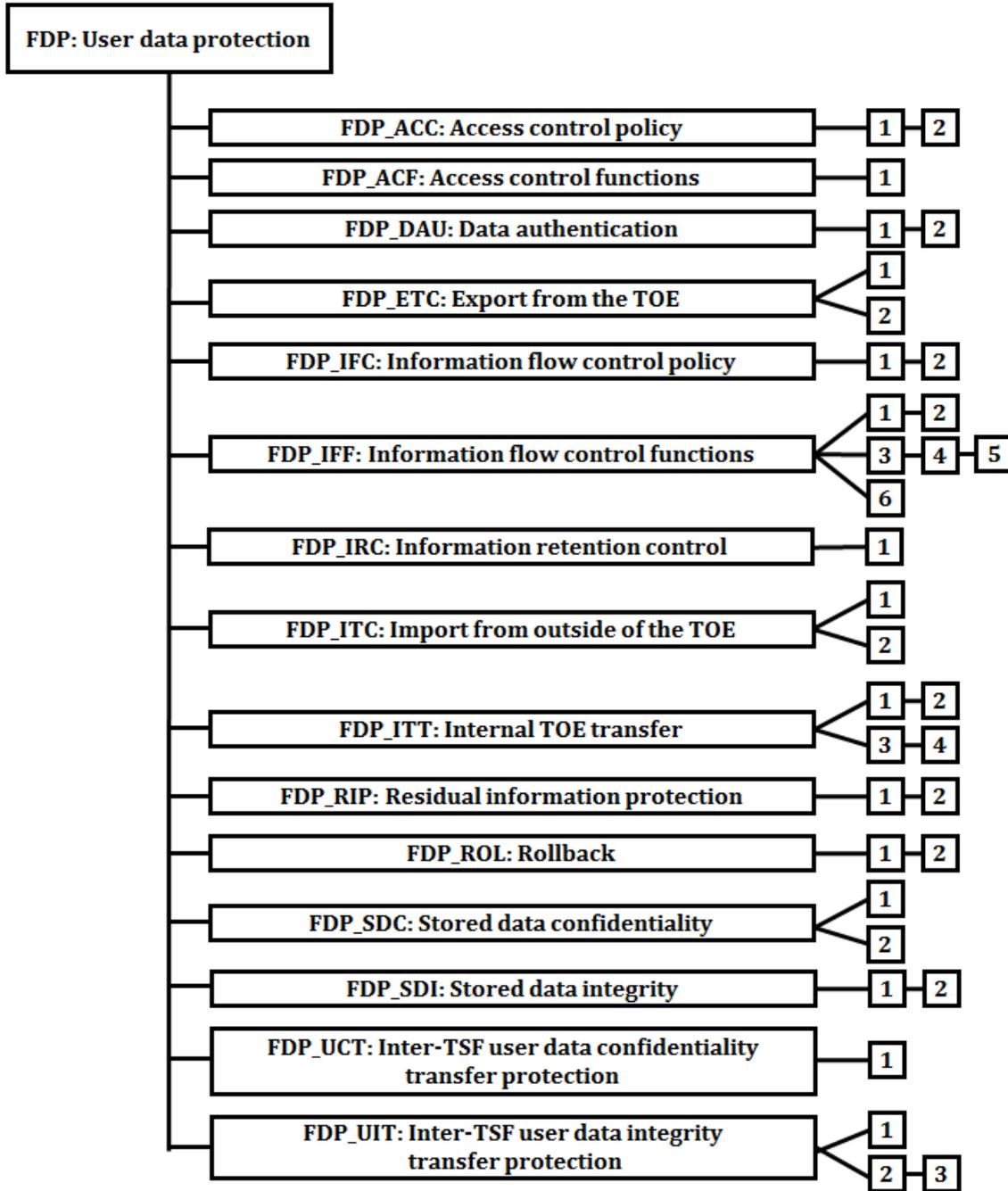
1962 — Inter-TSF user data confidentiality transfer protection (FDP_UCT); and

1963 — Inter-TSF user data integrity transfer protection (FDP_UIT).

1964 — Components in these families address communication between the TSF of the
 1965 TOE and another trusted IT product.

1966 Figure 22 shows the decomposition of this class, it's families and components. Elements are not
 1967 shown in the figure.

1968 Annex F provides explanatory information for this class and **should** be consulted when using
 1969 the components identified in this class.



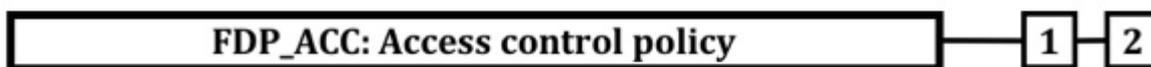
1970
 1971 **Figure 22 — FDP: User data protection class decomposition**

1972 **10.2 Access control policy (FDP_ACC)**

1973 **10.2.1 Family behaviour**

1974 This family identifies the access control SFPs (by name) and defines the scope of control of the
 1975 policies that form the identified access control portion of the SFRs related to the SFP. This scope
 1976 of control is characterized by three sets: the subjects under control of the policy, the objects
 1977 under control of the policy, and the operations among controlled subjects and controlled
 1978 objects that are covered by the policy. The criteria allow multiple policies to exist, each having a
 1979 unique name. This is accomplished by iterating components from this family once for each

1980 named access control policy. The rules that define the functionality of an access control SFP will
 1981 be defined by other families such as Access control functions (FDP_ACF) and Export from the
 1982 TOE (FDP_ETC). The names of the access control SFPs identified here in Access control policy
 1983 (FDP_ACC) are meant to be used throughout the remainder of the functional components that
 1984 have an operation that calls for an assignment or selection of an “access control SFP.”
 1985 Components leveling and description
 1986 Figure 23 shows the component leveling for this family.



1987 **Figure 23 — FDP_ACC: Component leveling**

1988 FDP_ACC.1 Subset access control, requires that each identified access control SFP be in place for
 1989 a subset of the possible operations on a subset of the objects in the TOE.

1990 FDP_ACC.2 Complete access control, requires that each identified access control SFP cover all
 1991 operations on subjects and objects covered by that SFP. It further requires that all objects and
 1992 operations protected by the TSF are covered by at least one identified access control SFP.

1993 **10.2.2 Management of FDP_ACC.1, FDP_ACC.2**

1994 The following actions **could** be considered for the management functions in FMT:

1995 a) There are no management activities foreseen.

1996 **10.2.3 Audit of FDP_ACC.1, FDP_ACC.2**

1997 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 1998 in the PP/ST:

1999 a) There are no auditable events foreseen.

2000 **10.2.4 FDP_ACC.1 Subset access control**

2001 **Component relationships**

2002 Hierarchical to: No other components.

2003 Dependencies: FDP_ACF.1 Security attribute-based access control

2004 **FDP_ACC.1.1**

2005 **The TSF shall enforce the [assignment: access control SFP] on [assignment: list of subjects,**
 2006 **objects, and operations among subjects and objects covered by the SFP].**

2007 **10.2.5 FDP_ACC.2 Complete access control**

2008 **Component relationships**

2009 Hierarchical to: FDP_ACC.1 Subset access control

2010 Dependencies: FDP_ACF.1 Security attribute-based access control

2011 **FDP_ACC.2.1**

2012 The TSF **shall** enforce the [assignment: access control SFP] on [assignment: list of subjects and
 2013 **objects] and all operations among subjects and objects covered by the SFP.**

2014 **FDP_ACC.2.2**

2015 **The TSF shall ensure that all operations between any subject controlled by the TSF and**
 2016 **any object controlled by the TSF are covered by an access control SFP.**

2017 **10.3 Access control functions (FDP_ACF)**

2018 **10.3.1 Family behaviour**

2019 This family describes the rules for the specific functions that can implement an access control
 2020 policy named in Access control policy (FDP_ACC). Access control policy (FDP_ACC) specifies the
 2021 scope of control of the policy.

2022 **10.3.2 Components leveling and description**

2023 Figure 24 shows the component leveling for this family.



2024

2025 **Figure 24 — FDP_ACF: Component leveling**

2026 This family addresses security attribute usage and characteristics of policies. The component
 2027 within this family is meant to be used to describe the rules for the function that implements the
 2028 SFP as identified in Access control policy (FDP_ACC). The PP/ST author may also iterate this
 2029 component to address multiple policies in the TOE.

2030 FDP_ACF.1 Security attribute-based access control Security attribute-based access control
 2031 allows the TSF to enforce access based upon security attributes and named groups of attributes.
 2032 Furthermore, the TSF may have the ability to explicitly authorize or deny access to an object
 2033 based upon security attributes.

2034 **10.3.3 Management of FDP_ACF.1**

2035 The following actions could be considered for the management functions in FMT:

- 2036 a) Managing the attributes used to make explicit access or denial-based decisions.

2037 **10.3.4 Audit of FDP_ACF.1**

2038 The following actions should be auditable if FAU_GEN Security audit data generation is included
 2039 in the PP/ST:

- 2040 a) Minimal: Successful requests to perform an operation on an object covered by the
 2041 SFP.
- 2042 b) Basic: All requests to perform an operation on an object covered by the SFP.
- 2043 c) Detailed: The specific security attributes used in making an access check.

2044 **10.3.5 FDP_ACF.1 Security attribute-based access control**

2045 **Component relationships**

2046	Hierarchical to:	No other components.
2047	Dependencies:	FDP_ACC.1 Subset access control
2048		FMT_MSA.3 Static attribute

2049 **FDP_ACF.1.1**

2050 The TSF **shall** enforce the [assignment: *access control SFP*] to objects based on the
 2051 following: [assignment: *list of subjects and objects controlled under the indicated SFP, and*
 2052 *for each, the SFP-relevant security attributes, or named groups of SFP-relevant security*
 2053 *attributes*].

2054 **FDP_ACF.1.2**

2055 The TSF **shall** enforce the following rules to determine if an operation among controlled
 2056 subjects and controlled objects is allowed: [assignment: *rules governing access among*
 2057 *controlled subjects and controlled objects using controlled operations on controlled*
 2058 *objects*].

2059 **FDP_ACF.1.3**

2060 The TSF **shall** explicitly authorize access of subjects to objects based on the following
 2061 additional rules: [assignment: *rules, based on security attributes, that explicitly authorize*
 2062 *access of subjects to objects*].

2063 **FDP_ACF.1.4**

2064 The TSF **shall** explicitly deny access of subjects to objects based on the following
 2065 additional rules: [assignment: *rules, based on security attributes, that explicitly deny*
 2066 *access of subjects to objects*].

2067 **10.4 Data authentication (FDP_DAU)**2068 **10.4.1 Family behaviour**

2069 Data authentication permits an entity to accept responsibility for the authenticity of
 2070 information. This family provides a method of providing a guarantee of the validity of a specific
 2071 unit of data that **can** be subsequently used to verify that the information content has not been
 2072 forged or fraudulently modified. In contrast to FAU: Security audit, this family is intended to be
 2073 applied to "static" data rather than data that is being transferred.

2074 **10.4.2 Components leveling and description**

2075 Figure 25 shows the component leveling for this family.



2077 **Figure 25 — FDP_DAU: Component leveling**

2078 FDP_DAU.1 Basic Data Authentication, requires that the TSF is capable of generating a
 2079 guarantee of authenticity of the information content of objects.

2080 FDP_DAU.2 Data Authentication with Identity of Guarantor additionally requires that the TSF is
 2081 capable of establishing the identity of the subject who provided the guarantee of authenticity.

2082 **10.4.3 Management of FDP_DAU.1, FDP_DAU.2**

2083 The following actions **could** be considered for the management functions in FMT:

- 2084 a) The assignment or modification of the objects for which data authentication **may**
 2085 apply **could** be configurable.

2086 **10.4.4 Audit of FDP_DAU.1**

2087 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2088 in the PP/ST:

- 2089 a) Minimal: Successful generation of validity evidence.
- 2090 b) Basic: Unsuccessful generation of validity evidence.
- 2091 c) Detailed: The identity of the subject that requested the evidence.

2092 **10.4.5 Audit of FDP_DAU.2**

2093 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2094 in the PP/ST:

- 2095 a) Minimal: Successful generation of validity evidence.
- 2096 b) Basic: Unsuccessful generation of validity evidence.
- 2097 c) Detailed: The identity of the subject that requested the evidence.
- 2098 d) Detailed: The identity of the subject that generated the evidence.

2099 **10.4.6 FDP_DAU.1 Basic Data Authentication**

2100 **Component relationships**

- 2101 Hierarchical to: No other components.
- 2102 Dependencies: No dependencies.

2103 **FDP_DAU.1.1**

2104 **The TSF **shall** provide a capability to generate evidence that **can** be used as a guarantee of**
2105 **the validity of [assignment: *list of objects or information types*].**

2106 **FDP_DAU.1.2**

2107 **The TSF **shall** provide [assignment: *list of subjects*] with the ability to verify evidence of**
2108 **the validity of the indicated information.**

2109 **10.4.7 FDP_DAU.2 Data Authentication with Identity of Guarantor**

2110 **Component relationships**

- 2111 Hierarchical to: FDP_DAU.1 Basic Data Authentication
- 2112 Dependencies: FIA_UID.1 Timing of identification

2113 **FDP_DAU.2.1**

2114 The TSF **shall** provide a capability to generate evidence that **can** be used as a guarantee of the
2115 validity of [assignment: *list of objects or information types*].

2116 **FDP_DAU.2.2**

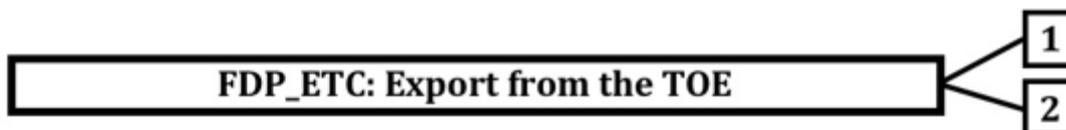
2117 The TSF **shall** provide [assignment: *list of subjects*] with the ability to verify evidence of the
2118 validity of the indicated information **and the identity of the user that generated the**
2119 **evidence.**

2120 **10.5 Export from the TOE (FDP_ETC)**2121 **10.5.1 Family behaviour**

2122 This family defines functions for TSF-mediated exporting of user data from the TOE such that its
 2123 security attributes and protection either **can** be explicitly preserved or **can** be ignored once it
 2124 has been exported. It is concerned with limitations on export and with the association of
 2125 security attributes with the exported user data.

2126 **10.5.2 Components leveling and description**

2127 Figure 26 shows the component leveling for this family.



2128

2129 **Figure 26 — FDP_ETC: Component leveling**

2130 FDP_ETC.1 Export of user data without security attributes, requires that the TSF enforces the
 2131 appropriate SFPs when exporting user data outside the TSF. User data that is exported by this
 2132 function is exported without its associated security attributes.

2133 FDP_ETC.2 Export of user data with security attributes, requires that the TSF enforces the
 2134 appropriate SFPs using a function that accurately and unambiguously associates security
 2135 attributes with the user data that is exported.

2136 **10.5.3 Management of FDP_ETC.1**

2137 The following actions **could** be considered for the management functions in FMT:

2138 a) There are no management activities foreseen.

2139 **10.5.4 Management of FDP_ETC.2**

2140 The following actions **could** be considered for the management functions in FMT:

2141 a) The additional exportation control rules **could** be configurable by a user in a
 2142 defined role.

2143 **10.5.5 Audit of FDP_ETC.1, FDP_ETC.2**

2144 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 2145 in the PP/ST:

2146 a) Minimal: Successful export of information.

2147 b) Basic: All attempts to export information.

2148 **10.5.6 FDP_ETC.1 Export of user data without security attributes**2149 **Component relationships**

2150 Hierarchical to: No other components.

2151 Dependencies: [FDP_ACC.1 Subset access control, or
 2152 FDP_IFC.1 Subset information flow control]

2153 **FDP_ETC.1.1**

2154 **The TSF shall enforce the [assignment: access control SFP(s) and/or information flow**
 2155 **control SFP(s)] when exporting user data, controlled under the SFP(s), outside of the TOE.**

2156 **FDP_ETC.1.2**

2157 **The TSF shall export the user data without the user data's associated security attributes.**

2158 **10.5.7 FDP_ETC.2 Export of user data with security attributes**

2159 **Component relationships**

2160 Hierarchical to: No other components.

2161 Dependencies: [FDP_ACC.1 Subset access control, or
2162 FDP_IFC.1 Subset information flow control]

2163 **FDP_ETC.2.1**

2164 **The TSF shall enforce the [assignment: *access control SFP(s) and/or information flow***
2165 ***control SFP(s)*] when exporting user data, controlled under the SFP(s), outside of the TOE.**

2166 **FDP_ETC.2.2**

2167 **The TSF shall export the user data with the user data's associated security attributes.**

2168 **FDP_ETC.2.3**

2169 **The TSF shall ensure that the security attributes, when exported outside the TOE, are**
2170 **unambiguously associated with the exported user data.**

2171 **FDP_ETC.2.4**

2172 **The TSF shall enforce the following rules when user data is exported from the TOE:**
2173 **[assignment: *additional exportation control rules*].**

2174 **10.6 Information flow control policy (FDP_IFC)**

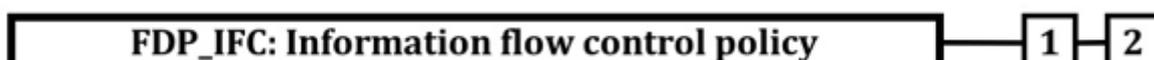
2175 **10.6.1 Family behaviour**

2176 This family identifies the information flow control SFPs (by name) and defines the scope of
2177 control for each named information flow control SFP. This scope of control is characterized by
2178 three sets: the subjects under control of the policy, the information under control of the policy,
2179 and operations which cause controlled information to flow to and from controlled subjects
2180 covered by the policy. The criteria allow multiple policies to exist, each having a unique name.
2181 This is accomplished by iterating components from this family once for each named information
2182 flow control policy. The rules that define the functionality of an information flow control SFP
2183 will be defined by other families such as Information flow control functions (FDP_IFF) and
2184 Export from the TOE (FDP_ETC). The names of the information flow control SFPs identified here
2185 in Information flow control policy (FDP_IFC) are meant to be used throughout the remainder of
2186 the functional components that have an operation that calls for an assignment or selection of an
2187 "information flow control SFP."

2188 The TSF mechanism controls the flow of information in accordance with the information flow
2189 control SFP. Operations that would change the security attributes of information are not
2190 generally permitted as this would be in violation of an information flow control SFP. However,
2191 such operations may be permitted as exceptions to the information flow control SFP if explicitly
2192 specified.

2193 **10.6.2 Components leveling and description**

2194 Figure 27 shows the component leveling for this family.



2195

Figure 27 — FDP_IFC: Component leveling

2196 FDP_IFC.1 Subset information flow control, requires that each identified information flow
2197 control SFPs be in place for a subset of the possible operations on a subset of information flows
2198 in the TOE.

2199 FDP_IFC.2 Complete information flow control, requires that each identified information flow
2200 control SFP cover all operations on subjects and information covered by that SFP. It further
2201 requires that all information flows and operations controlled by the TSF are covered by at least
2202 one identified information flow control SFP.

2203 10.6.3 Management of FDP_IFC.1, FDP_IFC.2

2204 The following actions **could** be considered for the management functions in FMT:

2205 a) There are no management activities foreseen.

2206 10.6.4 Audit of FDP_IFC.1, FDP_IFC.2

2207 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2208 in the PP/ST:

2209 a) There are no auditable events foreseen.

2210 10.6.5 FDP_IFC.1 Subset information flow control**2211 Component relationships**

2212 Hierarchical to: No other components.

2213 Dependencies: FDP_IFF.1 Simple security attributes

2214 FDP_IFC.1.1

2215 The TSF **shall** enforce the [assignment: *information flow control SFP*] on [assignment: *list*
2216 *of subjects, information, and operations that cause controlled information to flow to and*
2217 *from controlled subjects covered by the SFP*].

2218 10.6.6 FDP_IFC.2 Complete information flow control**2219 Component relationships**

2220 Hierarchical to: FDP_IFC.1 Subset information flow control

2221 Dependencies: FDP_IFF.1 Simple security attributes

2222 FDP_IFC.2.1

2223 The TSF **shall** enforce the [assignment: *information flow control SFP*] on [assignment: *list of*
2224 *subjects and information*] **and all** operations that cause **that** information to flow to and from
2225 subjects covered by the SFP.

2226 FDP_IFC.2.2

2227 The TSF **shall** ensure that all operations that cause any information in the TOE to flow to
2228 and from any subject in the TOE are covered by an information flow control SFP.

2229 10.7 Information flow control functions (FDP_IFF)**2230 10.7.1 Family behaviour**

2231 This family describes the rules for the specific functions that **can** implement the information
2232 flow control SFPs named in Information flow control policy (FDP_IFC), which also specifies the

2233 scope of control of the policy. It consists of two kinds of requirements: one addressing the
 2234 common information flow function issues, and a second addressing illicit information flows (i.e.
 2235 covert channels). This division arises because the issues concerning illicit information flows are,
 2236 in some sense, orthogonal to the rest of an information flow control SFP. By their nature, they
 2237 circumvent the information flow control SFP resulting in a violation of the policy. As such, they
 2238 require special functions to either limit or prevent their occurrence.

2239 **10.7.2 Components leveling and description**

2240 Figure 28 shows the component leveling for this family.



2241 **Figure 28 — FDP_IFF: Component leveling**

2242 FDP_IFF.1 Simple security attributes, requires security attributes on information, and on
 2243 subjects that cause that information to flow and on subjects that act as recipients of that
 2244 information. It specifies the rules that must be enforced by the function and describes how
 2245 security attributes are derived by the function.

2246 FDP_IFF.2 Hierarchical security attributes expands on the requirements of FDP_IFF.1 Simple
 2247 security attributes by requiring that all information flow control SFPs in the set of SFRs use
 2248 hierarchical security attributes that form a lattice (as defined in mathematics). FDP_IFF.2.6 is
 2249 derived from the mathematical properties of a lattice. A lattice consists of a set of elements with
 2250 an ordering relationship with the property defined in the first bullet, a least upper bound which
 2251 is the unique element in the set that is greater or equal (in the ordering relationship) than any
 2252 other element of the lattice, and a greatest lower bound, which is the unique element in the set
 2253 that is smaller or equal than any other element of the lattice.

2254 FDP_IFF.3 Limited illicit information flows, requires the SFP to cover illicit information flows,
 2255 but not necessarily eliminate them.

2256 FDP_IFF.4 Partial elimination of illicit information flows, requires the SFP to cover the
 2257 elimination of some (but not necessarily all) illicit information flows.

2258 FDP_IFF.5 No illicit information flows, requires SFP to cover the elimination of all illicit
 2259 information flows.

2260 FDP_IFF.6 Illicit information flow monitoring, requires the SFP to monitor illicit information
 2261 flows for specified and maximum capacities.

2262 **10.7.3 Management of FDP_IFF.1, FDP_IFF.2**

2263 The following actions **could** be considered for the management functions in FMT:

- 2264 a) Managing the attributes used to make explicit access-based decisions.

2265 **10.7.4 Management of FDP_IFF.3, FDP_IFF.4, FDP_IFF.5**

2266 The following actions **could** be considered for the management functions in FMT:

- 2267 a) There are no management activities foreseen.

2268 **10.7.5 Management of FDP_IFF.6**

2269 The following actions **could** be considered for the management functions in FMT:

- 2270 a) The enabling or disabling of the monitoring function.

2271 b) Modification of the maximum capacity at which the monitoring occurs.

2272 10.7.6 Audit of FDP_IFF.1, FDP_IFF.2, FDP_IFF.5

2273 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2274 in the PP/ST:

2275 a) Minimal: Decisions to permit requested information flows.

2276 b) Basic: All decisions on requests for information flow.

2277 c) Detailed: The specific security attributes used in making an information flow
2278 enforcement decision.

2279 d) Detailed: Some specific subsets of the information that has flowed based upon
2280 policy goals.

2281 10.7.7 Audit of FDP_IFF.3, FDP_IFF.4, FDP_IFF.6

2282 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2283 in the PP/ST:

2284 a) Minimal: Decisions to permit requested information flows;

2285 b) Basic: All decisions on requests for information flow;

2286 c) Basic: The use of identified illicit information flow channels;

2287 d) Detailed: The specific security attributes used in making an information flow
2288 enforcement decision;

2289 e) Detailed: Some specific subsets of the information that has flowed based upon
2290 policy goals;

2291 f) Detailed: The use of identified illicit information flow channels with estimated
2292 maximum capacity exceeding a specified value.

2293 10.7.8 FDP_IFF.1 Simple security attributes

2294 Component relationships

2295 Hierarchical to: No other components.

2296 Dependencies: FDP_IFC.1 Subset information flow control

2297 FMT_MSA.3 Static attribute

2298 FDP_IFF.1.1

2299 The TSF **shall** enforce the [assignment: *information flow control SFP*] based on the
2300 following types of subject and information security attributes: [assignment: *list of*
2301 *subjects and information controlled under the indicated SFP, and for each, the security*
2302 *attributes*].

2303 FDP_IFF.1.2

2304 The TSF **shall** permit an information flow between a controlled subject and controlled
2305 information via a controlled operation if the following rules hold: [assignment: *for each*
2306 *operation, the security attribute-based relationship that must hold between subject and*
2307 *information security attributes*].

2308 FDP_IFF.1.3

2309 The TSF **shall** enforce the [assignment: *additional information flow control SFP rules*].

2310 **FDP_IFF.1.4**

2311 **The TSF shall explicitly authorize an information flow based on the following rules:**
 2312 **[assignment: rules, based on security attributes, that explicitly authorize information**
 2313 **flows].**

2314 **FDP_IFF.1.5**

2315 **The TSF shall explicitly deny an information flow based on the following rules:**
 2316 **[assignment: rules, based on security attributes, that explicitly deny information flows].**

2317 **10.7.9 FDP_IFF.2 Hierarchical security attributes**

2318 **Component relationships**

2319 Hierarchical to: FDP_IFF.1 Simple security attributes

2320 Dependencies: FDP_IFC.1 Subset information flow control

2321 FMT_MSA.3 Static attribute

2322 **FDP_IFF.2.1**

2323 **The TSF shall enforce the [assignment: information flow control SFP] based on the following**
 2324 **types of subject and information security attributes: [assignment: list of subjects and**
 2325 **information controlled under the indicated SFP, and for each, the security attributes].**

2326 **FDP_IFF.2.2**

2327 **The TSF shall permit an information flow between a controlled subject and controlled**
 2328 **information via a controlled operation if the following rules, based on the ordering**
 2329 **relationships between security attributes hold: [assignment: for each operation, the security**
 2330 **attribute-based relationship that must hold between subject and information security attributes].**

2331 **FDP_IFF.2.3**

2332 **The TSF shall enforce the [assignment: additional information flow control SFP rules].**

2333 **FDP_IFF.2.4**

2334 **The TSF shall explicitly authorize an information flow based on the following rules:**
 2335 **[assignment: rules, based on security attributes, that explicitly authorize information flows].**

2336 **FDP_IFF.2.5**

2337 **The TSF shall explicitly deny an information flow based on the following rules: [assignment:**
 2338 **rules, based on security attributes, that explicitly deny information flows].**

2339 **FDP_IFF.2.6**

2340 **The TSF shall enforce the following relationships for any two valid information flow**
 2341 **control security attributes:**

- 2342 **a) There exists an ordering function that, given two valid security attributes,**
 2343 **determines if the security attributes are equal, if one security attribute is**
 2344 **greater than the other, or if the security attributes are incomparable; and**
- 2345 **b) There exists a “least upper bound” in the set of security attributes, such that,**
 2346 **given any two valid security attributes, there is a valid security attribute that**
 2347 **is greater than or equal to the two valid security attributes; and**

2348 c) There exists a “greatest lower bound” in the set of security attributes, such
 2349 that, given any two valid security attributes, there is a valid security attribute
 2350 that is not greater than the two valid security attributes.

2351 10.7.10 FDP_IFF.3 Limited illicit information flows

2352 Component relationships

2353 Hierarchical to: No other components.

2354 Dependencies: FDP_IFC.1 Subset information flow control

2355 FDP_IFF.3.1

2356 The TSF **shall** enforce the [assignment: *information flow control SFP*] to limit the capacity
 2357 of [assignment: *types of illicit information flows*] to a [assignment: *maximum capacity*].

2358 10.7.11 FDP_IFF.4 Partial elimination of illicit information flows

2359 Component relationships

2360 Hierarchical to: FDP_IFF.3 Limited illicit information flows

2361 Dependencies: FDP_IFC.1 Subset information flow control

2362 FDP_IFF.4.1

2363 The TSF **shall** enforce the [assignment: *information flow control SFP*] to limit the capacity of
 2364 [assignment: *types of illicit information flows*] to a [assignment: *maximum capacity*].

2365 FDP_IFF.4.2

2366 The TSF **shall** prevent [assignment: *types of illicit information flows*].

2367 10.7.12 FDP_IFF.5 No illicit information flows

2368 Component relationships

2369 Hierarchical to: FDP_IFF.4 Partial elimination of illicit information
 2370 flows

2371 Dependencies: FDP_IFC.1 Subset information flow control

2372 FDP_IFF.5.1

2373 The TSF **shall** ensure that **no illicit information flows exist to circumvent** [assignment:
 2374 *name of information flow control SFP*].

2375 10.7.13 FDP_IFF.6 Illicit information flow monitoring

2376 Component relationships

2377 Hierarchical to: No other components.

2378 Dependencies: FDP_IFC.1 Subset information flow control

2379 FDP_IFF.6.1

2380 The TSF **shall** enforce the [assignment: *information flow control SFP*] to monitor
 2381 [assignment: *types of illicit information flows*] when it exceeds the [assignment: *maximum*
 2382 *capacity*].

2383 **10.8 Information Retention Control (FDP_IRC)**

2384 **10.8.1 Family behaviour**

2385 The “Information retention control” family addresses a basic need in secure information
 2386 processing and storage applications for the secure management of data no longer needed by the
 2387 TOE to perform its operations, but that is still stored in the TOE.

2388 The historical view of IT systems as data storage systems suggested that once entered, data
 2389 would seldom be deleted from the system, and if it was deleted, this would mainly be because of
 2390 storage exhaustion problems.

2391 However, in a multilateral or high security environment it is important to minimize the
 2392 replication of data, as well as the time period during which data is stored in the system. It is also
 2393 possible that users **could** want their IT products to avoid retaining sensitive data that they
 2394 consider to be exploitable by third parties or that could threaten privacy. FDP_IRC **may** help
 2395 users to gain confidence that the product is secure by deleting every copy of the data when it is
 2396 no longer needed.

2397 The FDP_RIP “Residual information protection” family addresses one side of this problem, but
 2398 an explicit requirement on the management of data that is no longer needed is missing.

2399 Of course, competing requirements **may** arise, since some data **may** be needed by the system for
 2400 more operations over a longer time period. Possible solutions to this problem are:

- 2401 — Better protecting the information objects stored in the TOE from access,
- 2402 — Re-requesting the protected information from the user each time it is needed.

2403 Information retention control ensures, that data no longer necessary for the operation of the
 2404 TOE is deleted by the TOE. Components of this family require the PP/ST author to identify the
 2405 TOE operations, including both simple and complex processing and the information objects,
 2406 that are not to be kept in the TOE, that are the subject of those operations.

2407 It is also required that the TOE to keep track of such stored information objects, and to delete
 2408 both the on-line and off-line information objects that are no longer required.

2409 This family sets only requirements on information objects requested for specific activities in the
 2410 TOE operation, and not on general data gathering. The policies which control the collection,
 2411 storage, processing, disclosure, and elimination of general user data stored on the TOE must be
 2412 detailed elsewhere, and are domain of the environmental objectives and organizational policies,
 2413 not of the PP/ST.

2414 When more than one operation requires the presence of a protected object, all operations,
 2415 which refer to the required object **shall** end before deleting it.

2416 **10.8.2 Components leveling and description**

2417 Figure 29 shows the component leveling for this family.



2418

2419 **Figure 29 — FDP_IRC: Component leveling**

2420 FDP_IRC.1 Information retention control requires that the TSF ensure that any copy of a defined
 2421 set of objects in the TOE is deleted when no longer strictly necessary for the operation of the
 2422 TOE, and to identify and define the operations for which the object is required.

2423 **10.8.3 Management of FDP_IRC.1**

2424 The following actions **could** be considered for the management functions in FMT:

2425 a) There are no management actions foreseen.

2426 **10.8.4 Audit of FDP_IRC.1**

2427 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2428 in the PP/ST:

2429 a) There are no auditable events foreseen.

2430 **10.8.5 FDP_IRC.1 Information retention control**

2431 **Component relationships**

2432 Hierarchical to: No other components.

2433 Dependencies: No dependencies.

2434 **FDP_IRC.1.1**

2435 The TSF **shall** ensure enforce the [assignment: *information erasure policy*] on a
2436 [assignment: *list of objects*] required for [assignment: *list of operations*] so that the
2437 selected objects are deleted irreversibly and untraceably from the TOE promptly upon
2438 termination of the selected operations.

2439 **FDP_IRC.1.2**

2440 The TSF **shall** ensure that [assignment: *list of objects*] cannot be accessed after their
2441 release and prior to their irreversible and untraceable deletion.

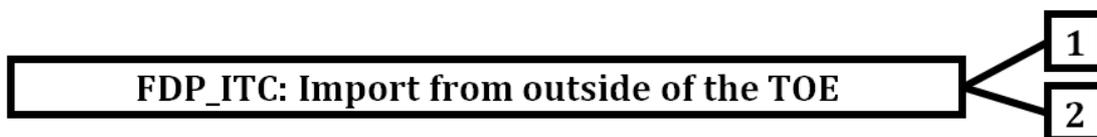
2442 **10.9 Import from outside of the TOE (FDP_ITC)**

2443 **10.9.1 Family behaviour**

2444 This family defines the mechanisms for TSF-mediated importing of user data into the TOE such
2445 that it has appropriate security attributes and is appropriately protected. It is concerned with
2446 limitations on importation, determination of desired security attributes, and interpretation of
2447 security attributes associated with the user data.

2448 **10.9.2 Components leveling and description**

2449 Figure 30 shows the component leveling for this family.



2450

2451 **Figure 30 — FDP_ITC: Component leveling**

2452 FDP_ITC.1 Import of user data without security attributes, requires that the security attributes
2453 correctly represent the user data and are supplied separately from the object.

2454 FDP_ITC.2 Import of user data with security attributes, requires that security attributes
2455 correctly represent the user data and are accurately and unambiguously associated with the
2456 user data imported from outside the TOE.

2457 **10.9.3 Management of FDP_ITC.1, FDP_ITC.2**

2458 The following actions **could** be considered for the management functions in FMT:

2459 a) The modification of the additional control rules used for import.

2460 **10.9.4 Audit of FDP_ITC.1, FDP_ITC.2**

2461 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2462 in the PP/ST:

- 2463 a) Minimal: Successful import of user data, including any security attributes.
- 2464 b) Basic: All attempts to import user data, including any security attributes.
- 2465 c) Detailed: The specification of security attributes for imported user data supplied by
2466 an authorized user.

2467 **10.9.5 FDP_ITC.1 Import of user data without security attributes**

2468 **Component relationships**

2469	Hierarchical to:	No other components.
2470	Dependencies:	[FDP_ACC.1 Subset access control, or
2471		FDP_IFC.1 Subset information flow control]
2472		FMT_MSA.3 Static attribute initialization

2473 **FDP_ITC.1.1**

2474 **The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow***
2475 ***control SFP(s)*] when importing user data, controlled under the SFP, from outside of the**
2476 **TOE.**

2477 **FDP_ITC.1.2**

2478 **The TSF **shall** ignore any security attributes associated with the user data when imported**
2479 **from outside the TOE.**

2480 **FDP_ITC.1.3**

2481 **The TSF **shall** enforce the following rules when importing user data controlled under the**
2482 **SFP from outside the TOE: [assignment: *additional importation control rules*].**

2483 **10.9.6 FDP_ITC.2 Import of user data with security attributes**

2484 **Component relationships**

2485	Hierarchical to:	No other components.
2486	Dependencies:	[FDP_ACC.1 Subset access control, or
2487		FDP_IFC.1 Subset information flow control]
2488		[FTP_ITC.1 Inter-TSF trusted channel, or
2489		FTP_TRP.1 Trusted path]
2490		FPT_TDC.1 Inter-TSF basic TSF data consistency

2491 **FDP_ITC.2.1**

2492 **The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow***
2493 ***control SFP(s)*] when importing user data, controlled under the SFP, from outside of the**
2494 **TOE.**

2495 **FDP_ITC.2.2**

2496 **The TSF **shall** use the security attributes associated with the imported user data.**

2497 **FDP_ITC.2.3**

2498 **The TSF shall ensure that the protocol used provides for the unambiguous association**
 2499 **between the security attributes and the user data received.**

2500 **FDP_ITC.2.4**

2501 **The TSF shall ensure that interpretation of the security attributes of the imported user**
 2502 **data is as intended by the source of the user data.**

2503 **FDP_ITC.2.5**

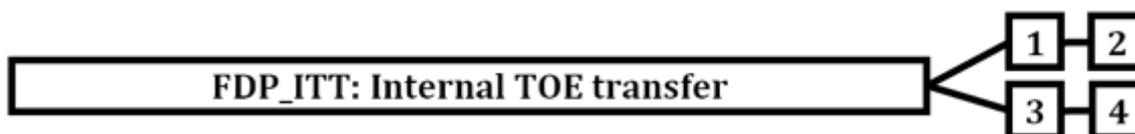
2504 **The TSF shall enforce the following rules when importing user data controlled under the**
 2505 **SFP from outside the TOE: [assignment: *additional importation control rules*].**

2506 **10.10 Internal TOE transfer (FDP_ITT)**2507 **10.10.1 Family behaviour**

2508 This family provides requirements that address protection of user data when it is transferred
 2509 between separated parts of a TOE across an internal channel. This may be contrasted with the
 2510 Inter-TSF user data confidentiality transfer protection (FDP_UCT) and Inter-TSF user data
 2511 integrity transfer protection (FDP_UIT) families, which provide protection for user data when it
 2512 is transferred between distinct TSFs across an external channel, and Export from the TOE
 2513 (FDP_ETC) and Import from outside of the TOE (FDP_ITC), which address TSF-mediated
 2514 transfer of data to or from outside the TOE.

2515 **10.10.2 Components leveling and description**

2516 Figure 31 shows the component leveling for this family.



2517

2518 **Figure 31 — FDP_ITT: Component leveling**

2519 FDP_ITT.1 Basic internal transfer protection, requires that user data be protected when
 2520 transmitted between parts of the TOE.

2521 FDP_ITT.2 Transmission separation by attribute, requires separation of data based on the value
 2522 of SFP-relevant attributes in addition to the first component.

2523 FDP_ITT.3 Integrity monitoring, requires that the TSF monitor user data transmitted between
 2524 parts of the TOE for identified integrity errors.

2525 FDP_ITT.4 Attribute-based integrity monitoring expands on the third component by allowing
 2526 the form of integrity monitoring to differ by SFP-relevant attribute.

2527 **10.10.3 Management of FDP_ITT.1, FDP_ITT.2**

2528 The following actions could be considered for the management functions in FMT:

- 2529 a) If the TSF provides multiple methods to protect user data during transmission
 2530 between physically separated parts of the TOE, the TSF could provide a pre-defined
 2531 role with the ability to select the method that will be used.

2532 **10.10.4 Management of FDP_ITT.3, FDP_ITT.4**

2533 The following actions could be considered for the management functions in FMT:

2534 a) The specification of the actions to be taken upon detection of an integrity error
 2535 **could** be configurable.

2536 **10.10.5 Audit of FDP_ITT.1, FDP_ITT.2**

2537 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 2538 in the PP/ST:

2539 a) Minimal: Successful transfers of user data, including identification of the protection
 2540 method used.

2541 b) Basic: All attempts to transfer user data, including the protection method used and
 2542 any errors that occurred.

2543 **10.10.6 Audit of FDP_ITT.3, FDP_ITT.4**

2544 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 2545 in the PP/ST:

2546 a) Minimal: Successful transfers of user data, including identification of the integrity
 2547 protection method used.

2548 b) Basic: All attempts to transfer user data, including the integrity protection method
 2549 used and any errors that occurred.

2550 c) Basic: Unauthorized attempts to change the integrity protection method.

2551 d) Detailed: The action taken upon detection of an integrity error.

2552 **10.10.7 FDP_ITT.1 Basic internal transfer protection**

2553 **Component relationships**

2554 Hierarchical to: No other components.

2555 Dependencies: [FDP_ACC.1 Subset access control, or
 2556 FDP_IFC.1 Subset information flow control]

2557 **FDP_ITT.1.1**

2558 **The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow*
 2559 *control SFP(s)] to prevent the [selection: *disclosure, modification, loss of use*] of user data
 2560 when it is transmitted between physically-separated parts of the TOE.***

2561 **10.10.8 FDP_ITT.2 Transmission separation by attribute**

2562 **Component relationships**

2563 Hierarchical to: FDP_ITT.1 Basic internal transfer protection

2564 Dependencies: [FDP_ACC.1 Subset access control, or
 2565 FDP_IFC.1 Subset information flow control]

2566 **FDP_ITT.2.1**

2567 The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow control*
 2568 *SFP(s)] to prevent the [selection: *disclosure, modification, loss of use*] of user data when it is
 2569 transmitted between physically-separated parts of the TOE.*

2570 **FDP_ITT.2.2**

2571 **The TSF **shall** separate data controlled by the SFP(s) when transmitted between
 2572 physically-separated parts of the TOE, based on the values of the following: [assignment:
 2573 *security attributes that require separation*].**

2574 **10.10.9 FDP_ITT.3 Integrity monitoring**2575 **Component relationships**

2576	Hierarchical to:	No other components.
2577	Dependencies:	[FDP_ACC.1 Subset access control, or
2578		FDP_IFC.1 Subset information flow control]
2579		FDP_ITT.1 Basic internal transfer protection

2580 **FDP_ITT.3.1**

2581 The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow*
2582 *control SFP(s)*] to monitor user data transmitted between physically-separated parts of
2583 the TOE for the following errors: [assignment: *integrity errors*].

2584 **FDP_ITT.3.2**

2585 Upon detection of a data integrity error, the TSF **shall** [assignment: *specify the action to*
2586 *be taken upon integrity error*].

2587 **10.10.10 FDP_ITT.4 Attribute-based integrity monitoring**2588 **Component relationships**

2589	Hierarchical to:	FDP_ITT.3 Integrity monitoring
2590	Dependencies:	[FDP_ACC.1 Subset access control, or
2591		FDP_IFC.1 Subset information flow control]
2592		FDP_ITT.2 Transmission separation by attribute

2593 **FDP_ITT.4.1**

2594 The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow control*
2595 *SFP(s)*] to monitor user data transmitted between physically-separated parts of the TOE for the
2596 following errors: [assignment: *integrity errors*], **based on the following attributes:**
2597 **[assignment: *security attributes that require separate transmission channels*].**

2598 **FDP_ITT.4.2**

2599 Upon detection of a data integrity error, the TSF **shall** [assignment: *specify the action to be taken*
2600 *upon integrity error*].

2601 **10.11 Residual information protection (FDP_RIP)**2602 **10.11.1 Family behaviour**

2603 This family addresses the need to ensure that any data contained in a resource is not available
2604 when the resource is de-allocated from one object and reallocated to a different object. This
2605 family requires protection for any data contained in a resource that has been logically deleted
2606 or released but **may** still be present within the TSF-controlled resource which in turn **may** be re-
2607 allocated to another object.

2608 **10.11.2 Components leveling and description**

2609 Figure 32 shows the component leveling for this family.



Figure 32 — FDP_RIP: Component leveling

2610

2611 FDP_RIP.1 Subset residual information protection, requires that the TSF ensure that any
 2612 residual information content of any resources is unavailable to a defined subset of the objects
 2613 controlled by the TSF upon the resource's allocation or deallocation.

2614 FDP_RIP.2 Full residual information protection, requires that the TSF ensure that any residual
 2615 information content of any resources is unavailable to all objects upon the resource's allocation
 2616 or deallocation.

2617 **10.11.3 Management of FDP_RIP.1, FDP_RIP.2**

2618 The following actions **could** be considered for the management functions in FMT:

2619 a) The choice of when to perform residual information protection (i.e. upon allocation
 2620 or deallocation) **could** be made configurable within the TOE.

2621 **10.11.4 Audit of FDP_RIP.1, FDP_RIP.2**

2622 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 2623 in the PP/ST:

2624 a) There are no auditable events foreseen.

2625 **10.11.5 FDP_RIP.1 Subset residual information protection**

2626 **Component relationships**

2627 Hierarchical to: No other components.

2628 Dependencies: No dependencies.

2629 **FDP_RIP.1.1**

2630 **The TSF shall ensure that any previous information content of a resource is made**
 2631 **unavailable upon the [selection: *allocation of the resource to, deallocation of the resource***
 2632 **from] the following objects: [assignment: *list of objects*].**

2633 **10.11.6 FDP_RIP.2 Full residual information protection**

2634 **Component relationships**

2635 Hierarchical to: FDP_RIP.1 Subset residual information protection

2636 Dependencies: No dependencies.

2637 **FDP_RIP.2.1**

2638 The TSF **shall** ensure that any previous information content of a resource is made unavailable
 2639 upon the [selection: *allocation of the resource to, deallocation of the resource from*] **all** objects.

2640 **10.12 Rollback (FDP_ROL)**

2641 **10.12.1 Family behaviour**

2642 The rollback operation involves undoing the last operation or a series of operations, bounded
 2643 by some limit, such as a period of time, and return to a previous known state. Rollback provides
 2644 the ability to undo the effects of an operation or series of operations to preserve the integrity of
 2645 the user data.

2646 **10.12.2 Components leveling and description**

2647 Figure 33 shows the component leveling for this family.

2648 **Figure 33 — FDP_ROL: Component leveling**2649 FDP_ROL.1 Basic rollback addresses a need to roll back or undo a limited number of operations
2650 within the defined bounds.2651 FDP_ROL.2 Advanced rollback addresses the need to roll back or undo all operations within the
2652 defined bounds.2653 **10.12.3 Management of FDP_ROL.1, FDP_ROL.2**2654 The following actions **could** be considered for the management functions in FMT:

- 2655 a) The boundary limit to which rollback **may** be performed **could** be a configurable
2656 item within the TOE.
- 2657 b) Permission to perform a rollback operation **could** be restricted to a well-defined
2658 role.

2659 **10.12.4 Audit of FDP_ROL.1, FDP_ROL.2**2660 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2661 in the PP/ST:

- 2662 a) Minimal: All successful rollback operations.
- 2663 b) Basic: All attempts to perform rollback operations.
- 2664 c) Detailed: All attempts to perform rollback operations, including identification of the
2665 types of operations rolled back.

2666 **10.12.5 FDP_ROL.1 Basic rollback**2667 **Component relationships**

2668	Hierarchical to:	No other components.
2669	Dependencies:	[FDP_ACC.1 Subset access control, or
2670		FDP_IFC.1 Subset information flow control]

2671 **FDP_ROL.1.1**

2672 **The TSF **shall** enforce [assignment: *access control SFP(s) and/or information flow control***
2673 ***SFP(s)*] to permit the rollback of the [assignment: *list of operations*] on the [assignment:**
2674 ***information and/or list of objects*].**

2675 **FDP_ROL.1.2**

2676 **The TSF **shall** permit operations to be rolled back within the [assignment: *boundary limit***
2677 ***to which rollback **may** be performed*].**

2678 **10.12.6 FDP_ROL.2 Advanced rollback**2679 **Component relationships**

2680	Hierarchical to:	FDP_ROL.1 Basic rollback
------	------------------	--------------------------

2681 Dependencies: [FDP_ACC.1 Subset access control, or
 2682 FDP_IFC.1 Subset information flow control]

2683 **FDP_ROL.2.1**

2684 The TSF **shall** enforce [assignment: *access control SFP(s) and/or information flow control SFP(s)*]
 2685 to permit the rollback of **all** the **operations** on the [assignment: *list of objects*].

2686 **FDP_ROL.2.2**

2687 The TSF **shall** permit operations to be rolled back within the [assignment: *boundary limit to*
 2688 *which rollback may be performed*].

2689 **10.13 Stored data confidentiality (FDP_SDC)**

2690 **10.13.1 Family behaviour**

2691 This family provides requirements that address protection of user data confidentiality while the
 2692 data is stored within memory areas protected by the TSF. The TSF provides access to the data in
 2693 the memory through the specified interfaces only and prevents compromise of their
 2694 information bypassing these interfaces. It complements the family Stored data integrity
 2695 (FDP_SDI) which protects the user data from integrity errors while being stored in the memory.

2696 **10.13.2 Components leveling and description**

2697 Figure 34 shows the component leveling for this family.



2698 **Figure 34 — FDP_SDC: Component leveling**

2699 FDP_SDC.1 Stored data confidentiality, requires the TSF to protect the confidentiality of
 2700 information of the user data in specified memory areas.

2701 FDP_SDC.2 Stored data confidentiality with dedicated method, requires the TSF to protect the
 2702 confidentiality of the user data according to data characteristics leading to specify a dedicated
 2703 method of protection of confidentiality.

2704 **10.13.3 Management of FDP_SDC.1, FDP_SDC.2**

2705 The following actions **could** be considered for the management functions in FMT:

- 2706 a) No specific management functions are identified

2707 **10.13.4 Audit of FDP_SDC.1, FDP_SDC.2**

2708 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 2709 in the PP/ST:

- 2710 a) There are no auditable events foreseen.

2711 **10.13.5 FDP_SDC.1 Stored data confidentiality**

2712 **Component relationships**

2713 Hierarchical to: No other components.

2714 Dependencies: No dependencies.

2715 **FDP_SDC.1.1**

2716 **The TSF shall ensure the confidentiality of [selection: *all user data, the following user data***
 2717 **[assignment: *list of user data*]] while it is stored in the [selection: *temporary memory,***
 2718 ***persistent memory, any memory*].**

2719 **10.13.6 FDP_SDC.2 Stored data confidentiality with dedicated method**2720 **Component relationships**

2721 Hierarchical to: No other components.

2722 Dependencies: FCS_COP.1.

2723 **FDP_SDC.2.1**

2724 **The TSF shall ensure the confidentiality of the [selection: *all user data, the following user***
 2725 ***data [assignment: *list of user data*]] according to [assignment: *data characteristics*] while it***
 2726 **is stored under the control of the TSF.**

2727 **FDP_SDC.2.2**

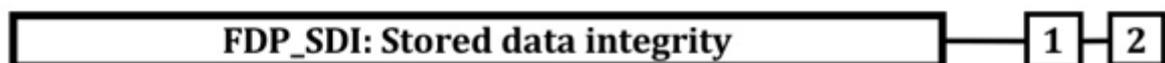
2728 **The TSF shall ensure the confidentiality of the user data specified in FDP_SDC.2.1 without**
 2729 **user intervention.**

2730 **10.14 Stored data integrity (FDP_SDI)**2731 **10.14.1 Family behaviour**

2732 This family provides requirements that address protection of user data while it is stored within
 2733 containers controlled by the TSF. Integrity errors **may** affect user data stored in memory, or in a
 2734 storage device. This family differs from Internal TOE transfer (FDP_ITT) which protects the user
 2735 data from integrity errors while being transferred within the TOE.

2736 **10.14.2 Components leveling and description**

2737 Figure 35 shows the component leveling for this family.



2738 **Figure 35 — FDP_SDI: Component leveling**

2739 FDP_SDI.1 Stored data integrity monitoring, requires that the TSF monitor user data stored
 2740 within containers controlled by the TSF for identified integrity errors.

2741 FDP_SDI.2 Stored data integrity monitoring and action adds the additional capability to the first
 2742 component by allowing for actions to be taken as a result of an error detection.

2743 **10.14.3 Management of FDP_SDI.1**

2744 The following actions **could** be considered for the management functions in FMT:

2745 a) There are no management activities foreseen.

2746 **10.14.4 Management of FDP_SDI.2**

2747 The following actions **could** be considered for the management functions in FMT:

2748 a) The actions to be taken upon the detection of an integrity error **could** be
 2749 configurable.

2750 **10.14.5 Audit of FDP_SDI.1**

2751 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2752 in the PP/ST:

- 2753 a) Minimal: Successful attempts to check the integrity of user data, including an
2754 indication of the results of the check.
- 2755 b) Basic: All attempts to check the integrity of user data, including an indication of the
2756 results of the check, if performed.
- 2757 c) Detailed: The type of integrity error that occurred.

2758 **10.14.6 Audit of FDP_SDI.2**

2759 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2760 in the PP/ST:

- 2761 a) Minimal: Successful attempts to check the integrity of user data, including an
2762 indication of the results of the check.
- 2763 b) Basic: All attempts to check the integrity of user data, including an indication of the
2764 results of the check, if performed.
- 2765 c) Detailed: The type of integrity error that occurred.
- 2766 d) Detailed: The action taken upon detection of an integrity error.

2767 **10.14.7 FDP_SDI.1 Stored data integrity monitoring**

2768 **Component relationships**

- 2769 Hierarchical to: No other components.
- 2770 Dependencies: No dependencies.

2771 **FDP_SDI.1.1**

2772 **The TSF **shall** monitor user data stored in containers controlled by the TSF for**
2773 **[assignment: *integrity errors*] on all objects, based on the following attributes:**
2774 **[assignment: *user data attributes*].**

2775 **10.14.8 FDP_SDI.2 Stored data integrity monitoring and action**

- 2776 Hierarchical to: FDP_SDI.1 Stored data integrity monitoring
- 2777 Dependencies: No dependencies.

2778 **FDP_SDI.2.1**

2779 The TSF **shall** monitor user data stored in containers controlled by the TSF for [assignment:
2780 *integrity errors*] on all objects, based on the following attributes: [assignment: *user data*
2781 *attributes*].

2782 **FDP_SDI.2.2**

2783 **Upon detection of a data integrity error, the TSF **shall** [assignment: *action to be taken*].**

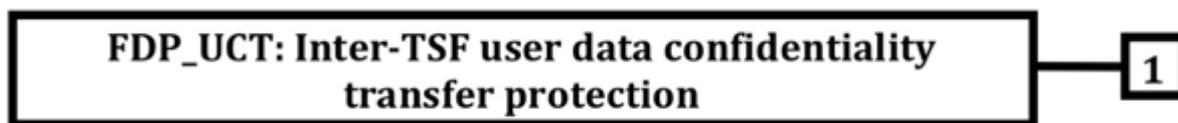
2784 **10.15 Inter-TSF user data confidentiality transfer protection (FDP_UCT)**

2785 **10.15.1 Family behaviour**

2786 This family defines the requirements for ensuring the confidentiality of user data when it is
2787 transferred using an external channel between the TOE and another trusted IT product.

2788 **10.15.2 Components leveling and description**

2789 Figure 36 shows the component leveling for this family.



2790

2791 **Figure 36 — FDP_UCT: Component leveling**

2792 In FDP_UCT.1 Basic data exchange confidentiality, the goal is to provide protection from
 2793 disclosure of user data while in transit.

2794 **10.15.3 Management of FDP_UCT.1**2795 The following actions **could** be considered for the management functions in FMT:

2796 a) There are no management activities foreseen.

2797 **10.15.4 Audit of FDP_UCT.1**

2798 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 2799 in the PP/ST:

2800 a) Minimal: The identity of any user or subject using the data exchange mechanisms.

2801 b) Basic: The identity of any unauthorized user or subject attempting to use the data
 2802 exchange mechanisms.

2803 c) Basic: A reference to the names or other indexing information useful in identifying
 2804 the user data that was transmitted or received. This **could** include security
 2805 attributes associated with the information.

2806 **10.15.5 FDP_UCT.1 Basic data exchange confidentiality**2807 **Component relationships**

2808 Hierarchical to: No other components.

2809 Dependencies: [FTP_ITC.1 Inter-TSF trusted channel, or
 2810 FTP_TRP.1 Trusted path]

2811 [FDP_ACC.1 Subset access control, or
 2812 FDP_IFC.1 Subset information flow control]

2813 **FDP_UCT.1.1**

2814 **The TSF shall enforce the [assignment: access control SFP(s) and/or information flow**
 2815 **control SFP(s)] to [selection: transmit, receive] user data in a manner protected from**
 2816 **unauthorized disclosure.**

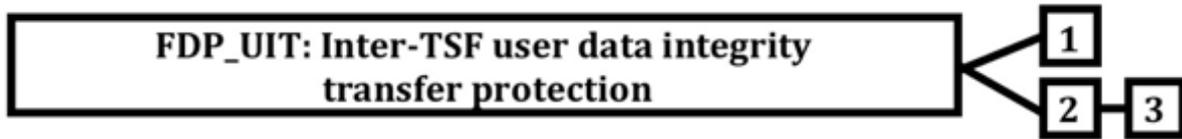
2817 **10.16 Inter-TSF user data integrity transfer protection (FDP_UIT)**2818 **10.16.1 Family behaviour**

2819 This family defines the requirements for providing integrity for user data in transit between the
 2820 TOE and another trusted IT product and recovering from detectable errors. At a minimum, this
 2821 family monitors the integrity of user data for modifications. Furthermore, this family supports
 2822 different ways of correcting detected integrity errors.

2823 **10.16.2 Components leveling and description**

2824 Figure 37 shows the component leveling for this family.

2825



2826 **Figure 37 — FDP_UIT: Component leveling**

2827 FDP_UIT.1 Data exchange integrity addresses detection of modifications, deletions, insertions,
2828 and replay errors of the user data transmitted.

2829 FDP_UIT.2 Source data exchange recovery addresses recovery of the original user data by the
2830 receiving TSF with help from the source trusted IT product.

2831 FDP_UIT.3 Destination data exchange recovery addresses recovery of the original user data by
2832 the receiving TSF on its own without any help from the source trusted IT product.

2833 **10.16.3 Management of FDP_UIT.1, FDP_UIT.2, FDP_UIT.3**

2834 The following actions **could** be considered for the management functions in FMT:

- 2835 a) There are no management activities foreseen.

2836 **10.16.4 Audit of FDP_UIT.1**

2837 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2838 in the PP/ST:

- 2839 a) Minimal: The identity of any user or subject using the data exchange mechanisms.
2840 b) Basic: The identity of any user or subject attempting to use the user data exchange
2841 mechanisms, but who is unauthorized to do so.
2842 c) Basic: A reference to the names or other indexing information useful in identifying
2843 the user data that was transmitted or received. This **could** include security
2844 attributes associated with the user data.
2845 d) Basic: Any identified attempts to block transmission of user data.
2846 e) Detailed: The types and/or effects of any detected modifications of transmitted
2847 user data.

2848 **10.16.5 Audit of FDP_UIT.2, FDP_UIT.3**

2849 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2850 in the PP/ST:

- 2851 a) Minimal: The identity of any user or subject using the data exchange mechanisms;
2852 b) Minimal: Successful recovery from errors including the type of error that was
2853 detected;
2854 c) Basic: The identity of any user or subject attempting to use the user data exchange
2855 mechanisms, but who is unauthorized to do so;
2856 d) Basic: A reference to the names or other indexing information useful in identifying
2857 the user data that was transmitted or received. This **could** include security
2858 attributes associated with the user data;
2859 e) Basic: Any identified attempts to block transmission of user data;

2860 f) Detailed: The types and/or effects of any detected modifications of transmitted
2861 user data.

2862 10.16.6 FDP_UIT.1 Data exchange integrity

2863 Component relationships

2864 Hierarchical to: No other components.
2865 Dependencies: [FDP_ACC.1 Subset access control, or
2866 FDP_IFC.1 Subset information flow control]
2867 [FTP_ITC.1 Inter-TSF trusted channel, or
2868 FTP_TRP.1 Trusted path]

2869 FDP_UIT.1.1

2870 The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow*
2871 *control SFP(s)*] to [selection: *transmit, receive*] user data in a manner protected from
2872 [selection: *modification, deletion, insertion, replay*] errors.

2873 FDP_UIT.1.2

2874 The TSF **shall** be able to determine on receipt of user data, whether [selection:
2875 *modification, deletion, insertion, replay*] has occurred.

2876 10.16.7 FDP_UIT.2 Source data exchange recovery

2877 Component relationships

2878 Hierarchical to: No other components.
2879 Dependencies: [FDP_ACC.1 Subset access control, or
2880 FDP_IFC.1 Subset information flow control]
2881 [FDP_UIT.1 Data exchange integrity, or
2882 FTP_ITC.1 Inter-TSF trusted channel]

2883 FDP_UIT.2.1

2884 The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow*
2885 *control SFP(s)*] to be able to recover from [assignment: *list of recoverable errors*] with the
2886 help of the source trusted IT product.

2887 10.16.8 FDP_UIT.3 Destination data exchange recovery

2888 Hierarchical to: FDP_UIT.2 Source data exchange recovery
2889 Dependencies: [FDP_ACC.1 Subset access control, or
2890 FDP_IFC.1 Subset information flow control]
2891 [FDP_UIT.1 Data exchange integrity, or
2892 FTP_ITC.1 Inter-TSF trusted channel]

2893 FDP_UIT.3.1

2894 The TSF **shall** enforce the [assignment: *access control SFP(s) and/or information flow*
2895 *control SFP(s)*] to be able to recover from [assignment: *list of recoverable errors*] **without any** help
2896 **from** the source trusted IT product.

2897

2898 **11 Class FIA: Identification and authentication**

2899 **11.1 Class description**

2900 Families in this class address the requirements for functions to establish and verify a claimed
 2901 user identity.

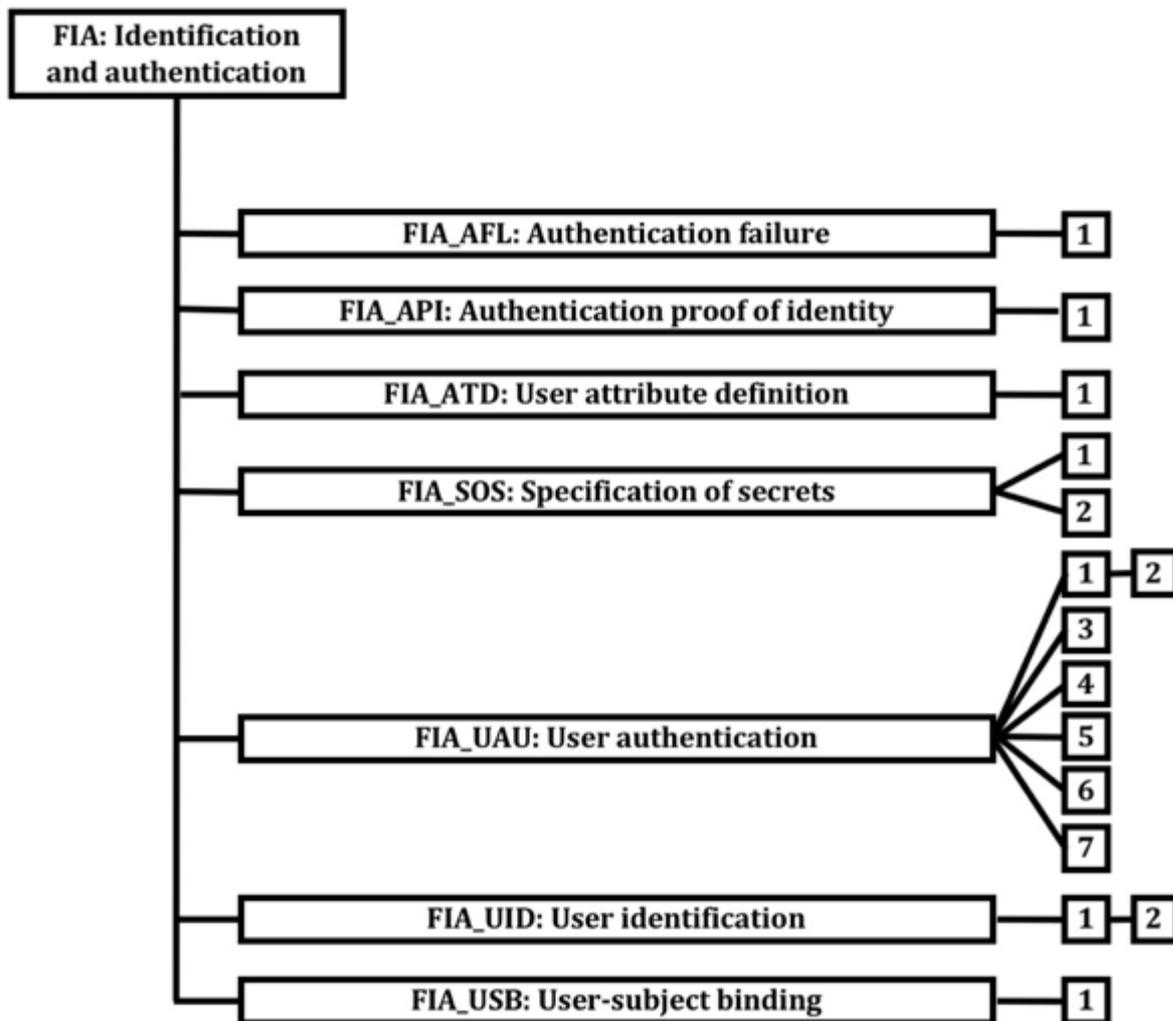
2902 Identification and authentication is required to ensure that users are associated with the proper
 2903 security attributes

2904 The unambiguous identification of authorized users and the correct association of security
 2905 attributes with users and subjects is critical to the enforcement of the intended security
 2906 policies. The families in this class deal with determining and verifying the identity of users,
 2907 determining their authority to interact with the TOE, and with the correct association of
 2908 security attributes for each authorized user. Other classes of requirements are dependent upon
 2909 correct identification and authentication of users in order to be effective.

2910 Figure 38 shows the decomposition of this class, it's families and components. Elements are not
 2911 shown in the figure.

2912 Annex G provides explanatory information for this class and **should** be consulted when using
 2913 the components identified in this class.

2914



2915 **Figure 38 — FIA: Identification and authentication class decomposition**

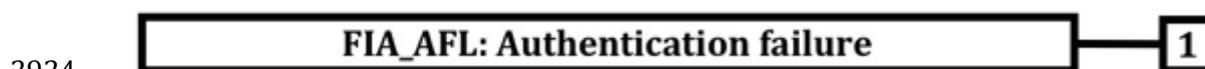
2916 11.2 Authentication failures (FIA_AFL)

2917 11.2.1 Family behaviour

2918 This family contains requirements for defining values for some number of unsuccessful
2919 authentication attempts and TSF actions in cases of authentication attempt failures. Parameters
2920 include, but are not limited to, the number of failed authentication attempts and time
2921 thresholds.

2922 11.2.2 Components leveling and description

2923 Figure 39 shows the component leveling for this family.



2925 **Figure 39 — FIA_AFL: Component leveling**

2926 FIA_AFL.1 Authentication failure handling, requires that the TSF be able to terminate the
2927 session establishment process after a specified number of unsuccessful user authentication
2928 attempts. It also requires that, after termination of the session establishment process, the TSF
2929 be able to disable the user account or the point of entry from which the attempts were made
2930 until an administrator-defined condition occurs.

2931 11.2.3 Management of FIA_AFL.1

2932 The following actions **could** be considered for the management functions in FMT:

- 2933 a) Management of the threshold for unsuccessful authentication attempts;
- 2934 b) Management of actions to be taken in the event of an authentication failure.

2935 11.2.4 Audit of FIA_AFL.1

2936 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2937 in the PP/ST:

- 2938 a) Minimal: the reaching of the threshold for the unsuccessful authentication attempts
2939 and the actions taken and the subsequent, if appropriate, restoration to the normal
2940 state.

2941 11.2.5 FIA_AFL.1 Authentication failure handling

2942 Component relationships

2943	Hierarchical to:	No other components.
2944	Dependencies:	FIA_UAU.1 Timing of authentication

2945 FIA_AFL.1.1

2946 **The TSF shall detect when [selection: [assignment: *positive integer number*], an
2947 administrator configurable positive integer within [assignment: *range of acceptable
2948 values*]] unsuccessful authentication attempts occur related to [assignment: *list of
2949 authentication events*].**

2950 FIA_AFL.1.2

2951 **When the defined number of unsuccessful authentication attempts has been [selection:
2952 *met, surpassed*], the TSF shall [assignment: *list of actions*].**

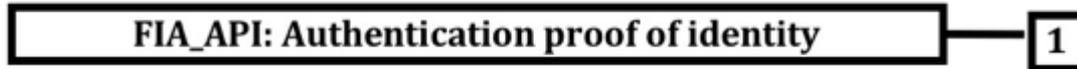
2953 **11.3 Authentication proof of identity (FIA_API)**

2954 **11.3.1 Family behaviour**

2955 This family defines functions provided by the TOE to prove its identity and so allow for
2956 verification of the TOE by an external entity in the TOE's IT environment.

2957 **11.3.2 Components leveling and description**

2958 Figure 40 shows the component leveling for this family.



2959 **Figure 40 — FIA_API: Component leveling**

2960 FIA_API.1 Authentication Proof of Identity, provides proof of the identity of the TOE to an
2961 external entity.

2962 **11.3.3 Management of FIA_API.1**

2963 The following actions **could** be considered for the management functions in FMT:

- 2964 a) Management of authentication information used to prove the claimed identity.

2965 **11.3.4 Audit of FIA_API.1**

2966 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2967 in the PP/ST:

- 2968 a) There are no auditable events foreseen.

2969 **11.3.5 FIA_API.1 Authentication proof of identity**

2970 **Component relationships**

2971 Hierarchical to: No other components.

2972 Dependencies: No dependencies.

2973 **FIA_API.1.1**

2974 The TSF **shall** provide an [assignment: *authentication mechanism*] to prove the identity of
2975 the [assignment: *object, authorized user, or role*] to an external entity.

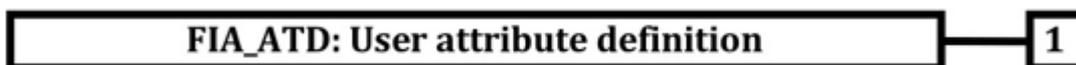
2976 **11.4 User attribute definition (FIA_ATD)**

2977 **11.4.1 Family behaviour**

2978 All authorized users **may** have a set of security attributes, other than the user's identity, that is
2979 used to enforce the SFRs. This family defines the requirements for associating user security
2980 attributes with users as needed to support the TSF in making security decisions.

2981 **11.4.2 Components leveling and description**

2982 Figure 41 shows the component leveling for this family.



2983 **Figure 41 — FIA_ATD: Component leveling**

2984 FIA_ATD.1 User attribute definition, allows user security attributes for each user to be
2985 maintained individually.

2986 11.4.3 Management of FIA_ATD.1

2987 The following actions **could** be considered for the management functions in FMT:

2988 a) if so indicated in the assignment, the authorized administrator might be able to
2989 define additional security attributes for users.

2990 11.4.4 Audit of FIA_ATD.1

2991 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
2992 in the PP/ST:

2993 a) There are no auditable events foreseen.

2994 11.4.5 FIA_ATD.1 User attribute definition

2995 Component relationships

2996 Hierarchical to: No other components.

2997 Dependencies: No dependencies.

2998 FIA_ATD.1.1

2999 The TSF **shall** maintain the following list of security attributes belonging to individual
3000 users: [assignment: *list of security attributes*].

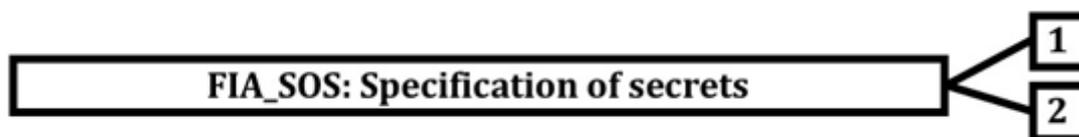
3001 11.5 Specification of secrets (FIA_SOS)

3002 11.5.1 Family behaviour

3003 This family defines requirements for mechanisms that enforce defined quality metrics on
3004 provided secrets and generate secrets to satisfy the defined metric.

3005 11.5.2 Components leveling and description

3006 Figure 42 shows the component leveling for this family.



3007 **Figure 42 — FIA_SOS: Component leveling**

3008 FIA_SOS.1 Verification of secrets, requires the TSF to verify that secrets meet defined quality
3009 metrics.

3010 FIA_SOS.2 TSF Generation of secrets, requires the TSF to be able to generate secrets that meet
3011 defined quality metrics.

3012 11.5.3 Management of FIA_SOS.1

3013 The following actions **could** be considered for the management functions in FMT:

3014 a) the management of the metric used to verify the secrets.

3015 11.5.4 Management of FIA_SOS.2

3016 The following actions **could** be considered for the management functions in FMT:

3017 a) the management of the metric used to generate the secrets.

3018 **11.5.5 Audit of FIA_SOS.1, FIA_SOS.2**

3019 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3020 in the PP/ST:

3021 a) Minimal: Rejection by the TSF of any tested secret;

3022 b) Basic: Rejection or acceptance by the TSF of any tested secret;

3023 c) Detailed: Identification of any changes to the defined quality metrics.

3024 **11.5.6 FIA_SOS.1 Verification of secrets**

3025 **Component relationships**

3026 Hierarchical to: No other components.

3027 Dependencies: No dependencies.

3028 **FIA_SOS.1.1**

3029 **The TSF shall provide a mechanism to verify that secrets meet [assignment: a defined**
3030 **quality metric].**

3031 **11.5.7 FIA_SOS.2 TSF Generation of secrets**

3032 **Component relationships**

3033 Hierarchical to: No other components.

3034 Dependencies: No dependencies.

3035 **FIA_SOS.2.1**

3036 The TSF **shall** provide a mechanism to **generate** secrets that meet [assignment: a defined
3037 *quality metric*].

3038 **FIA_SOS.2.2**

3039 **The TSF shall be able to enforce the use of TSF generated secrets for [assignment: list of**
3040 **TSF functions].**

3041 **11.6 User authentication (FIA_UAU)**

3042 **11.6.1 Family behaviour**

3043 This family defines the types of user authentication mechanisms supported by the TSF. This
3044 family also defines the required attributes on which the user authentication mechanisms must
3045 be based.

3046 **11.6.2 Components leveling and description**

3047 Figure 43 shows the component leveling for this family.

3048 **Figure 43 — FIA_UAU: Component leveling**3049 FIA_UAU.1 Timing of authentication, allows a user to perform certain actions prior to the
3050 authentication of the user's identity.3051 FIA_UAU.2 User authentication before any action, requires that users are authenticated before
3052 any other action will be allowed by the TSF.3053 FIA_UAU.3 Unforgeable authentication, requires the authentication mechanism to be able to
3054 detect and prevent the use of authentication data that has been forged or copied.3055 FIA_UAU.4 Single-use authentication mechanisms, requires an authentication mechanism that
3056 operates with single-use authentication data.3057 FIA_UAU.5 Multiple authentication mechanisms, requires that different authentication
3058 mechanisms be provided and used to authenticate user identities for specific events.3059 FIA_UAU.6 Re-authenticating, requires the ability to specify events for which the user needs to
3060 be re-authenticated.3061 FIA_UAU.7 Protected authentication feedback, requires that only limited feedback information
3062 is provided to the user during the authentication.3063 **11.6.3 Management of FIA_UAU.1**3064 The following actions **could** be considered for the management functions in FMT:

- 3065 a) management of the authentication data by an administrator;
- 3066 b) management of the authentication data by the associated user;
- 3067 c) managing the list of actions that **can** be taken before the user is authenticated.

3068 **11.6.4 Management of FIA_UAU.2**3069 The following actions **could** be considered for the management functions in FMT:

- 3070 a) management of the authentication data by an administrator;
- 3071 b) management of the authentication data by the user associated with this data.

3072 **11.6.5 Management of FIA_UAU.3, FIA_UAU.4, FIA_UAU.7**3073 The following actions **could** be considered for the management functions in FMT:

- 3074 a) There are no management activities foreseen.

3075 **11.6.6 Management of FIA_UAU.5**

3076 The following actions **could** be considered for the management functions in FMT:

- 3077 a) the management of authentication mechanisms;

3078 **11.6.7 Management of FIA_UAU.6**

3079 The following actions **could** be considered for the management functions in FMT:

- 3080 a) if an authorized administrator **could** request re-authentication, the management
3081 includes a re-authentication request.

3082 **11.6.8 Management of FIA_UAU.7**

3083 The following actions **could** be considered for the management functions in FMT:

- 3084 a) the management of the rules for authentication.

3085 **11.6.9 Audit of FIA_UAU.1**

3086 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3087 in the PP/ST:

- 3088 a) Minimal: Unsuccessful use of the authentication mechanism;
3089 b) Basic: All use of the authentication mechanism;
3090 c) Detailed: All TSF mediated actions performed before authentication of the user.

3091 **11.6.10 Audit of FIA_UAU.2**

3092 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3093 in the PP/ST:

- 3094 a) Minimal: Unsuccessful use of the authentication mechanism;
3095 b) Basic: All use of the authentication mechanism.

3096 **11.6.11 Audit of FIA_UAU.3**

3097 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3098 in the PP/ST:

- 3099 a) Minimal: Detection of fraudulent authentication data;
3100 b) Basic: All immediate measures taken and results of checks on the fraudulent data.

3101 **11.6.12 Audit of FIA_UAU.4**

3102 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3103 in the PP/ST:

- 3104 a) Minimal: Attempts to reuse authentication data.

3105 **11.6.13 Audit of FIA_UAU.5**

3106 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3107 in the PP/ST:

- 3108 a) Minimal: The final decision on authentication;
3109 b) Basic: The result of each activated mechanism together with the final decision.

3110 **11.6.14 Audit of FIA_UAU.6**

3111 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3112 in the PP/ST:

3113 a) Minimal: Failure of re-authentication;

3114 b) Basic: All re-authentication attempts.

3115 **11.6.15 Audit of FIA_UAU.7**

3116 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3117 in the PP/ST:

3118 a) Well-formedness of rules regarding the semantics of rule-set;

3119 b) Basic: verification of enforceability of rules (and their writing).

3120 **Editors' Note:**

3121 **b) should be changed to make it clearer.**

3122 **Comments are requested.**

3123 **11.6.16 FIA_UAU.1 Timing of authentication**

3124 **Component relationships**

3125 Hierarchical to: No other components.

3126 Dependencies: FIA_UID.1 Timing of identification

3127 **FIA_UAU.1.1**

3128 **The TSF shall allow [assignment: *list of TSF mediated actions*] on behalf of the user to be**
3129 **performed before the user is authenticated.**

3130 **FIA_UAU.1.2**

3131 **The TSF shall require each user to be successfully authenticated before allowing any**
3132 **other TSF-mediated actions on behalf of that user.**

3133 **11.6.17 FIA_UAU.2 User authentication before any action**

3134 **Component relationships**

3135 Hierarchical to: FIA_UAU.1 Timing of authentication

3136 Dependencies: FIA_UID.1 Timing of identification

3137 **FIA_UAU.2.1**

3138 The TSF **shall** require each user to be successfully authenticated before allowing any other TSF-
3139 mediated actions on behalf of that user.

3140 **11.6.18 FIA_UAU.3 Unforgeable authentication**

3141 **Component relationships**

3142 Hierarchical to: No other components.

3143 Dependencies: No dependencies.

3144 **FIA_UAU.3.1**

3145 **The TSF shall [selection: *detect, prevent*] use of authentication data that has been forged**
3146 **by any user of the TSF.**

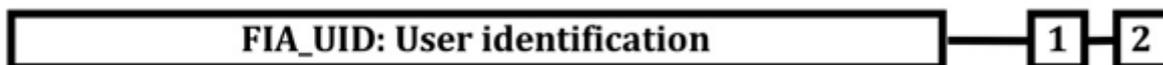
- 3147 **FIA_UAU.3.2**
- 3148 **The TSF shall [selection: *detect, prevent*] use of authentication data that has been copied**
 3149 **from any other user of the TSF.**
- 3150 **11.6.19 FIA_UAU.4 Single-use authentication mechanisms**
- 3151 **Component relationships**
- 3152 Hierarchical to: No other components.
- 3153 Dependencies: No dependencies.
- 3154 **FIA_UAU.4.1**
- 3155 **The TSF shall prevent reuse of authentication data related to [assignment: *identified***
 3156 ***authentication mechanism(s)*].**
- 3157 **11.6.20 FIA_UAU.5 Multiple authentication mechanisms**
- 3158 **Component relationships**
- 3159 Hierarchical to: No other components.
- 3160 Dependencies: No dependencies.
- 3161 **FIA_UAU.5.1**
- 3162 **The TSF shall provide [assignment: *list of multiple authentication mechanisms*] to support**
 3163 **user authentication.**
- 3164 **FIA_UAU.5.2**
- 3165 **The TSF shall authenticate any user's claimed identity according to the [assignment:**
 3166 ***rules describing how the multiple authentication mechanisms provide authentication*].**
- 3167 **11.6.21 FIA_UAU.6 Re-authenticating**
- 3168 **Component relationships**
- 3169 Hierarchical to: No other components.
- 3170 Dependencies: No dependencies.
- 3171 **FIA_UAU.6.1**
- 3172 **The TSF shall re-authenticate the user under the conditions [assignment: *list of***
 3173 ***conditions under which re-authentication is required*].**
- 3174 **11.6.22 FIA_UAU.7 Protected authentication feedback**
- 3175 **Component relationships**
- 3176 Hierarchical to: No other components.
- 3177 Dependencies: FIA_UID.1 Timing of identification
- 3178 **FIA_UAU.7.1**
- 3179 **The TSF shall provide only [assignment: *list of feedback*] to the user while the**
 3180 **authentication is in progress.**

3181 **11.7 User identification (FIA_UID)**3182 **11.7.1 Family behaviour**

3183 This family defines the conditions under which users **shall** be required to identify themselves
 3184 before performing any other actions that are to be mediated by the TSF and which require user
 3185 identification.

3186 **11.7.2 Components leveling and description**

3187 Figure 44 shows the component leveling for this family.



3188 **Figure 44 — FIA_UID: Component leveling**

3189 FIA_UID.1 Timing of identification, allows users to perform certain actions before being
 3190 identified by the TSF.

3191 FIA_UID.2 User identification before any action, requires that users identify themselves before
 3192 any action will be allowed by the TSF.

3193 **11.7.3 Management of FIA_UID.1**

3194 The following actions **could** be considered for the management functions in FMT:

- 3195 a) The management of the user identities;
- 3196 b) If an authorized administrator **can** change the actions allowed before identification,
 3197 the managing of the action lists.

3198 **11.7.4 Management of FIA_UID.2**

3199 The following actions **could** be considered for the management functions in FMT:

- 3200 a) The management of the user identities;

3201 **11.7.5 Audit of FIA_UID.1, FIA_UID.2**

3202 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 3203 in the PP/ST:

- 3204 a) Minimal: Unsuccessful use of the user identification mechanism, including the user
 3205 identity provided;
- 3206 b) Basic: All use of the user identification mechanism, including the user identity
 3207 provided.

3208 **11.7.6 FIA_UID.1 Timing of identification**3209 **Component relationships**

3210 Hierarchical to: No other components.

3211 Dependencies: No dependencies.

3212 **FIA_UID.1.1**

3213 **The TSF **shall** allow [assignment: *list of TSF-mediated actions*] on behalf of the user to be**
 3214 **performed before the user is identified.**

3215 **FIA_UID.1.2**

3216 **The TSF shall require each user to be successfully identified before allowing any TSF-**
 3217 **mediated actions on behalf of that user.**

3218 **11.7.7 FIA_UID.2 User identification before any action**

3219 Hierarchical to: FIA_UID.1 Timing of identification

3220 Dependencies: No dependencies.

3221 **FIA_UID.2.1**

3222 The TSF shall require each user to be successfully identified before allowing any TSF-mediated
 3223 actions on behalf of that user.

3224 **11.8 User-subject binding (FIA_USB)**

3225 **11.8.1 Family behaviour**

3226 An authenticated user, in order to use the TOE, typically activates a subject. The user's security
 3227 attributes are associated (totally or partially) with this subject. This family defines
 3228 requirements to create and maintain the association of the user's security attributes to a subject
 3229 acting on the user's behalf.

3230 **11.8.2 Components leveling and description**

3231 Figure 45 shows the component leveling for this family.



3232 **Figure 45 — FIA_USB: Component leveling**

3233 FIA_USB.1 User-subject binding, requires the specification of any rules governing the
 3234 association between user attributes and the subject attributes into which they are mapped.

3235 **11.8.3 Management of FIA_USB.1**

3236 The following actions could be considered for the management functions in FMT:

- 3237 a) An authorized administrator can define default subject security attributes;
- 3238 b) An authorized administrator can change subject security attributes.

3239 **11.8.4 Audit of FIA_USB.1**

3240 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3241 in the PP/ST:

- 3242 a) Minimal: Unsuccessful binding of user security attributes to a subject
- 3243 b) Basic: Success and failure of binding of user security attributes to a subject.

3244 **11.8.5 FIA_USB.1 User-subject binding**

3245 **Component relationships**

3246 Hierarchical to: No other components.

3247 Dependencies: FIA_ATD.1 User attribute definition

3248 **FIA_USB.1.1**

3249 **The TSF shall** associate the following user security attributes with subjects acting on the
3250 behalf of that user: [assignment: *list of user security attributes*].

3251 **FIA_USB.1.2**

3252 **The TSF shall** enforce the following rules on the initial association of user security
3253 attributes with subjects acting on the behalf of users: [assignment: *rules for the initial*
3254 *association of attributes*].

3255 **FIA_USB.1.3**

3256 **The TSF shall** enforce the following rules governing changes to the user security
3257 attributes associated with subjects acting on the behalf of users: [assignment: *rules for*
3258 *the changing of attributes*].

3259

3260 **12 Class FMT: Security management**

3261 **12.1 Class description**

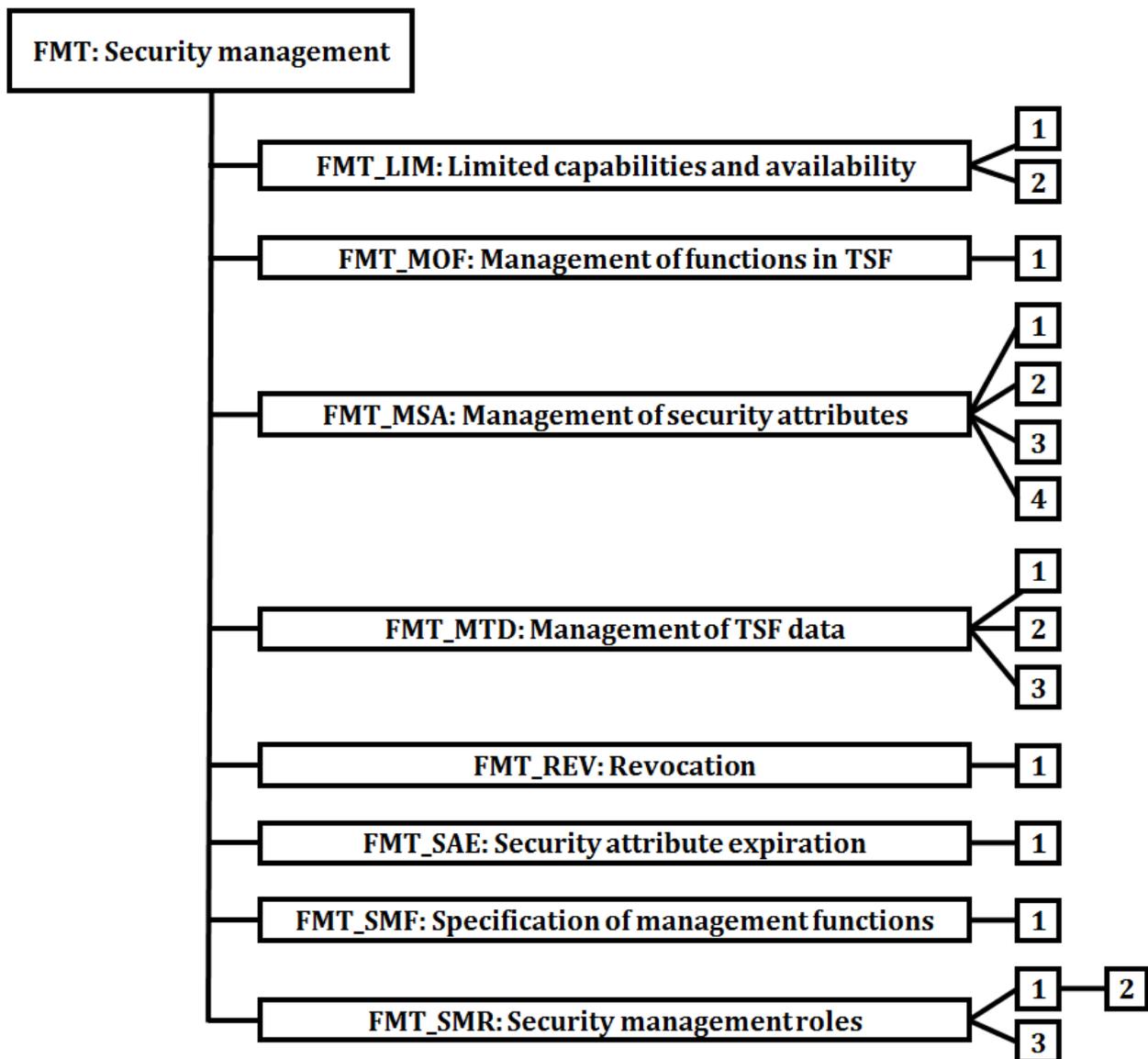
3262 This class is intended to specify the management of several aspects of the TSF: security
 3263 attributes, TSF data and functions. The different management roles and their interaction, such
 3264 as separation of capability, can be specified.

3265 This class has several objectives:

- 3266 a) Management of TSF data;
- 3267 b) Management of security attributes;
- 3268 c) Management of functions of the TSF;
- 3269 d) Definition of security roles.

3270 Figure 46 shows the decomposition of this class, it's families and components. Elements are not
 3271 shown in the figure.

3272 Annex H provides explanatory information for this class and should be consulted when using
 3273 the components identified in this class.



3274

3275 **Figure 46 — FMT: Security management class decomposition**

3276 **12.2 Limited capabilities and availability (FMT_LIM)**

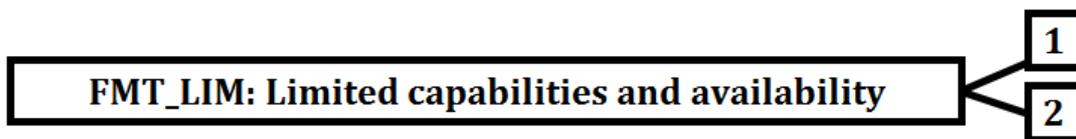
3277 **12.2.1 Family behaviour**

3278 This family defines requirements that limit the capabilities and availability of functions in a
3279 combined manner.

3280 Note FDP_ACF restricts the access to functions whereas the component Limited Capability of this family
3281 requires the functions themselves to be designed in a specific manner.

3282 **12.2.2 Components leveling and description**

3283 Figure 47 shows the component leveling for this family.



3284

3285 **Figure 47 — FMT_LIM: Component leveling**

3286 **Editors note:**

3287 **SMEs are asked to review the component leveling diagram and the relationships below.**

3288 **IF FMT_LIM.1 has a dependency on FMT_LIM.2 should the components be hierarchical?**

3289

3290 FMT_LIM.1 Limited capabilities requires that the TSF is built to provide only the capabilities
3291 (perform action, gather information) necessary for its genuine purpose.

3292 FMT_LIM.2 Limited availability requires that the TSF restrict the use of functions (refer to
3293 Limited capabilities (FMT_LIM.1)). This **can** be achieved, for instance, by removing or by
3294 disabling functions in a specific phase of the TOE's life-cycle.

3295 **12.2.3 Management of FMT_LIM.1, FMT_LIM.2**

3296 The following actions **could** be considered for the management functions in FMT:

3297 a) There are no management activities foreseen.

3298 **12.2.4 Audit of FMT_LIM.1**

3299 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3300 in the PP/ST:

3301 a) There are no auditable events foreseen.

3302 **12.2.5 FMT_LIM.1 Limited capabilities**

3303 **Component relationships**

3304 Hierarchical to: No other components.

3305 Dependencies: FMT_LIM.2 Limited availability

3306 **FMT_LIM.1.1**

3307 **The TSF shall limit its capabilities so that in conjunction with “Limited availability**
3308 **(FMT_LIM.2)” the following policy is enforced [assignment: Limited capability and**
3309 **availability policy].**

3310 **12.2.6 FMT_LIM.2 Limited availability**

3311 **Component relationships**

- 3312 Hierarchical to: No other components.
 3313 Dependencies: FMT_LIM.1 Limited capabilities

3314 **FMT_LIM.2.1**

3315 **The TSF shall be designed in a manner that limits its availability so that in conjunction**
 3316 **with “Limited capabilities (FMT_LIM.1)” the following policy is enforced [assignment:**
 3317 **Limited capability and availability policy].**

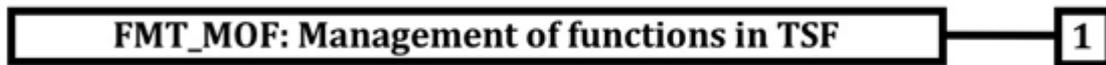
3318 **12.3 Management of functions in TSF (FMT_MOF)**

3319 **12.3.1 Family behaviour**

3320 This family allows authorized users to control over the management of functions in the TSF.

3321 **12.3.2 Components leveling and description**

3322 Figure 48 shows the component leveling for this family.



3323 **Figure 48 — FMT_MOF: Component leveling**

3324 FMT_MOF.1 Management of security functions behaviour allows the authorized users (roles) to
 3325 manage the behaviour of functions in the TSF that use rules or have specified conditions that
 3326 may be manageable.

3327 **12.3.3 Management of FMT_MOF.1**

3328 The following actions could be considered for the management functions in FMT:

- 3329 a) managing the group of roles that can interact with the functions in the TSF.

3330 **12.3.4 Audit of FMT_MOF.1**

3331 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3332 in the PP/ST:

- 3333 a) Basic: All modifications in the behaviour of the functions in the TSF.

3334 **12.3.5 FMT_MOF.1 Management of security functions behaviour**

3335 **Component relationships**

- 3336 Hierarchical to: No other components.
 3337 Dependencies: FMT_SMR.1 Security roles
 3338 FMT_SMF.1 Specification of Management Functions

3339 **FMT_MOF.1.1**

3340 **The TSF shall restrict the ability to [selection: determine the behaviour of, disable, enable,**
 3341 **modify the behaviour of] the functions [assignment: list of functions] to [assignment: the**
 3342 **authorized identified roles].**

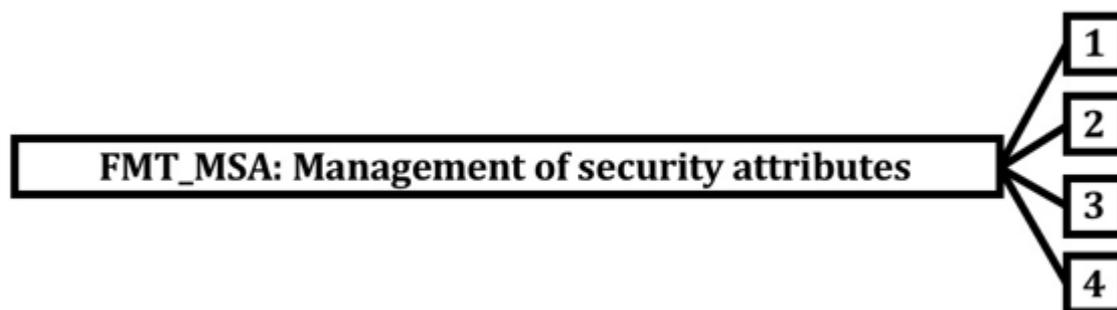
3343 **12.4 Management of security attributes (FMT_MSA)**

3344 **12.4.1 Family behaviour**

3345 This family allows authorized users control over the management of security attributes. This
3346 management might include capabilities for viewing and modifying of security attributes.

3347 **12.4.2 Components leveling and description**

3348 Figure 49 shows the component leveling for this family.



3349 **Figure 49 — FMT_MSA: Component leveling**

3350 FMT_MSA.1 Management of security attributes allows authorized users (roles) to manage the
3351 specified security attributes.

3352 FMT_MSA.2 Secure security attributes ensures that values assigned to security attributes are
3353 valid with respect to the secure state.

3354 FMT_MSA.3 Static attribute ensures that the default values of security attributes are
3355 appropriately either permissive or restrictive in nature.

3356 FMT_MSA.4 Security attribute value inheritance allows the rules/policies to be specified that
3357 will dictate the value to be inherited by a security attribute.

3358 **12.4.3 Management of FMT_MSA.1**

3359 The following actions **could** be considered for the management functions in FMT:

- 3360 a) Managing the group of roles that **can** interact with the security attributes;
- 3361 b) Management of rules by which security attributes inherit specified values.

3362 **12.4.4 Management of FMT_MSA.2**

3363 The following actions **could** be considered for the management functions in FMT:

- 3364 a) Management of rules by which security attributes inherit specified values.

3365 **12.4.5 Management of FMT_MSA.3**

3366 The following actions **could** be considered for the management functions in FMT:

- 3367 a) Managing the group of roles that **can** specify initial values;
- 3368 b) Managing the permissive or restrictive setting of default values for a given access
3369 control SFP;
- 3370 c) Management of rules by which security attributes inherit specified values.

3371 **12.4.6 Management of FMT_MSA.4**

3372 The following actions **could** be considered for the management functions in FMT:

3373 a) Specification of the role permitted to establish or modify security attributes.

3374 **12.4.7 Audit of FMT_MSA.1**

3375 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3376 in the PP/ST:

3377 a) Basic: All modifications of the values of security attributes.

3378 **12.4.8 Audit of FMT_MSA.2**

3379 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3380 in the PP/ST:

3381 a) Minimal: All offered and rejected values for a security attribute.

3382 b) Detailed: All offered and accepted secure values for a security attribute.

3383 **12.4.9 Audit of FMT_MSA.3**

3384 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3385 in the PP/ST:

3386 a) Basic: Modifications of the default setting of permissive or restrictive rules.

3387 b) Basic: All modifications of the initial values of security attributes.

3388 **12.4.10 Audit of FMT_MSA.4**

3389 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3390 in the PP/ST:

3391 a) Basic: Modifications of security attributes, possibly with the old and/or values of
3392 security attributes that were modified.

3393 **12.4.11 FMT_MSA.1 Management of security attributes**

3394 **Component relationships**

3395 Hierarchical to: No other components.

3396 Dependencies: [FDP_ACC.1 Subset access control, or
3397 FDP_IFC.1 Subset information flow control]

3398 FMT_SMR.1 Security roles

3399 FMT_SMF.1 Specification of Management Functions

3400 **FMT_MSA.1.1**

3401 **The TSF **shall** enforce the [assignment: *access control SFP(s), information flow control***
3402 ***SFP(s)*] to restrict the ability to [selection: *change_default, query, modify, delete,***
3403 ***[assignment: other operations]] the security attributes [assignment: *list of security****
3404 ***attributes*] to [assignment: *the authorized identified roles*].**

3405 **12.4.12 FMT_MSA.2 Secure security attributes**

3406 **Component relationships**

3407 Hierarchical to: No other components.

3408 Dependencies: [FDP_ACC.1 Subset access control, or
3409 FDP_IFC.1 Subset information flow control]

3410 FMT_MSA.1 Management of security attributes

- 3411 FMT_SMR.1 Security roles
- 3412 **FMT_MSA.2.1**
- 3413 **The TSF shall ensure that only secure values are accepted for [assignment: *list of security***
 3414 ***attributes*].**
- 3415 **12.4.13 FMT_MSA.3 Static attribute initialization**
- 3416 **Component relationships**
- 3417 Hierarchical to: No other components.
- 3418 Dependencies: FMT_MSA.1 Management of security attributes
 3419 FMT_SMR.1 Security roles
- 3420 **FMT_MSA.3.1**
- 3421 **The TSF shall enforce the [assignment: *access control SFP, information flow control SFP***
 3422 **to provide [selection, choose one of: *restrictive, permissive, [assignment: *other property*]***
 3423 **default values for security attributes that are used to enforce the SFP.**
- 3424 **FMT_MSA.3.2**
- 3425 **The TSF shall allow the [assignment: *the authorized identified roles*] to specify alternative**
 3426 **initial values to override the default values when an object or information is created.**
- 3427 **12.4.14 FMT_MSA.4 Security attribute value inheritance**
- 3428 **Component relationships**
- 3429 Hierarchical to: No other components.
- 3430 Dependencies: [FDP_ACC.1 Subset access control, or
 3431 FDP_IFC.1 Subset information flow control]
- 3432 **FMT_MSA.4.1**
- 3433 **The TSF shall use the following rules to set the value of security attributes: [assignment:**
 3434 ***rules for setting the values of security attributes*].**
- 3435 **12.5 Management of TSF data (FMT_MTD)**
- 3436 **12.5.1 Family behaviour**
- 3437 This family allows authorized users (roles) control over the management of TSF data.
- 3438 **12.5.2 Components leveling and description**
- 3439 Figure 50 shows the component leveling for this family.



3440 **Figure 50 — FMT_MTD: Component leveling**

3441 **Editors' Note**

3442 SMEs are asked to review the component leveling and the hierarchy/dependency information.
3443 If FMT_MTD.2 and FMT_MTD.3 are dependent on FMT_MTD.1 shouldn't they be hierarchical?

3444
3445 FMT_MTD.1 Management of TSF data allows authorized users to manage TSF data.

3446 FMT_MTD.2 Management of limits on TSF data specifies the action to be taken if limits on TSF
3447 data are reached or exceeded.

3448 FMT_MTD.3 Secure TSF data ensures that values assigned to TSF data are valid with respect to
3449 the secure state.

3450 12.5.3 Management of FMT_MTD.1

3451 The following actions **could** be considered for the management functions in FMT:

3452 a) Managing the group of roles that **can** interact with the TSF data.

3453 12.5.4 Management of FMT_MTD.2

3454 The following actions **could** be considered for the management functions in FMT:

3455 a) Managing the group of roles that **can** interact with the limits on the TSF data.

3456 12.5.5 Management of FMT_MTD.3

3457 The following actions **could** be considered for the management functions in FMT:

3458 a) There are no management activities foreseen.

3459 12.5.6 Audit of FMT_MTD.1

3460 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3461 in the PP/ST:

3462 a) Basic: All modifications to the values of TSF data.

3463 12.5.7 Audit of FMT_MTD.2

3464 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3465 in the PP/ST:

3466 a) Basic: All modifications to the limits on TSF data.

3467 b) Basic: All modifications in the actions to be taken in case of violation of the limits.

3468 12.5.8 Audit of FMT_MTD.3

3469 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3470 in the PP/ST:

3471 a) Minimal: All rejected values of TSF data.

3472 12.5.9 FMT_MTD.1 Management of TSF data

3473 Component relationships

3474 Hierarchical to: No other components.

3475 Dependencies: FMT_SMR.1 Security roles

3476 FMT_SMF.1 Specification of Management Functions

3477 **FMT_MTD.1.1**

3478 The TSF **shall** restrict the ability to [selection: *change_default, query, modify, delete, clear,*
3479 *[assignment: other operations]*] the [assignment: *list of TSF data*] to [assignment: *the*
3480 *authorized identified roles*].

3481 **12.5.10 FMT_MTD.2 Management of limits on TSF data**3482 **Component relationships**

3483	Hierarchical to:	No other components.
3484	Dependencies:	FMT_MTD.1 Management of TSF data
3485		FMT_SMR.1 Security roles

3486 **FMT_MTD.2.1**

3487 The TSF **shall** restrict the specification of the limits for [assignment: *list of TSF data*] to
3488 [assignment: *the authorized identified roles*].

3489 **FMT_MTD.2.2**

3490 The TSF **shall** take the following actions, if the TSF data are at, or exceed, the indicated
3491 limits: [assignment: *actions to be taken*].

3492 **12.5.11 FMT_MTD.3 Secure TSF data**3493 **Component relationships**

3494	Hierarchical to:	No other components.
3495	Dependencies:	FMT_MTD.1 Management of TSF data

3496 **FMT_MTD.3.1**

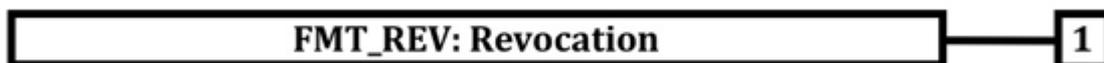
3497 The TSF **shall** ensure that only secure values are accepted for [assignment: *list of TSF*
3498 *data*].

3499 **12.6 Revocation (FMT_REV)**3500 **12.6.1 Family behaviour**

3501 This family addresses revocation of security attributes for a variety of entities within a TOE.

3502 **12.6.2 Components leveling and description**

3503 Figure 51 shows the component leveling for this family.



3504 **Figure 51 — FMT_REV: Component leveling**

3505 FMT_REV.1 Revocation provides for revocation of security attributes to be enforced at some
3506 point in time.

3507 **12.6.3 Management of FMT_REV.1**

3508 The following actions **could** be considered for the management functions in FMT:

3509 a) Managing the group of roles that **can** invoke revocation of security attributes;

- 3510 b) Managing the lists of users, subjects, objects, and other resources for which
- 3511 revocation is possible;
- 3512 c) Managing the revocation rules.

3513 **12.6.4 Audit of FMT_REV.1**

3514 The following actions **should** be auditable if FAU_GEN Security audit data generation is included

3515 in the PP/ST:

- 3516 a) Minimal: Unsuccessful revocation of security attributes;
- 3517 b) Basic: All attempts to revoke security attributes.

3518 **12.6.5 FMT_REV.1 Revocation**

3519 **Component relationships**

- 3520 Hierarchical to: No other components.
- 3521 Dependencies: FMT_SMR.1 Security roles

3522 **FMT_REV.1.1**

3523 The TSF **shall** restrict the ability to revoke [assignment: *list of security attributes*]

3524 associated with the [selection: *users, subjects, objects, [assignment: other additional*

3525 *resources*]] under the control of the TSF to [assignment: *the authorized identified roles*].

3526 **FMT_REV.1.2**

3527 The TSF **shall** enforce the rules [assignment: *specification of revocation rules*].

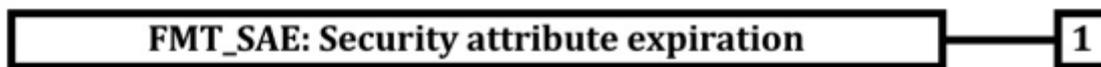
3528 **12.7 Security attribute expiration (FMT_SAE)**

3529 **12.7.1 Family behaviour**

3530 This family addresses the capability to enforce time limits for the validity of security attributes.

3531 **12.7.2 Components leveling and description**

3532 Figure 52 shows the component leveling for this family.



3533 **Figure 52 — FMT_SAE: Component leveling**

3534 FMT_SAE.1 Time-limited authorization provides the capability for an authorized user to specify

3535 an expiration time on specified security attributes.

3536 **12.7.3 Management of FMT_SAE.1**

3537 The following actions **could** be considered for the management functions in FMT:

- 3538 a) Managing the list of security attributes for which expiration is to be supported;
- 3539 b) The actions to be taken if the expiration time has passed.

3540 **12.7.4 Audit of FMT_SAE.1**

3541 The following actions **should** be auditable if FAU_GEN Security audit data generation is included

3542 in the PP/ST:

- 3543 a) Basic: Specification of the expiration time for an attribute;

3544 b) Basic: Action taken due to attribute expiration.

3545 12.7.5 FMT_SAE.1 Time-limited authorization

3546 Component relationships

3547 Hierarchical to: No other components.

3548 Dependencies: FMT_SMR.1 Security roles

3549 FPT_STM.1 Reliable time stamps

3550 FMT_SAE.1.1

3551 The TSF **shall** restrict the capability to specify an expiration time for [assignment: *list of*
3552 *security attributes for which expiration is to be supported*] to [assignment: *the authorized*
3553 *identified roles*].

3554 FMT_SAE.1.2

3555 For each of these security attributes, the TSF **shall** be able to [assignment: *list of actions*
3556 *to be taken for each security attribute*] after the expiration time for the indicated security
3557 attribute has passed.

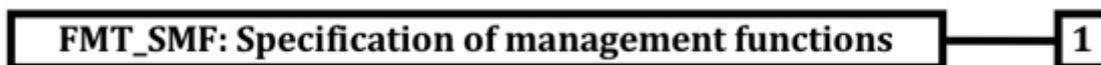
3558 12.8 Specification of Management Functions (FMT_SMF)

3559 12.8.1 Family behaviour

3560 This family allows the specification of the management functions to be provided by the TOE.
3561 Management functions provide TSFI that allow administrators to define the parameters that
3562 control the operation of security-related aspects of the TOE, such as data protection attributes,
3563 TOE protection attributes, audit attributes, and identification and authentication attributes.
3564 Management functions also include those functions performed by an operator to ensure
3565 continued operation of the TOE, such as backup and recovery. This family works in conjunction
3566 with the other components in the FMT: Security management class: the component in this
3567 family calls out the management functions, and other families in FMT: Security management
3568 restrict the ability to use these management functions.

3569 12.8.2 Components leveling and description

3570 Figure 53 shows the component leveling for this family.



3571 **Figure 53 — FMT_SMF: Component leveling**

3572 FMT_SMF.1 Specification of Management Functions requires that the TSF provide specific
3573 management functions.

3574 12.8.3 Management of FMT_SMF.1

3575 The following actions **could** be considered for the management functions in FMT:

3576 a) There are no management activities foreseen.

3577 12.8.4 Audit of FMT_SMF.1

3578 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3579 in the PP/ST:

3580 a) Minimal: Use of the management functions.

3581 **12.8.5 FMT_SMF.1 Specification of Management Functions**

3582 **Component relationships**

3583 Hierarchical to: No other components.

3584 Dependencies: No dependencies.

3585 **FMT_SMF.1.1**

3586 **The TSF shall be capable of performing the following management functions:**
 3587 **[assignment: list of management functions to be provided by the TSF].**

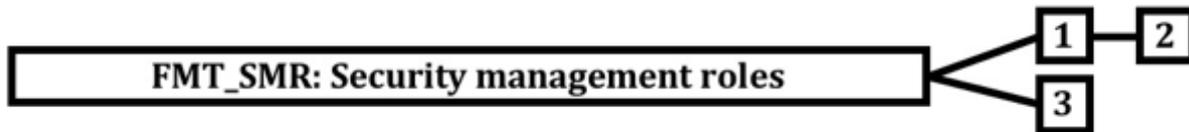
3588 **12.9 Security management roles (FMT_SMR)**

3589 **12.9.1 Family behaviour**

3590 This family is intended to control the assignment of different roles to users. The capabilities of
 3591 these roles with respect to security management are described in the other families in this class.

3592 **12.9.2 Components leveling and description**

3593 Figure 54 shows the component leveling for this family.



3594 **Figure 54 — FMT_SMR: Component leveling**

3595 FMT_SMR.1 Security roles specifies the roles with respect to security that the TSF recognizes.

3596 FMT_SMR.2 Restrictions on security roles specifies that in addition to the specification of the
 3597 roles, there are rules that control the relationship between the roles.

3598 FMT_SMR.3 Assuming roles, requires that an explicit request is given to the TSF to assume a
 3599 role.

3600 **12.9.3 Management of FMT_SMR.1**

3601 The following actions could be considered for the management functions in FMT:

- 3602 a) Managing the group of users that are part of a role.

3603 **12.9.4 Management of FMT_SMR.2**

3604 The following actions could be considered for the management functions in FMT:

- 3605 a) Managing the group of users that are part of a role;
- 3606 b) Managing the conditions that the roles must satisfy.

3607 **12.9.5 Management of FMT_SMR.3**

3608 There are no management activities foreseen.

3609 **12.9.6 Audit of FMT_SMR.1**

3610 The following actions should be auditable if FAU_GEN Security audit data generation is included
 3611 in the PP/ST:

- 3612 a) Minimal: modifications to the group of users that are part of a role;
- 3613 b) Detailed: every use of the rights of a role.

3614 **12.9.7 Audit of FMT_SMR.2**

3615 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3616 in the PP/ST:

- 3617 a) Minimal: modifications to the group of users that are part of a role;
3618 b) Minimal: unsuccessful attempts to use a role due to the given conditions on the
3619 roles;
3620 c) Detailed: every use of the rights of a role.

3621 **12.9.8 Audit of FMT_SMR.3**

3622 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3623 in the PP/ST:

- 3624 a) Minimal: explicit request to assume a role.

3625 **12.9.9 FMT_SMR.1 Security roles**3626 **Component relationships**

3627 Hierarchical to: No other components.
3628 Dependencies: FIA_UID.1 Timing of identification

3629 **FMT_SMR.1.1**

3630 **The TSF **shall** maintain the roles [assignment: *the authorized identified roles*].**

3631 **FMT_SMR.1.2**

3632 **The TSF **shall** be able to associate users with roles.**

3633 **12.9.10 FMT_SMR.2 Restrictions on security roles**3634 **Component relationships**

3635 Hierarchical to: FMT_SMR.1 Security roles
3636 Dependencies: FIA_UID.1 Timing of identification

3637 **FMT_SMR.2.1**

3638 The TSF **shall** maintain the roles: [assignment: *authorized identified roles*].

3639 **FMT_SMR.2.2**

3640 The TSF **shall** be able to associate users with roles.

3641 **FMT_SMR.2.3**

3642 **The TSF **shall** ensure that the conditions [assignment: *conditions for the different roles*]
3643 are satisfied.**

3644 **12.9.11 FMT_SMR.3 Assuming roles**

3645 Hierarchical to: No other components.
3646 Dependencies: FMT_SMR.1 Security roles

3647 **FMT_SMR.3.1**

3648 **The TSF **shall** require an explicit request to assume the following roles: [assignment: *the*
3649 *roles*].**

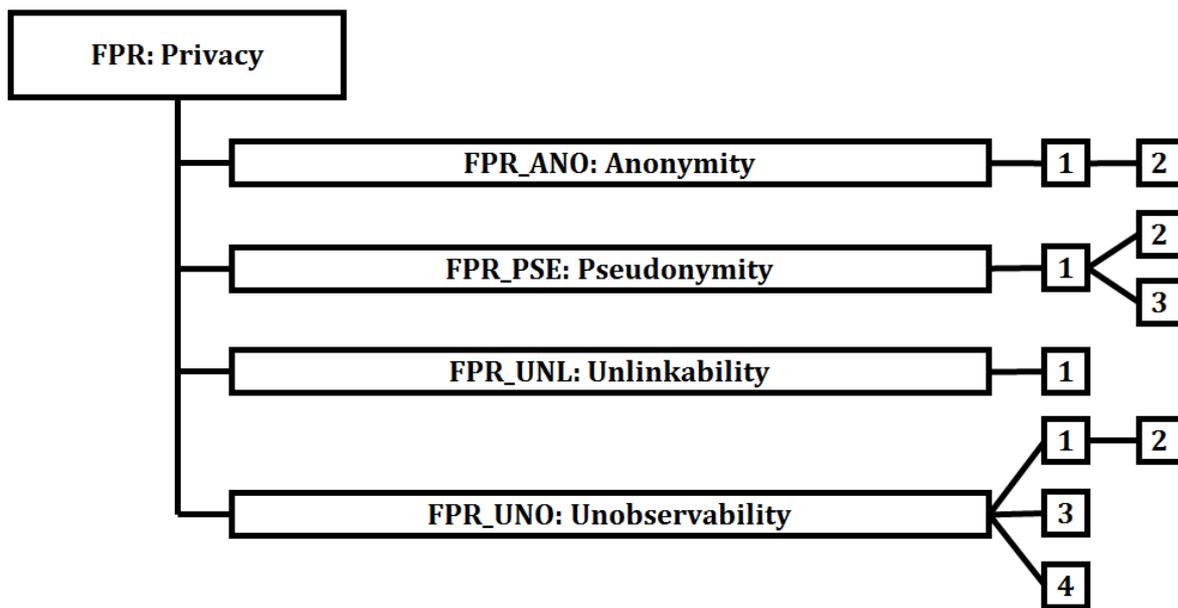
3651 **13 Class FPR: Privacy**

3652 **13.1 Class description**

3653 This class contains privacy requirements. These requirements provide a user protection against
3654 discovery and misuse of identity by other users.

3655 Figure 55 shows the decomposition of this class, it's families and components. Elements are not
3656 shown in the figure.

3657 Annex I provides explanatory information for this class and **should** be consulted when using the
3658 components identified in this class.



3659

3660

Figure 55 — FPR: Privacy class decomposition

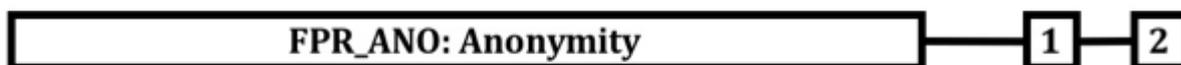
3661 **13.2 Anonymity (FPR_ANO)**

3662 **13.2.1 Family behaviour**

3663 This family ensures that a user **may** use a resource or service without disclosing the user's
3664 identity. The requirements for Anonymity provide protection of the user identity. Anonymity is
3665 not intended to protect the subject identity.

3666 **13.2.2 Components leveling and description**

3667 Figure 56 shows the component leveling for this family.



3668

Figure 56 — FPR_ANO: Component leveling

3669 FPR_ANO.1 Anonymity, requires that other users or subjects are unable to determine the
3670 identity of a user bound to a subject or operation.

3671 FPR_ANO.2 Anonymity without soliciting information enhances the requirements of FPR_ANO.1
3672 Anonymity by ensuring that the TSF does not ask for the user identity.

3673 **13.2.3 Management of FPR_ANO.1, FPR_ANO.2**

3674 The following actions **could** be considered for the management functions in FMT:

- 3675 a) There are no management activities foreseen.

3676 **13.2.4 Audit of FPR_ANO.1, FPR_ANO.2**

3677 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3678 in the PP/ST:

- 3679 a) Minimal: The invocation of the anonymity mechanism.

3680 **13.2.5 FPR_ANO.1 Anonymity**

3681 **Component relationships**

3682 Hierarchical to: No other components.

3683 Dependencies: No dependencies.

3684 **FPR_ANO.1.1**

3685 The TSF **shall** ensure that [assignment: *set of users and/or subjects*] are unable to
3686 determine the real user name bound to [assignment: *list of subjects and/or operations*
3687 *and/or objects*].

3688 **13.2.6 FPR_ANO.2 Anonymity without soliciting information**

3689 **Component relationships**

3690 Hierarchical to: FPR_ANO.1 Anonymity

3691 Dependencies: No dependencies.

3692 **FPR_ANO.2.1**

3693 The TSF **shall** ensure that [assignment: *set of users and/or subjects*] are unable to determine the
3694 real user name bound to [assignment: *list of subjects and/or operations and/or objects*].

3695 **FPR_ANO.2.2**

3696 The TSF **shall** provide [assignment: *list of services*] to [assignment: *list of subjects*]
3697 without soliciting any reference to the real user name.

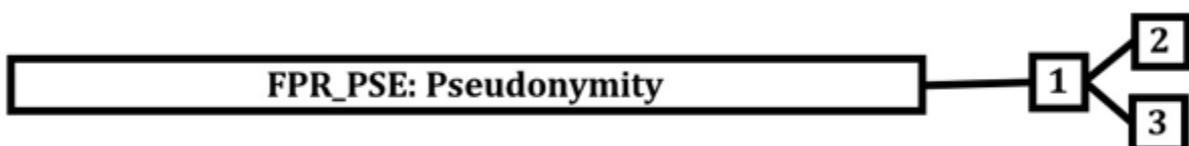
3698 **13.3 Pseudonymity (FPR_PSE)**

3699 **13.3.1 Family behaviour**

3700 This family ensures that a user **may** use a resource or service without disclosing its user
3701 identity but **can** still be accountable for that use.

3702 **13.3.2 Components leveling and description**

3703 Figure 57 shows the component leveling for this family.



3704 **Figure 57 — FPR_PSE: Component leveling**

3705 FPR_PSE.1 Pseudonymity requires that a set of users and/or subjects are unable to determine
3706 the identity of a user bound to a subject or operation, but that this user is still accountable for
3707 its actions.

3708 FPR_PSE.2 Reversible pseudonymity, requires the TSF to provide a capability to determine the
3709 original user identity based on a provided alias.

3710 FPR_PSE.3 Alias pseudonymity, requires the TSF to follow certain construction rules for the
3711 alias to the user identity.

3712 **13.3.3 Management of FPR_PSE.1, FPR_PSE.2, FPR_PSE.3**

3713 The following actions **could** be considered for the management functions in FMT:

3714 a) There are no management activities foreseen.

3715 **13.3.4 Audit of FPR_PSE.1, FPR_PSE.2, FPR_PSE.3**

3716 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3717 in the PP/ST:

3718 a) Minimal: The subject/user that requested resolution of the user identity **should** be
3719 audited.

3720 **13.3.5 FPR_PSE.1 Pseudonymity**

3721 **Component relationships**

3722 Hierarchical to: No other components.

3723 Dependencies: No dependencies.

3724 **FPR_PSE.1.1**

3725 **The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to**
3726 **determine the real user name bound to [assignment: *list of subjects and/or operations***
3727 ***and/or objects*].**

3728 **FPR_PSE.1.2**

3729 **The TSF shall be able to provide [assignment: *number of aliases*] aliases of the real user**
3730 **name to [assignment: *list of subjects*].**

3731 **FPR_PSE.1.3**

3732 **The TSF shall [selection, choose one of: *determine an alias for a user, accept the alias from***
3733 ***the user*] and verify that it conforms to the [assignment: *alias metric*].**

3734 **13.3.6 FPR_PSE.2 Reversible pseudonymity**

3735 **Component relationships**

3736 Hierarchical to: FPR_PSE.1 Pseudonymity

3737 Dependencies: FIA_UID.1 Timing of identification

3738 **FPR_PSE.2.1**

3739 **The TSF shall ensure that [assignment: *set of users and/or subjects*] are unable to determine the**
3740 **real user name bound to [assignment: *list of subjects and/or operations and/or objects*].**

3741 **FPR_PSE.2.2**

3742 The TSF **shall** be able to provide [assignment: *number of aliases*] aliases of the real user name to
3743 [assignment: *list of subjects*].

3744 **FPR_PSE.2.3**

3745 The TSF **shall** [selection, choose one of: *determine an alias for a user, accept the alias from the*
3746 *user*] and verify that it conforms to the [assignment: *alias metric*].

3747 **FPR_PSE.2.4**

3748 **The TSF shall provide** [selection: *an authorized user, [assignment: list of trusted subjects]*
3749 **a capability to determine the user identity based on the provided alias only under the**
3750 **following** [assignment: *list of conditions*].

3751 **13.3.7 FPR_PSE.3 Alias pseudonymity**3752 **Component relationships**

3753 Hierarchical to: FPR_PSE.1 Pseudonymity

3754 Dependencies: No dependencies.

3755 **FPR_PSE.3.1**

3756 The TSF **shall** ensure that [assignment: *set of users and/or subjects*] are unable to determine the
3757 real user name bound to [assignment: *list of subjects and/or operations and/or objects*].

3758 **FPR_PSE.3.2**

3759 The TSF **shall** be able to provide [assignment: *number of aliases*] aliases of the real user name to
3760 [assignment: *list of subjects*].

3761 **FPR_PSE.3.3**

3762 The TSF **shall** [selection, choose one of: *determine an alias for a user, accept the alias from the*
3763 *user*] and verify that it conforms to the [assignment: *alias metric*].

3764 **FPR_PSE.3.4**

3765 **The TSF shall provide an alias to the real user name which shall be identical to an alias**
3766 **provided previously under the following** [assignment: *list of conditions*] **otherwise the**
3767 **alias provided shall be unrelated to previously provided aliases.**

3768 **13.4 Unlinkability (FPR_UNL)**3769 **13.4.1 Family behaviour**

3770 This family ensures that selected entities **may** be linked together without external entities being
3771 able to back trace these links.

3772 **13.4.2 Components leveling and description**

3773 Figure 58 shows the component leveling for this family.



3775 **Figure 58 — FPR_UNL: Component leveling**

3776 **FPR_UNL.1 Unlinkability of operations** requires that users and/or subjects are unable to
 3777 determine whether the same user caused certain specific operations in the system, or whether
 3778 operations are related in some other manner. This component ensures that users cannot link
 3779 different operations in the system and thereby obtain information.

3780 **13.4.3 Management of FPR_UNL.1**

3781 The following actions **could** be considered for the management functions in FMT:

- 3782 a) The management of the unlinkability function.

3783 **13.4.4 Audit of FPR_UNL.1**

3784 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 3785 in the PP/ST:

- 3786 a) Minimal: The invocation of the unlinkability mechanism.

3787 **13.4.5 FPR_UNL.1 Unlinkability of operations**

3788 **Component relationships**

3789 Hierarchical to: No other components.

3790 Dependencies: No dependencies.

3791 **FPR_UNL.1.1**

3792 **The TSF shall ensure that [assignment: set of entities and/or operations] are unable to**
 3793 **determine whether [assignment: list of entities and/or operations] [selection: were**
 3794 **caused by the same user, are related as follows [assignment: list of relations]].**

3795 NOTE For “operations” the term transactions should be used.

3796 **Editors' Note:**
 3797 **The above NOTE is not clear, it is better to reword it or delete it.**
 3798 **Unless comments are received with a proposed rewording this note will be deleted in the next draft.**

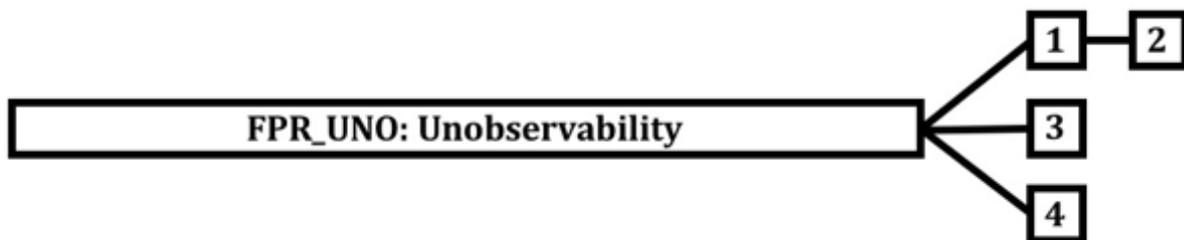
3799 **13.5 Unobservability (FPR_UNO)**

3800 **13.5.1 Family behaviour**

3801 This family ensures that a user **may** use a resource or service without others, especially third
 3802 parties, being able to observe that the resource or service is being used.

3803 **13.5.2 Components leveling and description**

3804 Figure 59 shows the component leveling for this family.



3805 **Figure 59 — FPR_UNO: Component leveling**

3806 FPR_UNO.1 Unobservability, requires that users and/or subjects cannot determine whether an
 3807 operation is being performed.

3808 FPR_UNO.2 Allocation of information impacting unobservability, requires that the TSF provide
 3809 specific mechanisms to avoid the concentration of privacy related information within the TOE.
 3810 Such concentrations might impact unobservability if a security compromise occurs.

3811 FPR_UNO.3 Unobservability without soliciting information, requires that the TSF does not try to
 3812 obtain privacy related information that might be used to compromise unobservability.

3813 FPR_UNO.4 Authorized user observability, requires the TSF to provide one or more authorized
 3814 users with a capability to observe the usage of resources and/or services.

3815 **13.5.3 Management of FPR_UNO.1, FPR_UNO.2**

3816 The following actions **could** be considered for the management functions in FMT:

3817 a) The management of the behaviour of the unobservability function.

3818 **13.5.4 Management of FPR_UNO.3**

3819 The following actions **could** be considered for the management functions in FMT:

3820 a) There are no management activities foreseen.

3821 **13.5.5 Management of FPR_UNO.4**

3822 The following actions **could** be considered for the management functions in FMT:

3823 a) The list of authorized users that are capable of determining the occurrence of
 3824 operations.

3825 **13.5.6 Audit of FPR_UNO.1, FPR_UNO.2**

3826 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 3827 in the PP/ST:

3828 a) Minimal: The invocation of the unobservability mechanism.

3829 **13.5.7 Audit of FPR_UNO.3**

3830 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 3831 in the PP/ST:

3832 a) There are no auditable events foreseen.

3833 **13.5.8 Audit of FPR_UNO.4**

3834 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 3835 in the PP/ST:

3836 a) Minimal: The observation of the use of a resource or service by a user or subject.

3837 **13.5.9 FPR_UNO.1 Unobservability**

3838 **Component relationships**

3839 Hierarchical to: No other components.

3840 Dependencies: No dependencies.

3841 **FPR_UNO.1.1**

3842 **The TSF shall ensure that [assignment: list of users and/or subjects] are unable to**
 3843 **observe the operation [assignment: list of operations] on [assignment: list of objects] by**
 3844 **[assignment: list of protected users and/or subjects].**

3845 **13.5.10 FPR_UNO.2 Allocation of information impacting unobservability**

3846 **Component relationships**

3847 Hierarchical to: FPR_UNO.1 Unobservability

3848 Dependencies: No dependencies.

3849 **FPR_UNO.2.1**

3850 The TSF **shall** ensure that [assignment: *list of users and/or subjects*] are unable to observe the
3851 operation [assignment: *list of operations*] on [assignment: *list of objects*] by [assignment: *list of*
3852 *protected users and/or subjects*].

3853 **FPR_UNO.2.2**

3854 The TSF **shall** allocate the [assignment: *unobservability related information*] among
3855 different parts of the TOE such that the following conditions hold during the lifetime of
3856 the information: [assignment: *list of conditions*].

3857 **13.5.11 FPR_UNO.3 Unobservability without soliciting information**

3858 **Component relationships**

3859 Hierarchical to: No other components.

3860 Dependencies: FPR_UNO.1 Unobservability

3861 **FPR_UNO.3.1**

3862 The TSF **shall** provide [assignment: *list of services*] to [assignment: *list of subjects*]
3863 without soliciting any reference to [assignment: *privacy related information*].

3864 **13.5.12 FPR_UNO.4 Authorized user observability**

3865 **Component relationships**

3866 Hierarchical to: No other components.

3867 Dependencies: No dependencies.

3868 **FPR_UNO.4.1**

3869 The TSF **shall** provide [assignment: *set of authorized users*] with the capability to observe
3870 the usage of [assignment: *list of resources and/or services*].

3871

3872 **14 Class FPT: Protection of the TSF**

3873 **14.1 Class description**

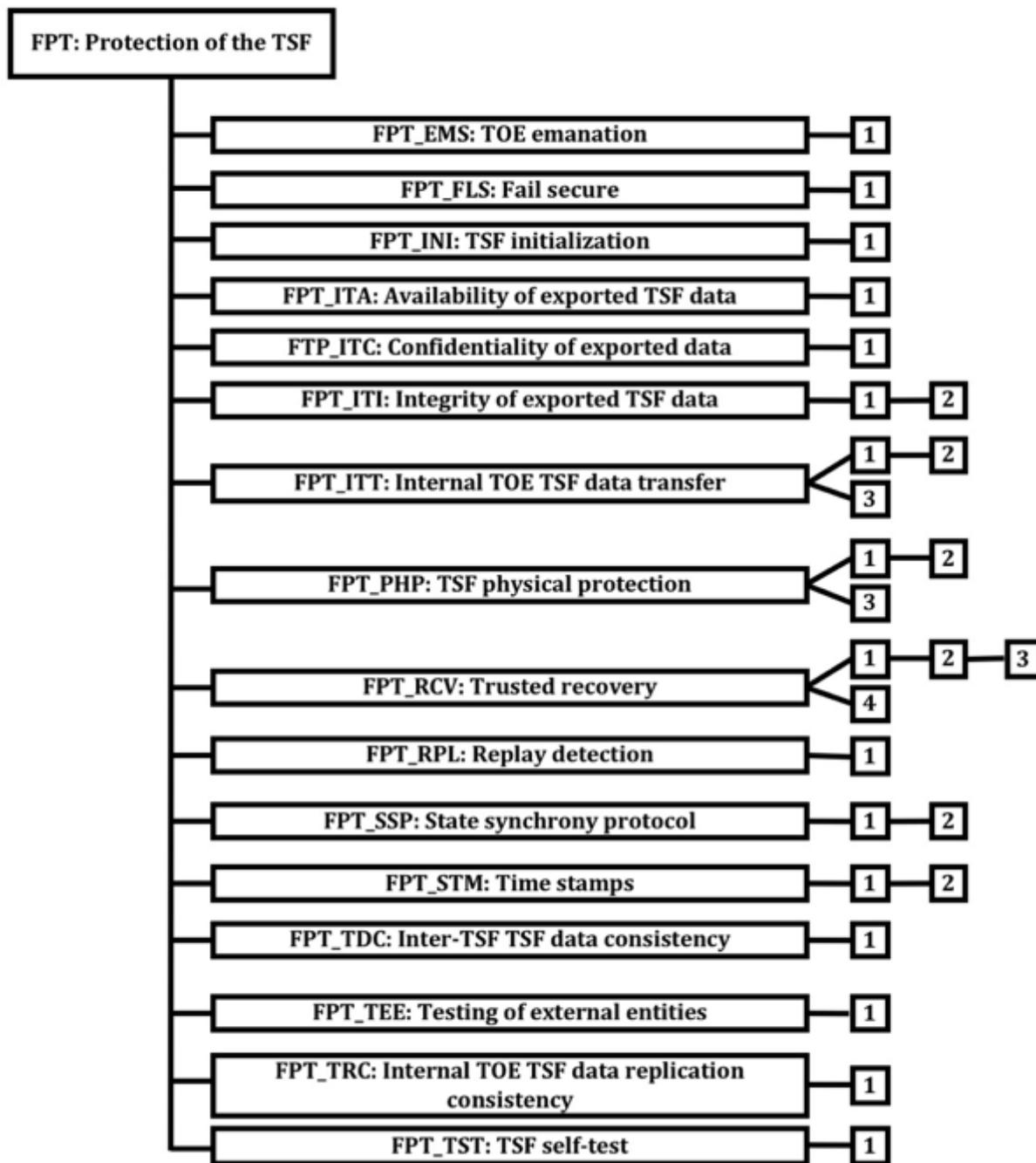
3874 This class contains families of functional requirements that relate to the integrity and
3875 management of the mechanisms that constitute the TSF and to the integrity of TSF data. In some
3876 sense, families in this class **may** appear to duplicate components in the FDP: User data
3877 protection class; they **may** even be implemented using the same mechanisms. However, FDP:
3878 User data protection focuses on user data protection, while FPT: Protection of the TSF focuses
3879 on TSF data protection. In fact, Components from the FPT: Protection of the TSF class are
3880 necessary to provide requirements that the SFPs in the TOE cannot be tampered with or
3881 bypassed.

3882 From the point of view of this class, regarding to the TSF there are three significant elements:

- 3883 a) The TSF's implementation, which executes and implements the mechanisms that
3884 enforce the SFRs.
- 3885 b) The TSF's data, which are the administrative databases that guide the enforcement
3886 of the SFRs.
- 3887 c) The external entities that the TSF **may** interact with in order to enforce the SFRs.

3888 Figure 60 shows the decomposition of this class, it's families and components. Elements are not
3889 shown in the figure.

3890 Annex J provides explanatory information for this class and **should** be consulted when using the
 3891 components identified in this class.



3892 **Figure 60 — FPT: Protection of the TSF class decomposition**

3893 **14.2 TOE emanation (FPT_EMS)**

3894 **14.2.1 Family behaviour**

3895 The family FPT_EMS (TOE Emanation) of the Class FPT (Protection of the TSF) is defined here
 3896 to describe the IT security functional requirements of the TOE related to leakage of information
 3897 based on emanation.

3898 The TOE **shall** prevent attacks against the TOE and secret data processed by the TOE where the
 3899 attack is based on external observable phenomena of the TOE during its operation. Hereby,

3900 different types of emissions and interfaces of the TOE as well as different types of TSF data and
 3901 user data may be addressed.

EXAMPLE

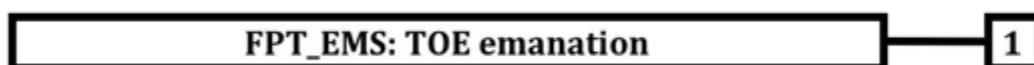
Examples of such attacks against the TOE and its processed secret data are simple power analysis (SPA), differential power analysis (DPA), simple electromagnetic analysis (SEMA), differential electromagnetic analysis (DEMA), timing attacks, padding oracle attacks, cache miss attacks, etc.

3902

3903 This family describes the functional requirements for the limitation of intelligible emanations
 3904 which are not directly addressed by any other component of ISO/IEC 15408-2.

3905 **14.2.2 Components leveling and description**

3906 Figure 61 shows the component leveling for this family.



3907

Figure 61 — FPT_EMS: Component leveling

3908 This family consists of only one component, FPT_EMS.1 Emanation of TSF and User data, which
 3909 defines requirements for the TOE to mitigate intelligible emanations.

3910 **14.2.3 Management of FPT_EMS.1**

3911 The following actions **could** be considered for the management functions in FMT:

- 3912 a) There are no management activities foreseen.

3913 **14.2.4 Audit of FPT_EMS.1**

3914 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 3915 in the PP/ST:

- 3916 a) There are no auditable events foreseen.

3917 **14.2.5 FPT_EMS.1 Emanation of TSF and User data**

3918 **Component relationships**

3919 Hierarchical to: No other components.

3920 Dependencies: No dependencies.

3921 **FPT_EMS.1.1**

3922 The TSF **shall** ensure that the TOE does not emit emissions over its interfaces in such
 3923 amount that these emissions enable access to TSF data and user data as specified in the
 3924 following table:

Table 2 - FPT_EMS.1.1 Table

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

3926

3927 **Editors' note:**

3928 It has been identified that the term “interface” as used in FPT_EMS above might be too restricted in view
 3929 of the already existing definition of “interface” in ISO/IEC 15408-1 (1st CD) on the one hand and the
 3930 needs for FPT_EMS on the other hand. The solution could be to widen appropriately the scope and
 3931 therefore the related definition of “interface” in Part 1, or alternatively to introduce a new term as e.g.
 3932 “attack surface” (including its definition in Part 1) and use this new term instead of “interface” in
 3933 FPT_EMS above. In order to avoid any impact on other CC texts it seems that the second proposal is the
 3934 better one. Contributions by other CC experts to solve this open issue are welcome.

3935

3936 Another way to address this is to introduce a second term in ISO/IEC 15408 defining “interface” and
 3937 specifying that the context is for it to be used is for <Emanation>

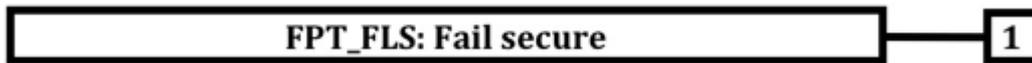
3938 **14.3 Fail secure (FPT_FLS)**

3939 **14.3.1 Family behaviour**

3940 The requirements of this family ensure that the TOE will always enforce its SFRs in the event of
 3941 identified categories of failures in the TSF.

3942 **14.3.2 Components leveling and description**

3943 Figure 62 shows the component leveling for this family.



3944 **Figure 62 — FPT_FLS: Component leveling**

3945 This family consists of only one component, FPT_FLS.1 Failure with preservation of secure
 3946 state, which requires that the TSF preserve a secure state in the face of the identified failures.

3947 **14.3.3 Management of FPT_FLS.1**

3948 The following actions **could** be considered for the management functions in FMT:

- 3949 a) There are no management activities foreseen.

3950 **14.3.4 Audit of FPT_FLS.1**

3951 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 3952 in the PP/ST:

- 3953 a) Basic: Failure of the TSF.

3954 **14.3.5 FPT_FLS.1 Failure with preservation of secure state**

3955 **Component relationships**

3956 Hierarchical to: No other components.

3957 Dependencies: No dependencies.

3958 **FPT_FLS.1.1**

3959 **The TSF shall preserve a secure state when the following types of failures occur:**
 3960 **[assignment: list of types of failures in the TSF].**

3961 **14.4 TSF initialization (FPT_INI)**3962 **14.4.1 Family behaviour**

3963 This family describes the functional requirements for the initialization of the TSF by a dedicated
3964 function of the TOE that ensures the initialization in a correct and secure operational state.

3965 **14.4.2 Components leveling and description**

3966 Figure 63 shows the component leveling for this family.



3967 **Figure 63 — FPT_INI: Component leveling**

3968 This family consists of only one component, Component FPT_INI.1. This component requires the
3969 TOE to provide a TSF initialization function that brings the TSF into a secure operational state
3970 at power-on.

3971 **14.4.3 Management of FPT_INI.1**

3972 The following actions **could** be considered for the management functions in FMT:

3973 a) There are no management activities foreseen.

3974 **14.4.4 Audit of FPT_INI.1**

3975 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
3976 in the PP/ST:

3977 a) There are no auditable events foreseen.

3978 **14.4.5 FPT_INI.1 TSF initialization**3979 **Component relationships**

3980 Hierarchical to: No other components.

3981 Dependencies: No dependencies.

3982 **FPT_INI.1.1**

3983 **The TOE shall provide an initialization function which is self-protected with regard to the**
3984 **following properties [selection: *integrity, authenticity, unicity*, [assignment: *list of***
3985 ***properties or none*]].**

3986 **FPT_INI.1.2**

3987 **The TOE initialization function shall verify the [selection: *authenticity, integrity*] of**
3988 **[assignment: *list of TSF firmware, software, or data*] prior to establishing the TSF in a**
3989 **secure initial state.**

3990 **FPT_INI.1.3**

3991 **The TOE initialization function shall detect and respond to errors and failures during**
3992 **initialization such that the TOE either successfully completes initialization or is halted.**

3993 **FPT_INI.1.4**

3994 **The TOE initialization function shall not be able to arbitrarily interact with the TSF after**
3995 **TOE initialization completes.**

3996 **14.5 Availability of exported TSF data (FPT_ITA)**

3997 **14.5.1 Family behaviour**

3998 This family defines the rules for the prevention of loss of availability of TSF data moving
3999 between the TSF and another trusted IT product.

4000 **14.5.2 Components leveling and description**

4001 Figure 64 shows the component leveling for this family.



4002

4003 **Figure 64 — FPT_ITA: Component leveling**

4004 This family consists of only one component, FPT_ITA.1 Inter-TSF availability within a defined
4005 availability metric. This component requires that the TSF ensure, to an identified degree of
4006 probability, the availability of TSF data provided to another trusted IT product.

4007 **14.5.3 Management of FPT_ITA.1**

4008 The following actions **could** be considered for the management functions in FMT:

- 4009 a) management of the list of types of TSF data that must be available to another
4010 trusted IT product.

4011 **14.5.4 Audit of FPT_ITA.1**

4012 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4013 in the PP/ST:

- 4014 a) Minimal: the absence of TSF data when required by a TOE.

4015 **14.5.5 FPT_ITA.1 Inter-TSF availability within a defined availability metric**

4016 **Component relationships**

4017 Hierarchical to: No other components.

4018 Dependencies: No dependencies.

4019 **FPT_ITA.1.1**

4020 **The TSF shall ensure the availability of [assignment: *list of types of TSF data*] provided to**
4021 **another trusted IT product within [assignment: *a defined availability metric*] given the**
4022 **following conditions [assignment: *conditions to ensure availability*].**

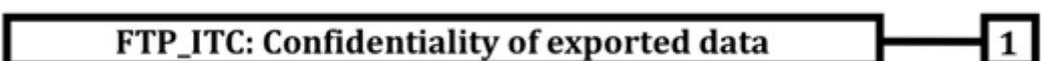
4023 **14.6 Confidentiality of exported TSF data (FPT_ITC)**

4024 **14.6.1 Family behaviour**

4025 This family defines the rules for the protection from unauthorized disclosure of TSF data during
4026 transmission between the TSF and another trusted IT product.

4027 **14.6.2 Components leveling and description**

4028 Figure 65 shows the component leveling for this family.



4029

4030 **Figure 65 — FPT_ITC: Component leveling**

4031 This family consists of only one component, FPT_ITC.1 Inter-TSF confidentiality during
4032 transmission, which requires that the TSF ensure that data transmitted between the TSF and
4033 another trusted IT product is protected from disclosure while in transit.

4034 **14.6.3 Management of FPT_ITC.1**

4035 The following actions **could** be considered for the management functions in FMT:

4036 a) There are no management activities foreseen.

4037 **14.6.4 Audit of FPT_ITC.1**

4038 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4039 in the PP/ST:

4040 a) There are no auditable events foreseen.

4041 **14.6.5 FPT_ITC.1 Inter-TSF confidentiality during transmission**

4042 **Component relationships**

4043 Hierarchical to: No other components.

4044 Dependencies: No dependencies.

4045 **FPT_ITC.1.1**

4046 **The TSF shall protect all TSF data transmitted from the TSF to another trusted IT product**
4047 **from unauthorized disclosure during transmission.**

4048 **14.7 Integrity of exported TSF data (FPT_ITI)**

4049 **14.7.1 Family behaviour**

4050 This family defines the rules for the protection, from unauthorized modification, of TSF data
4051 during transmission between the TSF and another trusted IT product.

4052 **14.7.2 Components leveling and description**

4053 Figure 66 shows the component leveling for this family.



4055 **Figure 66 — FPT_ITI: Component leveling**

4056 FPT_ITI.1 Inter-TSF detection of modification, provides the ability to detect modification of TSF
4057 data during transmission between the TSF and another trusted IT product, under the
4058 assumption that another trusted IT product is cognizant of the mechanism used.

4059 FPT_ITI.2 Inter-TSF detection and correction of modification, provides the ability for another
4060 trusted IT product not only to detect modification, but to correct modified TSF data under the
4061 assumption that another trusted IT product is cognizant of the mechanism used.

4062 **14.7.3 Management of FPT_ITI.1**

4063 The following actions **could** be considered for the management functions in FMT:

4064 a) There are no management activities foreseen.

4065 **14.7.4 Management of FPT_ITI.2**

4066 The following actions **could** be considered for the management functions in FMT:

- 4067 a) Management of the types of TSF data that the TSF tries to correct if modified in
- 4068 transit;
- 4069 b) Management of the types of action that the TSF takes if TSF data is modified in
- 4070 transit.

4071 **14.7.5 Audit of FPT_ITI.1**

4072 The following actions **should** be auditable if FAU_GEN Security audit data generation is included

4073 in the PP/ST:

- 4074 a) Minimal: the detection of modification of transmitted TSF data.
- 4075 b) Basic: the action taken upon detection of modification of transmitted TSF data.

4076 **14.7.6 Audit of FPT_ITI.2**

4077 The following actions **should** be auditable if FAU_GEN Security audit data generation is included

4078 in the PP/ST:

- 4079 a) Minimal: the detection of modification of transmitted TSF data.
- 4080 b) Basic: the action taken upon detection of modification of transmitted TSF data.
- 4081 c) Basic: the use of the correction mechanism.

4082 **14.7.7 FPT_ITI.1 Inter-TSF detection of modification**

4083 **Component relationships**

- 4084 Hierarchical to: No other components.
- 4085 Dependencies: No dependencies.

4086 **FPT_ITI.1.1**

4087 **The TSF shall provide the capability to detect modification of all TSF data during**

4088 **transmission between the TSF and another trusted IT product within the following**

4089 **metric: [assignment: a defined modification metric].**

4090 **FPT_ITI.1.2**

4091 **The TSF shall provide the capability to verify the integrity of all TSF data transmitted**

4092 **between the TSF and another trusted IT product and perform [assignment: action to be**

4093 **taken] if modifications are detected.**

4094 **14.7.8 FPT_ITI.2 Inter-TSF detection and correction of modification**

4095 **Component relationships**

- 4096 Hierarchical to: FPT_ITI.1 Inter-TSF detection of modification
- 4097 Dependencies: No dependencies.

4098 **FPT_ITI.2.1**

4099 The TSF **shall** provide the capability to detect modification of all TSF data during transmission

4100 between the TSF and another trusted IT product within the following metric: [assignment: a

4101 *defined modification metric*].

4102 **FPT_ITI.2.2**

4103 The TSF **shall** provide the capability to verify the integrity of all TSF data transmitted between
 4104 the TSF and another trusted IT product and perform [assignment: *action to be taken*] if
 4105 modifications are detected.

4106 **FPT_ITI.2.3**

4107 **The TSF shall provide the capability to correct [assignment: *type of modification*] of all**
 4108 **TSF data transmitted between the TSF and another trusted IT product.**

4109 **14.8 Internal TOE TSF data transfer (FPT_ITT)**4110 **14.8.1 Family behaviour**

4111 This family provides requirements that address protection of TSF data when it is transferred
 4112 between separate parts of a TOE across an internal channel.

4113 **14.8.2 Components leveling and description**

4114 Figure 67 shows the component leveling for this family.



4115

4116 **Figure 67 — FPT_ITT: Component leveling**

4117 FPT_ITT.1 Basic internal TSF data transfer protection, requires that TSF data be protected when
 4118 transmitted between separate parts of the TOE.

4119 FPT_ITT.2 TSF data transfer separation, requires that the TSF separate user data from TSF data
 4120 during transmission.

4121 FPT_ITT.3 TSF data integrity monitoring, requires that the TSF data transmitted between
 4122 separate parts of the TOE is monitored for identified integrity errors.

4123 **14.8.3 Management of FPT_ITT.1**

4124 The following actions **could** be considered for the management functions in FMT:

- 4125 a) management of the types of modification against which the TSF **should** protect;
- 4126 b) management of the mechanism used to provide the protection of the data in transit
- 4127 between different parts of the TSF.

4128 **14.8.4 Management of FPT_ITT.2**

4129 The following actions **could** be considered for the management functions in FMT:

- 4130 a) management of the types of modification against which the TSF **should** protect;
- 4131 b) management of the mechanism used to provide the protection of the data in transit
- 4132 between different parts of the TSF;
- 4133 c) management of the separation mechanism.

4134 **14.8.5 Management of FPT_ITT.3**

4135 The following actions **could** be considered for the management functions in FMT:

- 4136 a) management of the types of modification against which the TSF **should** protect;

- 4137 b) management of the mechanism used to provide the protection of the data in transit
- 4138 between different parts of the TSF;
- 4139 c) management of the types of modification of TSF data the TSF **should** try to detect;
- 4140 d) management of the actions that will be taken.

4141 **14.8.6 Audit of FPT_ITT.1, FPT_ITT.2**

4142 The following actions **should** be auditable if FAU_GEN Security audit data generation is included

4143 in the PP/ST:

- 4144 a) There are no auditable events foreseen.

4145 **14.8.7 Audit of FPT_ITT.3**

4146 The following actions **should** be auditable if FAU_GEN Security audit data generation is included

4147 in the PP/ST:

- 4148 a) Minimal: the detection of modification of TSF data;
- 4149 b) Basic: the action taken following detection of an integrity error.

4150 **14.8.8 FPT_ITT.1 Basic internal TSF data transfer protection**

4151 **Component relationships**

- 4152 Hierarchical to: No other components.
- 4153 Dependencies: No dependencies.

4154 **FPT_ITT.1.1**

4155 **The TSF **shall** protect TSF data from [selection: *disclosure, modification*] when it is**

4156 **transmitted between separate parts of the TOE.**

4157 **14.8.9 FPT_ITT.2 TSF data transfer separation**

4158 **Component relationships**

- 4159 Hierarchical to: FPT_ITT.1 Basic internal TSF data transfer
- 4160 protection
- 4161 Dependencies: No dependencies.

4162 **FPT_ITT.2.1**

4163 The TSF **shall** protect TSF data from [selection: *disclosure, modification*] when it is transmitted

4164 between separate parts of the TOE.

4165 **FPT_ITT.2.2**

4166 **The TSF **shall** separate user data from TSF data when such data is transmitted between**

4167 **separate parts of the TOE.**

4168 **14.8.10 FPT_ITT.3 TSF data integrity monitoring**

4169 **Component relationships**

- 4170 Hierarchical to: No other components.
- 4171 Dependencies: FPT_ITT.1 Basic internal TSF data transfer
- 4172 protection

4173 **FPT_ITT.3.1**

4174 **The TSF shall be able to detect [selection: modification of data, substitution of data, re-**
 4175 **ordering of data, deletion of data, [assignment: other integrity errors]] for TSF data**
 4176 **transmitted between separate parts of the TOE.**

4177 **FPT_ITT.3.2**

4178 **Upon detection of a data integrity error, the TSF shall take the following actions:**
 4179 **[assignment: specify the action to be taken].**

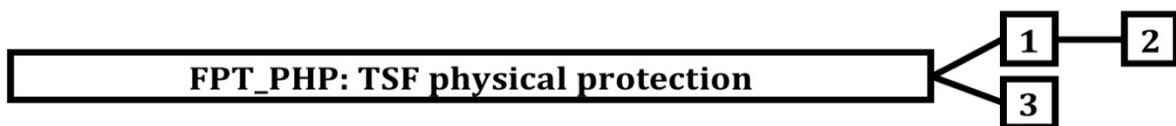
4180 **14.9 TSF physical protection (FPT_PHP)**4181 **14.9.1 Family behaviour**

4182 TSF physical protection components refer to restrictions on unauthorized physical access to the
 4183 TSF, and to the deterrence of, and resistance to, unauthorized physical modification, or
 4184 substitution of the TSF.

4185 The requirements of components in this family ensure that the TSF is protected from physical
 4186 tampering and interference. Satisfying the requirements of these components results in the TSF
 4187 being packaged and used in such a manner that physical tampering is detectable, or resistance
 4188 to physical tampering is enforced. Without these components, the protection functions of a TSF
 4189 lose their effectiveness in environments where physical damage cannot be prevented. This
 4190 family also provides requirements regarding how the TSF shall respond to physical tampering
 4191 attempts.

4192 **14.9.2 Components leveling and description**

4193 Figure 68 shows the component leveling for this family.



4194

4195 **Figure 68 — FPT_PHP: Component leveling**

4196 FPT_PHP.1 Passive detection of physical attack, provides for features that indicate when a TSF
 4197 device or TSF element is subject to tampering. However, notification of tampering is not
 4198 automatic; an authorized user must invoke a security administrative function or perform
 4199 manual inspection to determining if tampering has occurred.

4200 FPT_PHP.2 Notification of physical attack, provides for automatic notification of tampering for
 4201 an identified subset of physical penetrations.

4202 FPT_PHP.3 Resistance to physical attack, provides for features that prevent or resist physical
 4203 tampering with TSF devices and TSF elements.

4204 **14.9.3 Management of FPT_PHP.1**

4205 The following actions could be considered for the management functions in FMT:

4206 a) Management of the user or role that determines whether physical tampering has
 4207 occurred.

4208 **14.9.4 Management of FPT_PHP.2**

4209 The following actions could be considered for the management functions in FMT:

4210 a) Management of the user or role that gets informed about intrusions;

4211 b) Management of the list of devices that **should** inform the indicated user or role
4212 about the intrusion.

4213 **14.9.5 Management of FPT_PHP.3**

4214 The following actions **could** be considered for the management functions in FMT:

4215 a) Management of the automatic responses to physical tampering.

4216 **14.9.6 Audit of FPT_PHP.1**

4217 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4218 in the PP/ST:

4219 a) Minimal: if detection by IT means, detection of intrusion.

4220 **14.9.7 Audit of FPT_PHP.2**

4221 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4222 in the PP/ST:

4223 a) Minimal: detection of intrusion.

4224 **14.9.8 Audit of FPT_PHP.3**

4225 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4226 in the PP/ST:

4227 a) There are no auditable events foreseen.

4228 **14.9.9 FPT_PHP.1 Passive detection of physical attack**

4229 **Component relationships**

4230 Hierarchical to: No other components.

4231 Dependencies: No dependencies.

4232 **FPT_PHP.1.1**

4233 **The TSF **shall** provide unambiguous detection of physical tampering that might**
4234 **compromise the TSF.**

4235 **FPT_PHP.1.2**

4236 **The TSF **shall** provide the capability to determine whether physical tampering with the**
4237 **TSF's devices or TSF's elements has occurred.**

4238 **14.9.10 FPT_PHP.2 Notification of physical attack**

4239 **Component relationships**

4240 Hierarchical to: FPT_PHP.1 Passive detection of physical attack

4241 Dependencies: FMT_LIM.1 Limited capabilities

4242 **FPT_PHP.2.1**

4243 The TSF **shall** provide unambiguous detection of physical tampering that might compromise the
4244 TSF.

4245 **FPT_PHP.2.2**

4246 The TSF **shall** provide the capability to determine whether physical tampering with the TSF's
4247 devices or TSF's elements has occurred.

4248 **FPT_PHP.2.3**

4249 For [assignment: *list of TSF devices/elements for which active detection is required*], the
 4250 TSF **shall** monitor the devices and elements and notify [assignment: *a designated user or*
 4251 *role*] when physical tampering with the TSF's devices or TSF's elements has occurred.

4252 **14.9.11 FPT_PHP.3 Resistance to physical attack**4253 **Component relationships**

4254 Hierarchical to: No other components.

4255 Dependencies: No dependencies.

4256 **FPT_PHP.3.1**

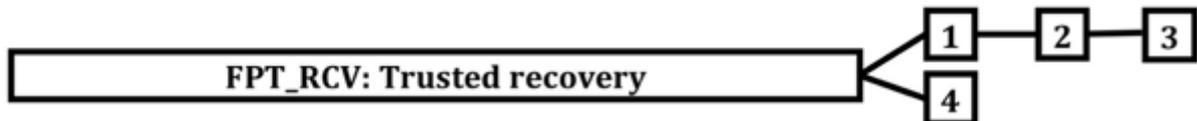
4257 The TSF **shall** resist [assignment: *physical tampering scenarios*] to the [assignment: *list of*
 4258 *TSF devices/elements*] by responding automatically such that the SFRs are always
 4259 **enforced**.

4260 **14.10 Trusted recovery (FPT_RCV)**4261 **14.10.1 Family behaviour**

4262 The requirements of this family ensure that the TSF **can** determine that the TOE is started up
 4263 without protection compromise and **can** recover without protection compromise after
 4264 discontinuity of operations. This family is important because the start-up state of the TSF
 4265 determines the protection of subsequent states.

4266 **14.10.2 Components leveling and description**

4267 Figure 69 shows the component leveling for this family.



4268

4269 **Figure 69 — FPT_RCV: Component leveling**

4270 FPT_RCV.1 Manual recovery, allows a TOE to only provide mechanisms that involve human
 4271 intervention to return to a secure state.

4272 FPT_RCV.2 Automated recovery, provides, for at least one type of service discontinuity,
 4273 recovery to a secure state without human intervention; recovery for other discontinuities that
 4274 **can** require human intervention.

4275 FPT_RCV.3 Automated recovery without undue loss, also provides for automated recovery, but
 4276 strengthens the requirements by disallowing undue loss of protected objects.

4277 FPT_RCV.4 Function recovery, provides for recovery at the level of particular functions,
 4278 ensuring either successful completion or rollback of TSF data to a secure state.

4279 **14.10.3 Management of FPT_RCV.1**

4280 The following actions **could** be considered for the management functions in FMT:

4281 a) Management of who **can** access the restore capability within the maintenance
 4282 mode.

4283 **14.10.4 Management of FPT_RCV.2, FPT_RCV.3**

4284 The following actions **could** be considered for the management functions in FMT:

- 4285 a) Management of who **can** access the restore capability within the maintenance
- 4286 mode;
- 4287 b) Management of the list of failures/service discontinuities that will be handled
- 4288 through the automatic procedures.

4289 **14.10.5 Management of FPT_RCV.4**

4290 The following actions **could** be considered for the management functions in FMT:

- 4291 a) There are no management activities foreseen.

4292 **14.10.6 Audit of FPT_RCV.1, FPT_RCV.2, FPT_RCV.3**

4293 The following actions **should** be auditable if FAU_GEN Security audit data generation is included

4294 in the PP/ST:

- 4295 a) Minimal: the fact that a failure or service discontinuity occurred;
- 4296 b) Minimal: resumption of the regular operation;
- 4297 c) Basic: type of failure or service discontinuity.

4298 **14.10.7 Audit of FPT_RCV.4**

4299 The following actions **should** be auditable if FAU_GEN Security audit data generation is included

4300 in the PP/ST:

- 4301 a) Minimal: if possible, the impossibility to return to a secure state after a failure of
- 4302 the TSF;
- 4303 b) Basic: if possible, the detection of a failure of a function.

4304 **14.10.8 FPT_RCV.1 Manual recovery**

4305 **Component relationships**

- 4306 Hierarchical to: No other components.
- 4307 Dependencies: AGD_OPE.1 Operational user guidance

4308 **FPT_RCV.1.1**

4309 **After [assignment: *list of failures/service discontinuities*] the TSF **shall** enter a**

4310 **maintenance mode where the ability to return to a secure state is provided.**

4311 **14.10.9 FPT_RCV.2 Automated recovery**

4312 **Component relationships**

- 4313 Hierarchical to: FPT_RCV.1 Manual recovery
- 4314 Dependencies: AGD_OPE.1 Operational user guidance

4315 **FPT_RCV.2.1**

4316 **When automated recovery from [assignment: *list of failures/service discontinuities*] is not**

4317 **possible, the TSF **shall** enter a maintenance mode where the ability to return to a secure state is**

4318 **provided.**

4319 **FPT_RCV.2.2**

4320 For [assignment: *list of failures/service discontinuities*], the TSF **shall** ensure the return of
4321 the TOE to a secure state using automated procedures.

4322 **14.10.10 FPT_RCV.3 Automated recovery without undue loss**4323 **Component relationships**

4324 Hierarchical to: FPT_RCV.2 Automated recovery

4325 Dependencies: AGD_OPE.1 Operational user guidance

4326 **FPT_RCV.3.1**

4327 When automated recovery from [assignment: *list of failures/service discontinuities*] is not
4328 possible, the TSF **shall** enter a maintenance mode where the ability to return to a secure state is
4329 provided.

4330 **FPT_RCV.3.2**

4331 For [assignment: *list of failures/service discontinuities*], the TSF **shall** ensure the return of the
4332 TOE to a secure state using automated procedures.

4333 **FPT_RCV.3.3**

4334 **The functions provided by the TSF to recover from failure or service discontinuity shall**
4335 **ensure that the secure initial state is restored without exceeding [assignment:**
4336 ***quantification*] for loss of TSF data or objects under the control of the TSF.**

4337 **FPT_RCV.3.4**

4338 **The TSF shall provide the capability to determine the objects that were or were not**
4339 **capable of being recovered.**

4340 **14.10.11 FPT_RCV.4 Function recovery**4341 **Component relationships**

4342 Hierarchical to: No other components.

4343 Dependencies: No dependencies.

4344 **FPT_RCV.4.1**

4345 **The TSF shall ensure that [assignment: *list of functions and failure scenarios*] have the**
4346 **property that the function either completes successfully, or for the indicated failure**
4347 **scenarios, recovers to a consistent and secure state.**

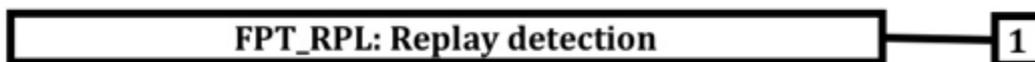
4348 **14.11 Replay detection (FPT_RPL)**4349 **14.11.1 Family behaviour**

4350 This family addresses detection of replay for various types of entities and subsequent actions to
4351 correct. In the case where replay **may** be detected, this effectively prevents it.

4352 **14.11.2 Components leveling and description**

4353 Figure 70 shows the component leveling for this family.

4354



4355 **Figure 70 — FPT_RPL: Component leveling**

4356 The family consists of only one component, FPT_RPL.1 Replay detection, which requires that
 4357 the TSF **shall** be able to detect the replay of identified entities.

4358 **14.11.3 Management of FPT_RPL.1**

4359 The following actions **could** be considered for the management functions in FMT:

- 4360 a) Management of the list of identified entities for which replay is detected;
- 4361 b) Management of the list of actions that need to be taken in case of replay.

4362 **14.11.4 Audit of FPT_RPL.1**

4363 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 4364 in the PP/ST:

- 4365 a) Basic: Detected replay attacks.
- 4366 b) Detailed: Action to be taken based on the specific actions.

4367 **14.11.5 FPT_RPL.1 Replay detection**

4368 **Component relationships**

4369	Hierarchical to:	No other components.
4370	Dependencies:	No dependencies.

4371 **FPT_RPL.1.1**

4372 **The TSF **shall** detect replay for the following entities: [assignment: *list of identified***
 4373 ***entities*].**

4374 **FPT_RPL.1.2**

4375 **The TSF **shall** perform [assignment: *list of specific actions*] when replay is detected.**

4376 **14.12 State synchrony protocol (FPT_SSP)**

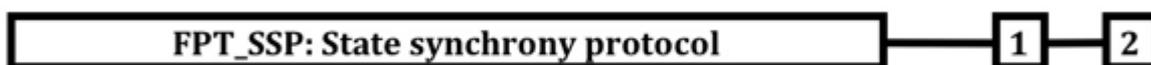
4377 **14.12.1 Family behaviour**

4378 Distributed TOEs **can** give rise to greater complexity than monolithic TOEs through the
 4379 potential for differences in state between parts of the TOE, and through delays in
 4380 communication. In most cases synchronization of state between distributed functions involves
 4381 an exchange protocol, not a simple action. When malice exists in the distributed environment of
 4382 these protocols, more complex defensive protocols are required.

4383 State synchrony protocol (FPT_SSP) establishes the requirement for certain critical functions of
 4384 the TSF to use this trusted protocol. State synchrony protocol (FPT_SSP) ensures that two
 4385 distributed parts of the TOE have synchronized their states after a security-relevant action.

4386 **14.12.2 Components leveling and description**

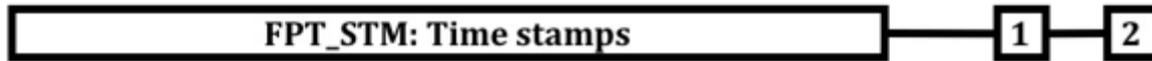
4387 Figure 71 shows the component leveling for this family.



4388

4389 **Figure 71 — FPT_SSP: Component leveling**

- 4390 FPT_SSP.1 Simple trusted acknowledgement, requires only a simple acknowledgment by the
4391 data recipient.
- 4392 FPT_SSP.2 Mutual trusted acknowledgement, requires mutual acknowledgment of the data
4393 exchange.
- 4394 **14.12.3 Management of FPT_SSP.1, FPT_SSP.2**
- 4395 The following actions **could** be considered for the management functions in FMT:
- 4396 a) There are no management activities foreseen.
- 4397 **14.12.4 Audit of FPT_SSP.1, FPT_SSP.2**
- 4398 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4399 in the PP/ST:
- 4400 a) Minimal: failure to receive an acknowledgement when expected.
- 4401 **14.12.5 FPT_SSP.1 Simple trusted acknowledgement**
- 4402 **Component relationships**
- | | | |
|------|------------------|--|
| 4403 | Hierarchical to: | No other components. |
| 4404 | Dependencies: | FPT_ITT.1 Basic internal TSF data transfer |
| 4405 | | protection |
- 4406 **FPT_SSP.1.1**
- 4407 **The TSF shall acknowledge, when requested by another part of the TSF, the receipt of an**
4408 **unmodified TSF data transmission.**
- 4409 **14.12.6 FPT_SSP.2 Mutual trusted acknowledgement**
- 4410 **Component relationships**
- | | | |
|------|------------------|--|
| 4411 | Hierarchical to: | FPT_SSP.1 Simple trusted acknowledgement |
| 4412 | Dependencies: | FPT_ITT.1 Basic internal TSF data transfer |
| 4413 | | protection |
- 4414 **FPT_SSP.2.1**
- 4415 The TSF **shall** acknowledge, when requested by another part of the TSF, the receipt of an
4416 unmodified TSF data transmission.
- 4417 **FPT_SSP.2.2**
- 4418 **The TSF shall ensure that the relevant parts of the TSF know the correct status of**
4419 **transmitted data among its different parts, using acknowledgements.**
- 4420 **14.13 Time stamps (FPT_STM)**
- 4421 **14.13.1 Family behaviour**
- 4422 This family addresses requirements for a reliable time stamp function within a TOE.
- 4423 **14.13.2 Components leveling and description**
- 4424 Figure 72 shows the component leveling for this family.



4425

4426

Figure 72 — FPR_STM: Component leveling

4427 FPT_STM.1 Reliable time stamps, requires that the TSF provide reliable time stamps for TSF
4428 functions.

4429 FPT_STM.2 Time source, requires the description of the time source used in timestamps

4430 **14.13.3 Management of FPT_STM.1**

4431 The following actions **could** be considered for the management functions in FMT:

- 4432 a) Management of the time.

4433 **14.13.4 Management of FPT_STM.2**

4434 The following actions **could** be considered for the management functions in FMT:

- 4435 a) Setting of time by user authorized according to security policy.

4436 **14.13.5 Audit of FPT_STM.1**

4437 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4438 in the PP/ST:

- 4439 a) Minimal: changes to the time.
4440 b) Detailed: providing a timestamp.

4441 **14.13.6 Audit of FPT_STM.2**

4442 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4443 in the PP/ST:

- 4444 a) Minimal: discontinuous changes to the time;
4445 b) Detailed: changes to the time source.

4446 **14.13.7 FPT_STM.1 Reliable time stamps**

4447 **Component relationships**

- 4448 Hierarchical to: No other components.
4449 Dependencies: No dependencies.

4450 **FPT_STM.1.1**

4451 **The TSF shall be able to provide reliable time stamps.**

4452 **14.13.8 FPT_STM.2 Time source**

4453 **Component relationships**

- 4454 Hierarchical to: No other components.
4455 Dependencies: FPT_STM.1 Reliable time stamps
4456 FMT_SMR.1 Security roles

4457 **FPT_STM.2.1**

4458 The TSF **shall** allow the [assignment: *user authorized by security policy*] to [assignment:
4459 *set the time, configure another time source*].

4460 **14.14 Inter-TSF TSF data consistency (FPT_TDC)**4461 **14.14.1 Family behaviour**

4462 In a distributed environment, a TOE **may** need to exchange TSF data with another trusted IT
4463 product. This family defines the requirements for sharing and consistent interpretation of these
4464 attributes between the TSF of the TOE and a different trusted IT product.

4465 **14.14.2 Components leveling and description**

4466 Figure 73 shows the component leveling for this family.



4467

4468 **Figure 73 — FPT_TDC: Component leveling**

4469 FPT_TDC.1 Inter-TSF basic TSF data consistency, requires that the TSF provide the capability to
4470 ensure consistency of attributes between TSFs.

4471 **14.14.3 Management of FPT_TDC.1**

4472 The following actions **could** be considered for the management functions in FMT:

4473 a) There are no management activities foreseen.

4474 **14.14.4 Audit of FPT_TDC.1**

4475 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4476 in the PP/ST:

4477 a) Minimal: Successful use of TSF data consistency mechanisms.

4478 b) Basic: Use of the TSF data consistency mechanisms.

4479 c) Basic: Identification of which TSF data have been interpreted.

4480 d) Basic: Detection of modified TSF data.

4481 **14.14.5 FPT_TDC.1 Inter-TSF basic TSF data consistency**4482 **Component relationships**

4483 Hierarchical to: No other components.

4484 Dependencies: No dependencies.

4485 **FPT_TDC.1.1**

4486 The TSF **shall** provide the capability to consistently interpret [assignment: *list of TSF data*
4487 *types*] when shared between the TSF and another trusted IT product.

4488 **FPT_TDC.1.2**

4489 The TSF **shall** use [assignment: *list of interpretation rules to be applied by the TSF*] when
4490 interpreting the TSF data from another trusted IT product.

4491 **14.15 Testing of external entities (FPT_TEE)**

4492 **14.15.1 Family behaviour**

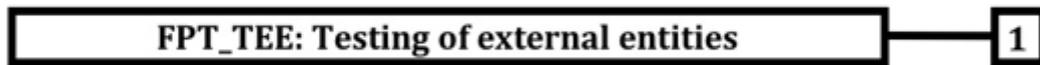
4493 This family defines requirements for the TSF to perform tests on one or more external entities.

4494 This component is not intended to be applied to human users.

4495 External entities **can** include applications running on the TOE, hardware or software running
 4496 “underneath” the TOE (platforms, operating systems etc.) or applications/boxes connected to
 4497 the TOE (intrusion detection systems, firewalls, login servers, time servers etc.).

4498 **14.15.2 Components leveling and description**

4499 Figure 74 shows the component leveling for this family.



4500

4501 **Figure 74 — FPT_TEE: Component leveling**

4502 FPT_TEE.1 Testing of external entities, provides for testing of the external entities by the TSF.

4503 **14.15.3 Management of FPT_TEE.1**

4504 The following actions **could** be considered for the management functions in FMT:

- 4505 a) Management of the conditions under which the testing of external entities occurs,
 4506 such as during initial start-up, regular interval, or under specified conditions;
- 4507 b) Management of the time interval if appropriate.

4508 **14.15.4 Audit of FPT_TEE.1**

4509 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 4510 in the PP/ST:

- 4511 a) Basic: Execution of the tests of the external entities and the results of the tests.

4512 **14.15.5 FPT_TEE.1 Testing of external entities**

4513 **Component relationships**

4514 Hierarchical to: No other components.

4515 Dependencies: No dependencies.

4516 **FPT_TEE.1.1**

4517 **The TSF shall run a suite of tests [selection: *during initial start-up, periodically during***
 4518 ***normal operation, at the request of an authorized user, [assignment: other conditions]] to***
 4519 **check the fulfillment of [assignment: *list of properties of the external entities*].**

4520 **FPT_TEE.1.2**

4521 **If the test fails, the TSF shall [assignment: *action(s)*].**

4522 **14.16 Internal TOE TSF data replication consistency (FPT_TRC)**

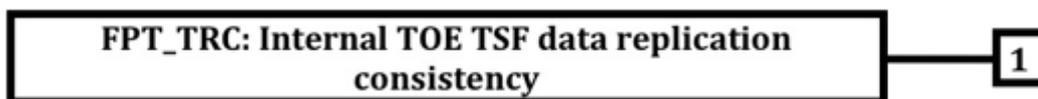
4523 **14.16.1 Family behaviour**

4524 The requirements of this family are needed to ensure the consistency of TSF data when such
 4525 data is replicated internal to the TOE. Such data **may** become inconsistent if the internal channel

4526 between parts of the TOE becomes inoperative. If the TOE is internally structured as a network
 4527 and parts of the TOE network connections are broken, this **may** occur when parts become
 4528 disabled.

4529 14.16.2 Components leveling and description

4530 Figure 75 shows the component leveling for this family.



4531

4532 **Figure 75 — FPT_TRC: Component leveling**

4533 This family consists of only one component, FPT_TRC.1 Internal TSF consistency, which
 4534 requires that the TSF ensure the consistency of TSF data that is replicated in multiple locations.

4535 14.16.3 Management of FPT_TRC.1

4536 The following actions **could** be considered for the management functions in FMT:

4537 a) There are no management activities foreseen.

4538 14.16.4 Audit of FPT_TRC.1

4539 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 4540 in the PP/ST:

4541 a) Minimal: restoring consistency upon reconnection;

4542 b) Basic: Detected inconsistency between TSF data.

4543 14.16.5 FPT_TRC.1 Internal TSF consistency

4544 Component relationships

4545 Hierarchical to: No other components.

4546 Dependencies: FPT_ITT.1 Basic internal TSF data transfer
 4547 protection

4548 FPT_TRC.1.1

4549 The TSF **shall** ensure that TSF data is consistent when replicated between parts of the
 4550 TOE.

4551 FPT_TRC.1.2

4552 When parts of the TOE containing replicated TSF data are disconnected, the TSF **shall**
 4553 ensure the consistency of the replicated TSF data upon reconnection before processing
 4554 any requests for [assignment: *list of functions dependent on TSF data replication*
 4555 *consistency*].

4556 14.17 TSF self-test (FPT_TST)

4557 14.17.1 Family behaviour

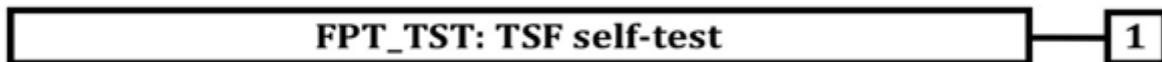
4558 The family defines the requirements for the self-testing of the TSF with respect to some
 4559 expected correct operation. Examples are interfaces to enforcement functions, and sample
 4560 arithmetical operations on critical parts of the TOE. These tests **can** be carried out at start-up,

4561 periodically, at the request of the authorized user, or when other conditions are met. The
 4562 actions to be taken by the TOE as the result of self-testing are defined in other families.

4563 The requirements of this family are also needed to detect the corruption of TSF data and TSF
 4564 itself (i.e. TSF executable code or TSF hardware component) by various failures that do not
 4565 necessarily stop the TOE's operation (which would be handled by other families). These checks
 4566 must be performed because these failures **cannot** necessarily be prevented. Such failures **can**
 4567 occur either because of unforeseen failure modes or associated oversights in the design of
 4568 hardware, firmware, or software, or because of malicious corruption of the TSF due to
 4569 inadequate logical and/or physical protection.

4570 **14.17.2 Components leveling and description**

4571 Figure 76 shows the component leveling for this family.



4572

4573 **Figure 76 — FPT_TST: Component leveling**

4574 FPT_TST.1 TSF self-testing, provides the ability to test the TSF's correct operation. These tests
 4575 **can** be performed at start-up, periodically, at the request of the authorized user, or when other
 4576 conditions are met. It also provides the ability to verify the integrity of TSF data and TSF itself.

4577 **14.17.3 Management of FPT_TST.1**

4578 The following actions **could** be considered for the management functions in FMT:

- 4579 a) Management of the conditions under which TSF self-testing occurs, such as during
- 4580 initial start-up, regular interval, or under specified conditions;
- 4581 b) Management of the time interval if appropriate.

4582 **14.17.4 Audit of FPT_TST.1**

4583 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 4584 in the PP/ST:

- 4585 a) Minimal: Indication that the TSF self-tests were completed and any failures of the
- 4586 tests.
- 4587 b) Basic: Execution of the TSF self-tests and the results of the tests.

4588 **14.17.5 FPT_TST.1 TSF self-testing**

4589 **Component relationships**

4590 Hierarchical to: No other components.

4591 Dependencies: No dependencies.

4592 **FPT_TST.1.1**

4593 **The TSF shall run a suite of the following self-tests [selection: *during initial start-up,***
 4594 ***periodically during normal operation, at the request of the authorized user, at the***
 4595 ***conditions [assignment: conditions under which self-test should occur]] to demonstrate the***
 4596 ***correct operation of [selection: [assignment: parts of TSF], the TSF]: [assignment: list of***
 4597 ***self-tests run by the TSF].***

4598 **FPT_TST.1.2**

4599 **The TSF shall provide authorized users with the capability to verify the integrity of**
 4600 **[selection: [assignment: parts of TSF data], TSF data].**

4601 **FPT_TST.1.3**

4602 **The TSF shall provide authorized users with the capability to verify the integrity of**
4603 **[selection: *[assignment: parts of TSF], TSF*].**

4604

4605 **15 Class FRU: Resource utilization**

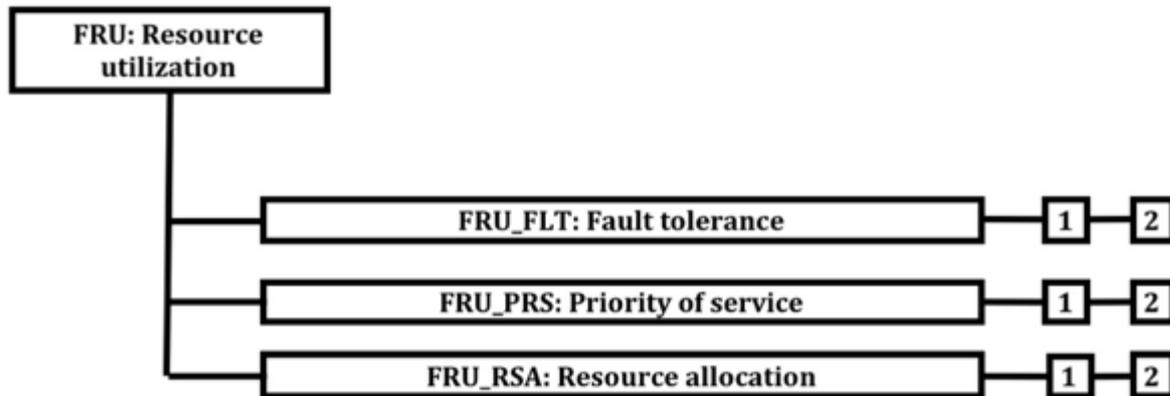
4606 **15.1 Class description**

4607 This class provides three families that support the availability of required resources such as
 4608 processing capability and/or storage capacity. The family Fault Tolerance provides protection
 4609 against unavailability of capabilities caused by failure of the TOE. The family Priority of Service
 4610 ensures that the resources will be allocated to the more important or time-critical tasks and
 4611 cannot be monopolized by lower priority tasks. The family Resource Allocation provides limits
 4612 on the use of available resources, therefore preventing users from monopolizing the resources.

4613 Figure 77 shows the decomposition of this class, it's families and components. Elements are not
 4614 shown in the figure.

4615 Annex K provides explanatory information for this class and **should** be consulted when using
 4616 the components identified in this class.

4617
 4618



4619 **Figure 77 — FRU: Resource utilization class decomposition**

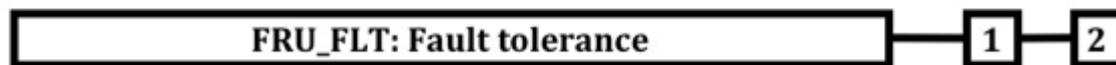
4620 **15.2 Fault tolerance (FRU_FLT)**

4621 **15.2.1 Family behaviour**

4622 The requirements of this family ensure that the TOE will maintain correct operation even in the
 4623 event of failures.

4624 **15.2.2 Components leveling and description**

4625 Figure 78 shows the component leveling for this family.



4626
 4627 **Figure 78 — FRU_FLT: Component leveling**

4628 FRU_FLT.1 Degraded fault tolerance, requires the TOE to continue correct operation of
 4629 identified capabilities in the event of identified failures.

4630 FRU_FLT.2 Limited fault tolerance, requires the TOE to continue correct operation of all
 4631 capabilities in the event of identified failures.

4632 **15.2.3 Management of FRU_FLT.1, FRU_FLT.2**4633 The following actions **could** be considered for the management functions in FMT:

4634 a) There are no management activities foreseen.

4635 **15.2.4 Audit of FRU_FLT.1**4636 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4637 in the PP/ST:

4638 a) Minimal: Any failure detected by the TSF.

4639 b) Basic: All TOE capabilities being discontinued due to a failure.

4640 **15.2.5 Audit of FRU_FLT.2**4641 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4642 in the PP/ST:

4643 a) Minimal: Any failure detected by the TSF.

4644 **15.2.6 FRU_FLT.1 Degraded fault tolerance**4645 **Component relationships**

4646 Hierarchical to: No other components.

4647 Dependencies: FPT_FLS.1 Failure with preservation of secure state

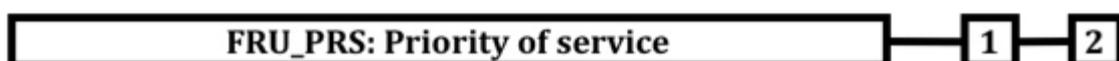
4648 **FRU_FLT.1.1**4649 **The TSF shall ensure the operation of [assignment: *list of TOE capabilities*] when the**
4650 **following failures occur: [assignment: *list of type of failures*].**4651 **15.2.7 FRU_FLT.2 Limited fault tolerance**4652 **Component relationships**

4653 Hierarchical to: FRU_FLT.1 Degraded fault tolerance

4654 Dependencies: FPT_FLS.1 Failure with preservation of secure state

4655 **FRU_FLT.2.1**4656 The TSF **shall** ensure the operation of **all the TOE's capabilities** when the following failures
4657 occur: [assignment: *list of type of failures*].4658 **15.3 Priority of service (FRU_PRS)**4659 **15.3.1 Family behaviour**4660 The requirements of this family allow the TSF to control the use of resources under the control
4661 of the TSF by users and subjects such that high priority activities under the control of the TSF
4662 will always be accomplished without undue interference or delay caused by low priority
4663 activities.4664 **15.3.2 Components leveling and description**

4665 Figure 79 shows the component leveling for this family.



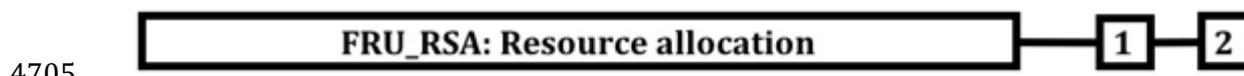
4666

Figure 79 — FRU_PRS: Component leveling

- 4667
- 4668 FRU_PRS.1 Limited priority of service, provides priorities for a subject's use of a subset of the
4669 resources under the control of the TSF.
- 4670 FRU_PRS.2 Full priority of service, provides priorities for a subject's use of all of the resources
4671 under the control of the TSF.
- 4672 **15.3.3 Management of FRU_PRS.1, FRU_PRS.2**
- 4673 The following actions **could** be considered for the management functions in FMT:
- 4674 a) Assignment of priorities to each subject in the TSF.
- 4675 **15.3.4 Audit of FRU_PRS.1, FRU_PRS.2**
- 4676 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4677 in the PP/ST:
- 4678 a) Minimal: Rejection of operation based on the use of priority within an allocation.
- 4679 b) Basic: All attempted uses of the allocation function which involves the priority of
4680 the service functions.
- 4681 **15.3.5 FRU_PRS.1 Limited priority of service**
- 4682 Hierarchical to: No other components.
- 4683 Dependencies: No dependencies.
- 4684 **FRU_PRS.1.1**
- 4685 **The TSF shall assign a priority to each subject in the TSF.**
- 4686 **FRU_PRS.1.2**
- 4687 **The TSF shall ensure that each access to [assignment: *controlled resources*] shall be**
4688 **mediated on the basis of the subjects assigned priority.**
- 4689 **15.3.6 FRU_PRS.2 Full priority of service**
- 4690 **Component relationships**
- 4691 Hierarchical to: FRU_PRS.1 Limited priority of service
- 4692 Dependencies: No dependencies.
- 4693 **FRU_PRS.2.1**
- 4694 The TSF **shall** assign a priority to each subject in the TSF.
- 4695 **FRU_PRS.2.2**
- 4696 The TSF **shall** ensure that each access to **all shareable resources shall** be mediated on the
4697 basis of the subjects assigned priority.
- 4698 **15.4 Resource allocation (FRU_RSA)**
- 4699 **15.4.1 Family behaviour**
- 4700 The requirements of this family allow the TSF to control the use of resources by users and
4701 subjects such that denial of service will not occur because of unauthorized monopolization of
4702 resources.

4703 **15.4.2 Components leveling and description**

4704 Figure 80 shows the component leveling for this family.



4705

4706 **Figure 80 — FRU_RSA: Component leveling**4707 FRU_RSA.1 Maximum quotas, provides requirements for quota mechanisms that ensure that
4708 users and subjects will not monopolize a controlled resource.4709 FRU_RSA.2 Minimum and maximum quotas, provides requirements for quota mechanisms that
4710 ensure that users and subjects will always have at least a minimum of a specified resource and
4711 that they will not be able to monopolize a controlled resource.4712 **15.4.3 Management of FRU_RSA.1**4713 The following actions **could** be considered for the management functions in FMT:

- 4714 a) Specifying maximum limits for a resource for groups and/or individual users
-
- 4715 and/or subjects by an administrator.

4716 **15.4.4 Management of FRU_RSA.2**4717 The following actions **could** be considered for the management functions in FMT:

- 4718 a) Specifying minimum and maximum limits for a resource for groups and/or
-
- 4719 individual users and/or subjects by an administrator.

4720 **15.4.5 Audit of FRU_RSA.1, FRU_RSA.2**4721 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4722 in the PP/ST:

- 4723 a) Minimal: Rejection of allocation operation due to resource limits.
-
- 4724 b) Basic: All attempted uses of the resource allocation functions for resources that are
-
- 4725 under control of the TSF.

4726 **15.4.6 FRU_RSA.1 Maximum quotas**4727 **Component relationships**

4728 Hierarchical to: No other components.

4729 Dependencies: No dependencies.

4730 **FRU_RSA.1.1**4731 **The TSF shall enforce maximum quotas of the following resources: [assignment:**
4732 **controlled resources] that [selection: individual user, defined group of users, subjects] can**
4733 **use [selection: simultaneously, over a specified period of time].**4734 **15.4.7 FRU_RSA.2 Minimum and maximum quotas**4735 **Component relationships**

4736 Hierarchical to: FRU_RSA.1 Maximum quotas

4737 Dependencies: No dependencies.

4738 **FRU_RSA.2.1**

4739 The TSF **shall** enforce maximum quotas of the following resources [assignment: *controlled*
4740 *resources*] that [selection: *individual user, defined group of users, subjects*] **can** use [selection:
4741 *simultaneously, over a specified period of time*].

4742 **FRU_RSA.2.2**

4743 **The TSF shall ensure the provision of minimum quantity of each [assignment: *controlled***
4744 ***resource*] that is available for [selection: *an individual user, defined group of users,***
4745 ***subjects*] to use [selection: *simultaneously, over a specified period of time*].**

4746

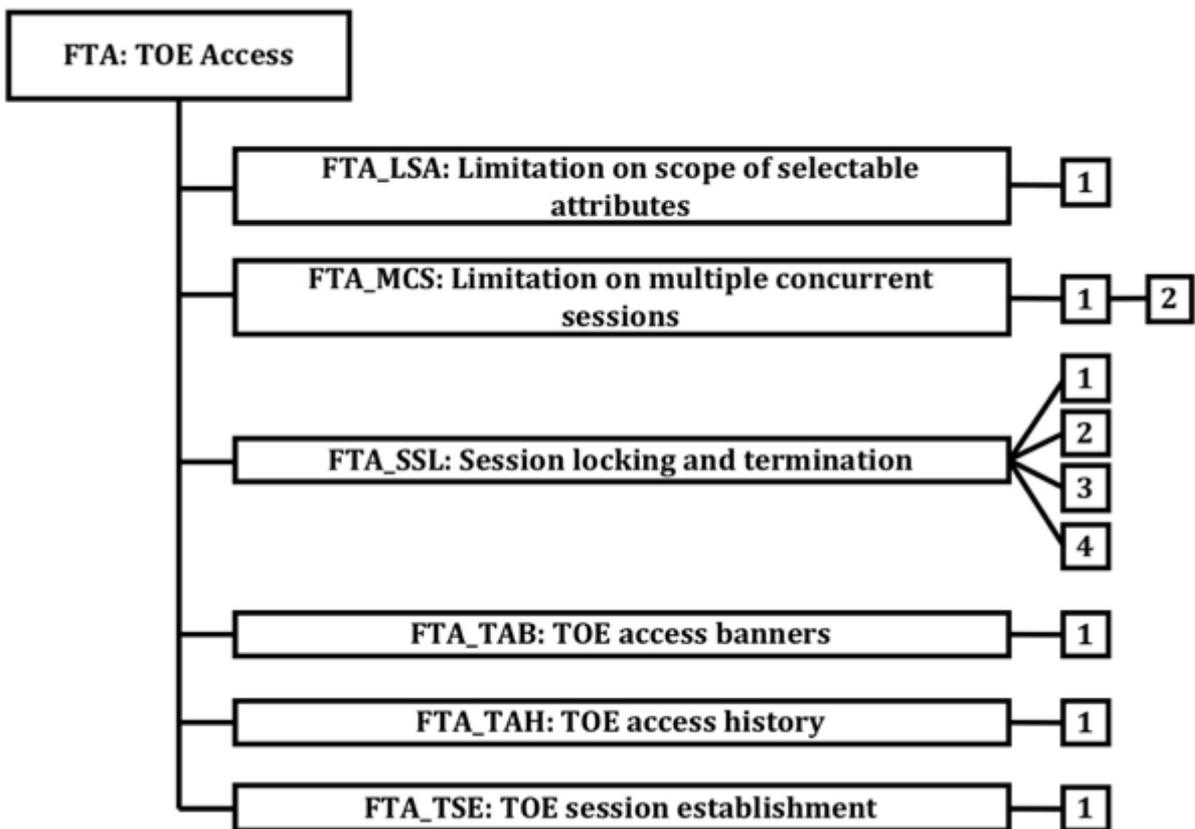
4747 **16 Class FTA: TOE access**

4748 **16.1 Class description**

4749 This family specifies functional requirements for controlling the establishment of a user's
4750 session.

4751 Figure 81 shows the decomposition of this class, it's families and components. Elements are not
4752 shown in the figure.

4753 Annex L provides explanatory information for this class and **should** be consulted when using
4754 the components identified in this class.



4755 **Figure 81 — FTA: TOE access class decomposition**

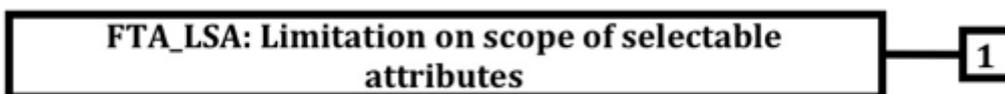
4756 **16.2 Limitation on scope of selectable attributes (FTA_LSA)**

4757 **16.2.1 Family behaviour**

4758 This family defines requirements to limit the scope of session security attributes that a user **can**
4759 select for a session.

4760 **16.2.2 Components leveling and description**

4761 Figure 82 shows the component leveling for this family.



4762

4763 **Figure 82 — FTA_LSA: Component leveling**

4764 FTA_LSA.1 Limitation on scope of selectable attributes, provides the requirement for a TOE to
 4765 limit the scope of the session security attributes during session establishment.

4766 **16.2.3 Management of FTA_LSA.1**

4767 The following actions **could** be considered for the management functions in FMT:

- 4768 a) Management of the scope of the session security attributes by an administrator.

4769 **16.2.4 Audit of FTA_LSA.1**

4770 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 4771 in the PP/ST:

- 4772 a) Minimal: All failed attempts at selecting session security attributes.
- 4773 b) Basic: All attempts at selecting session security attributes.
- 4774 c) Detailed: Capture of the values of each of the session security attributes.

4775 **16.2.5 FTA_LSA.1 Limitation on scope of selectable attributes**

4776 **Component relationships**

4777	Hierarchical to:	No other components.
4778	Dependencies:	No dependencies.

4779 **FTA_LSA.1.1**

4780 **The TSF shall restrict the scope of the session security attributes [assignment: session**
 4781 **security attributes], based on [assignment: attributes].**

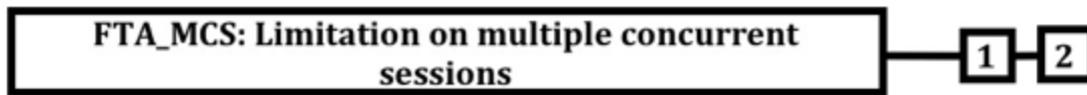
4782 **16.3 Limitation on multiple concurrent sessions (FTA_MCS)**

4783 **16.3.1 Family behaviour**

4784 This family defines requirements to place limits on the number of concurrent sessions that
 4785 belong to the same user.

4786 **16.3.2 Components leveling and description**

4787 Figure 83 shows the component leveling for this family.



4788

4789 **Figure 83 — FTA_MCS: Component leveling**

4790 FTA_MCS.1 Basic limitation on multiple concurrent sessions, provides limitations that apply to
 4791 all users of the TSF.

4792 FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions extends FTA_MCS.1
 4793 Basic limitation on multiple concurrent sessions by requiring the ability to specify limitations
 4794 on the number of concurrent sessions based on the related security attributes.

4795 **16.3.3 Management of FTA_MCS.1**

4796 The following actions **could** be considered for the management functions in FMT:

- 4797 a) Management of the maximum allowed number of concurrent user sessions by an
 4798 administrator.

4799 **16.3.4 Management of FTA_MCS.2**4800 The following actions **could** be considered for the management functions in FMT:

- 4801 a) Management of the rules that govern the maximum allowed number of concurrent
-
- 4802 user sessions by an administrator.

4803 **16.3.5 Audit of FTA_MCS.1, FTA_MCS.2**4804 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4805 in the PP/ST:

- 4806 a) Minimal: Rejection of a new session based on the limitation of multiple concurrent
-
- 4807 sessions.

- 4808 b) Detailed: Capture of the number of currently concurrent user sessions and the user
-
- 4809 security attribute(s).

4810 **16.3.6 FTA_MCS.1 Basic limitation on multiple concurrent sessions**4811 **Component relationships**

4812 Hierarchical to: No other components.

4813 Dependencies: FIA_UID.1 Timing of identification

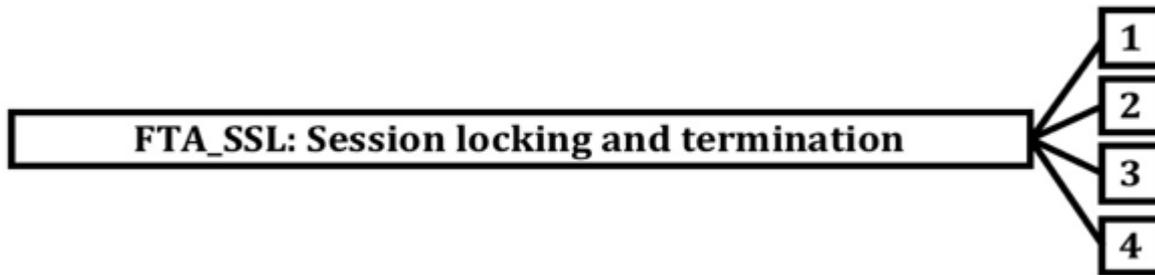
4814 **FTA_MCS.1.1**4815 The TSF **shall** restrict the maximum number of concurrent sessions that belong to the
4816 same user.4817 **FTA_MCS.1.2**4818 The TSF **shall** enforce, by default, a limit of [assignment: *default number*] sessions per
4819 user.4820 **16.3.7 FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions**4821 **Component relationships**4822 Hierarchical to: FTA_MCS.1 Basic limitation on multiple concurrent
4823 sessions

4824 Dependencies: FIA_UID.1 Timing of identification

4825 **FTA_MCS.2.1**4826 The TSF **shall** restrict the maximum number of concurrent sessions that belong to the same user
4827 according to the rules [assignment: *rules for the number of maximum concurrent*
4828 *sessions*].4829 **FTA_MCS.2.2**4830 The TSF **shall** enforce, by default, a limit of [assignment: *default number*] sessions per user.4831 **16.4 Session locking and termination (FTA_SSL)**4832 **16.4.1 Family behaviour**4833 This family defines requirements for the TSF to provide the capability for TSF-initiated and
4834 user-initiated locking, unlocking, and termination of interactive sessions.

4835 **16.4.2 Components leveling and description**

4836 Figure 84 shows the component leveling for this family.



4837

4838 **Figure 84 — FTA_SSL: Component leveling**

4839 FTA_SSL.1 TSF-initiated session locking includes system-initiated locking of an interactive
4840 session after a specified period of user inactivity.

4841 FTA_SSL.2 User-initiated locking, provides capabilities for the user to lock and unlock the user's
4842 own interactive sessions.

4843 FTA_SSL.3 TSF-initiated termination, provides requirements for the TSF to terminate the
4844 session after a specified period of user inactivity.

4845 FTA_SSL.4 User-initiated termination, provides capabilities for the user to terminate the user's
4846 own interactive sessions.

4847 **16.4.3 Management of FTA_SSL.1**

4848 The following actions **could** be considered for the management functions in FMT:

- 4849 a) Specification of the time of user inactivity after which lock-out occurs for an
4850 individual user;
- 4851 b) Specification of the default time of user inactivity after which lock-out occurs;
- 4852 c) Management of the events that occur prior to unlocking the session.

4853 **16.4.4 Management of FTA_SSL.2**

4854 The following actions **could** be considered for the management functions in FMT:

- 4855 a) Management of the events that occur prior to unlocking the session.

4856 **16.4.5 Management of FTA_SSL.3**

4857 The following actions **could** be considered for the management functions in FMT:

- 4858 a) Specification of the time of user inactivity after which termination of the interactive
4859 session occurs for an individual user;
- 4860 b) Specification of the default time of user inactivity after which termination of the
4861 interactive session occurs.

4862 **16.4.6 Management of FTA_SSL.4**

4863 The following actions **could** be considered for the management functions in FMT:

- 4864 a) There are no management activities foreseen.

4865 **16.4.7 Audit of FTA_SSL.1, FTA_SSL.2**

4866 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4867 in the PP/ST:

- 4868 a) Minimal: Locking of an interactive session by the session locking mechanism.
 4869 b) Minimal: Successful unlocking of an interactive session.
 4870 c) Basic: Any attempts at unlocking an interactive session.

4871 16.4.8 Audit of FTA_SSL.3

4872 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 4873 in the PP/ST:

- 4874 a) Minimal: Termination of an interactive session by the session locking mechanism.

4875 16.4.9 Audit of FTA_SSL.4

4876 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 4877 in the PP/ST:

- 4878 a) Minimal: Termination of an interactive session by the user.

4879 16.4.10 FTA_SSL.1 TSF-initiated session locking

4880 Component relationships

4881 Hierarchical to: No other components.

4882 Dependencies: FIA_UAU.1 Timing of authentication

4883 FTA_SSL.1.1

4884 The TSF **shall** lock an interactive session after [assignment: *time interval of user*
 4885 *inactivity*] by:

- 4886 a) clearing or overwriting display devices, making the current contents
 4887 unreadable;
 4888 b) disabling any activity of the user's data access/display devices other than
 4889 unlocking the session.

4890 FTA_SSL.1.2

4891 The TSF **shall** require the following events to occur prior to unlocking the session:
 4892 [assignment: *events to occur*].

4893 16.4.11 FTA_SSL.2 User-initiated locking

4894 Component relationships

4895 Hierarchical to: No other components.

4896 Dependencies: FIA_UAU.1 Timing of authentication

4897 FTA_SSL.2.1

4898 The TSF **shall** allow user-initiated locking of the user's own interactive session, by:

- 4899 a) clearing or overwriting display devices, making the current contents
 4900 unreadable;
 4901 b) disabling any activity of the user's data access/display devices other than
 4902 unlocking the session.

4903 FTA_SSL.2.2

4904 The TSF **shall** require the following events to occur prior to unlocking the session:
 4905 [assignment: *events to occur*].

4906 **16.4.12 FTA_SSL.3 TSF-initiated termination**

4907 **Component relationships**

4908 Hierarchical to: No other components.

4909 Dependencies: FMT_SMR.1 Security roles

4910 **FTA_SSL.3.1**

4911 **The TSF shall terminate an interactive session after a [assignment: time interval of user**
 4912 **inactivity].**

4913 **16.4.13 FTA_SSL.4 User-initiated termination**

4914 **Component relationships**

4915 Hierarchical to: No other components.

4916 Dependencies: No dependencies.

4917 **FTA_SSL.4.1**

4918 **The TSF shall allow user-initiated termination of the user's own interactive session.**

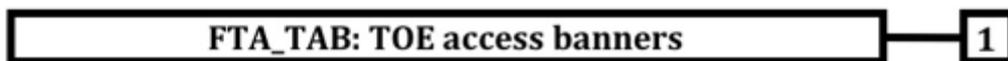
4919 **16.5 TOE access banners (FTA_TAB)**

4920 **16.5.1 Family behaviour**

4921 This family defines requirements to display a configurable advisory warning message to users
 4922 regarding the appropriate use of the TOE.

4923 **16.5.2 Components leveling and description**

4924 Figure 85 shows the component leveling for this family.



4925

4926 **Figure 85 — FTA_TAB: Component leveling**

4927 FTA_TAB.1 Default TOE access banners, provides the requirement for a TOE Access Banner.
 4928 This banner is displayed prior to the establishment dialogue for a session.

4929 **16.5.3 Management of FTA_TAB.1**

4930 The following actions could be considered for the management functions in FMT:

- 4931 a) Maintenance of the banner by the authorized administrator.

4932 **16.5.4 Audit of FTA_TAB.1**

4933 The following actions should be auditable if FAU_GEN Security audit data generation is included
 4934 in the PP/ST:

- 4935 a) There are no auditable events foreseen.

4936 **16.5.5 FTA_TAB.1 Default TOE access banners**

4937 **Component relationships**

4938 Hierarchical to: No other components.

4939 Dependencies: No dependencies.

4940 FTA_TAB.1.1

4941 Before establishing a user session, the [selection: *TSF, TOE platform*] **shall** display an
4942 [assignment: *description of the message*] message.

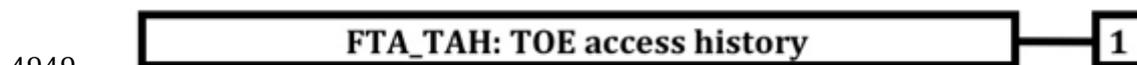
4943 16.6 TOE access history (FTA_TAH)

4944 16.6.1 Family behaviour

4945 This family defines requirements for the TSF to display to a user, upon successful session
4946 establishment, a history of successful and unsuccessful attempts to access the user's account.

4947 16.6.2 Components leveling and description

4948 Figure 86 shows the component leveling for this family.



4950 **Figure 86 — FTA_TAH: Component leveling**

4951 FTA_TAH.1 TOE access history, provides the requirement for a TOE to display information
4952 related to previous attempts to establish a session.

4953 16.6.3 Management of FTA_TAH.1

4954 The following actions **could** be considered for the management functions in FMT:

4955 a) There are no management activities foreseen.

4956 16.6.4 Audit of FTA_TAH.1

4957 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4958 in the PP/ST:

4959 a) There are no auditable events foreseen.

4960 16.6.5 FTA_TAH.1 TOE access history

4961 Component relationships

4962 Hierarchical to: No other components.

4963 Dependencies: No dependencies.

4964 FTA_TAH.1.1

4965 Upon successful session establishment, the TSF **shall** display the [selection: *date, time,*
4966 *method, location*] of the last successful session establishment to the user.

4967 FTA_TAH.1.2

4968 Upon successful session establishment, the TSF **shall** display the [selection: *date, time,*
4969 *method, location*] of the last unsuccessful attempt to session establishment and the
4970 number of unsuccessful attempts since the last successful session establishment.

4971 FTA_TAH.1.3

4972 The TSF **shall** not erase the access history information from the user interface without
4973 giving the user an opportunity to review the information.

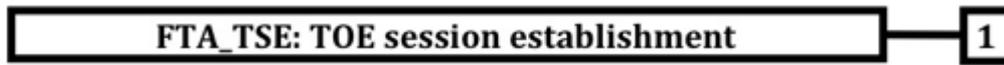
4974 **16.7 TOE session establishment (FTA_TSE)**

4975 **16.7.1 Family behaviour**

4976 This family defines requirements to deny a user permission to establish a session with the TOE.

4977 **16.7.2 Components leveling and description**

4978 Figure 87 shows the component leveling for this family.



4979

4980 **Figure 87 — FTA_TSE: Component leveling**

4981 FTA_TSE.1 TOE session establishment, provides requirements for denying users access to the
4982 TOE based on attributes.

4983 **16.7.3 Management of FTA_TSE.1**

4984 The following actions **could** be considered for the management functions in FMT:

- 4985 a) Management of the session establishment conditions by the authorized
4986 administrator.

4987 **16.7.4 Audit of FTA_TSE.1**

4988 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
4989 in the PP/ST:

- 4990 a) Minimal: Denial of a session establishment due to the session establishment
4991 mechanism.
- 4992 b) Basic: All attempts at establishment of a user session.
- 4993 c) Detailed: Capture of the value of the selected access parameters.

4994 **16.7.5 FTA_TSE.1 TOE session establishment**

4995 **Component relationships**

4996 Hierarchical to: No other components.

4997 Dependencies: No dependencies.

4998 **FTA_TSE.1.1**

4999 **The TSF **shall** be able to deny session establishment based on [assignment: *attributes*].**

5000

5001 **17 Class FTP: Trusted path/channels**5002 **17.1 Class description**

5003 Families in this class provide requirements for a trusted communication path between users
5004 and the TSF, and for a trusted communication channel between the TSF and other trusted IT
5005 products. Trusted paths and channels have the following general characteristics:

5006 — The communications path is constructed using internal and external communications
5007 channels (as appropriate for the component) that isolate an identified subset of TSF data
5008 and commands from the remainder of the TSF and user data.

5009 — Use of the communications path **can** be initiated by the user and/or the TSF (as appropriate
5010 for the component).

5011 — The communications path is capable of providing assurance that the user is communicating
5012 with the correct TSF, and that the TSF is communicating with the correct user (as
5013 appropriate for the component).

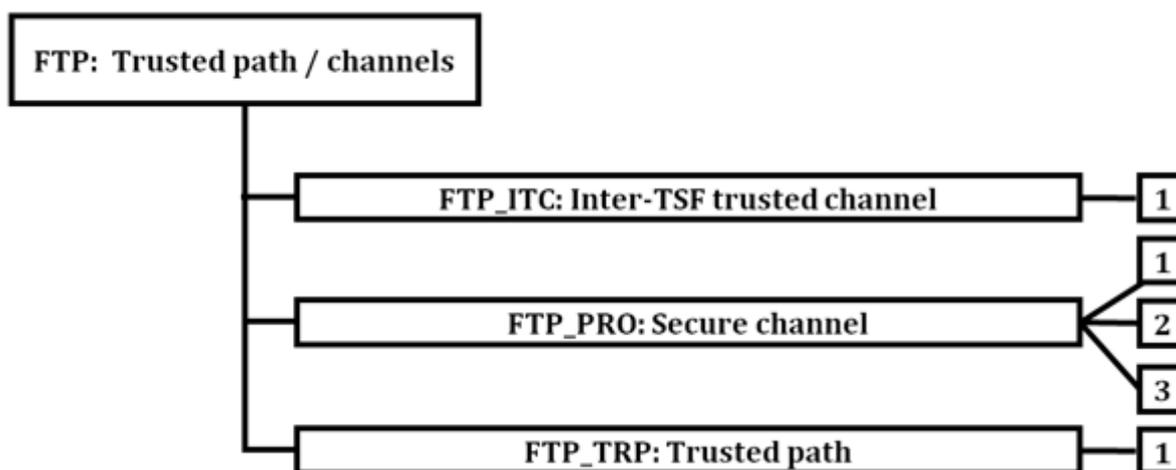
5014 In this paradigm, a trusted channel is a communication channel that **can** be initiated by either
5015 side of the channel and provides non-repudiation characteristics with respect to the identity of
5016 the sides of the channel.

5017 A trusted path provides a means for users to perform functions through an assured direct
5018 interaction with the TSF. Trusted path is usually desired for user actions such as initial
5019 identification and/or authentication but **can** also be desired at other times during a user's
5020 session. Trusted path exchanges **can** be initiated by a user or the TSF. User responses via the
5021 trusted path are guaranteed to be protected from modification by or disclosure to untrusted
5022 applications.

5023 Families describing the use of commonly used communication protocols used in the provision
5024 of trusted channels and paths are also given.

5025 Figure 88 shows the decomposition of this class, it's families and components. Elements are not
5026 shown in the figure.

5027 Annex M provides explanatory information for this class and **should** be consulted when using
5028 the components identified in this class.



5029

5030

Figure 88 — FTP: Trusted path/channels class decomposition

5031 **17.2 Inter-TSF trusted channel (FTP_ITC)**

5032 **17.2.1 Family behaviour**

5033 This family defines requirements for the creation of a trusted channel between the TSF and
 5034 other trusted IT products for the performance of security critical operations. The components
 5035 of this family **may** be included whenever there are requirements for the secure communication
 5036 of user or TSF data between the TOE and other trusted IT products.

5037 **17.2.2 Components leveling and description**

5038 Figure 88 — FTP: Trusted path/channels class decomposition shows the component leveling
 5039 for this family.

5040 FTP_ITC.1 Inter-TSF trusted channel, requires that the TSF provide a trusted communication
 5041 channel between itself and another trusted IT product.

5042 **17.2.3 Management of FTP_ITC.1**

5043 The following actions **could** be considered for the management functions in FMT:

- 5044 a) Configuring the actions that require trusted channel, if supported.

5045 **17.2.4 Audit of FTP_ITC.1**

5046 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
 5047 in the PP/ST:

- 5048 a) Minimal: Failure of the trusted channel functions.
 5049 b) Minimal: Identification of the initiator and target of failed trusted channel
 5050 functions.
 5051 c) Basic: All attempted uses of the trusted channel functions.
 5052 d) Basic: Identification of the initiator and target of all trusted channel functions.

5053 **17.2.5 FTP_ITC.1 Inter-TSF trusted channel**

5054 **Component relationships**

5055 Hierarchical to: No other components.

5056 Dependencies: No dependencies.

5057 **FTP_ITC.1.1**

5058 **The TSF shall provide a communication channel between itself and another trusted IT**
 5059 **product that is logically distinct from other communication channels and provides**
 5060 **assured identification of its end points and protection of the channel data from**
 5061 **modification or disclosure.**

5062 **FTP_ITC.1.2**

5063 **The TSF shall permit [selection: *the TSF, another trusted IT product*] to initiate**
 5064 **communication via the trusted channel.**

5065 **FTP_ITC.1.3**

5066 **The TSF shall initiate communication via the trusted channel for [assignment: *list of***
 5067 ***functions for which a trusted channel is required*].**

5068 **17.3 Secure channel (FTP_PRO)**5069 **17.3.1 Family behavior**

5070 This family defines requirements for establishing a secure channel and using the secure channel
5071 to transfer the TSF data or user data securely.

5072 **17.3.2 Components leveling and description**

5073 Figure 89 shows the component leveling for this family.



5074

5075 **Figure 89 — FTP_PRO: Family decomposition**

5076 Minimal: Establishment of the secure channel.

5077 a) Minimal: Failures of the secure channel functions.

5078 b) Minimal: Identification of the user associated with all secure channel failures, if
5079 available.

5080 c) Basic: All attempted uses of the secure channel functions.

5081 d) Basic: Identification of the user associated with all secure channel invocations, if
5082 available.

5083 FTP_PRO.1 Secure channel protocol requires that communication be established in accordance
5084 with a defined protocol.FTP_PRO.1.5

5085 The TSF shall enforce the following **static protocol options**: [assignment: *list of options and*
5086 *references to standards in which each is defined*].

5087 **FTP_PRO.1.6**

5088 **The TSF shall negotiate one of the following protocol configurations with its peer:**
5089 **[assignment: *list of configurations and reference to standards in which each is defined*].**

5090 FTP_PRO.2 Secure channel establishment requires that keys be securely established between
5091 the peers.

5092 FTP_PRO.3 Secure channel data protection requires that data in transit be protected.

5093 **17.3.3 Management of FTP_PRO.1**

5094 The following actions **could** be considered for the management functions in FMT:

5095 a) Configuring the actions that require secure channel, if supported.

5096 **17.3.4 Audit of FTP_PRO.1**

5097 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
5098 in the PP/ST:

5099 b) Minimal: Establishment of the secure channel.

5100 c) Minimal: Failures of the secure channel functions.

5101 d) Minimal: Identification of the user associated with all secure channel failures, if
5102 available.

- 5103 e) Basic: All attempted uses of the secure channel functions.
- 5104 f) Basic: Identification of the user associated with all secure channel invocations, if
5105 available.
- 5106 **17.3.5 FTP_PRO.1 Secure channel protocol**
- 5107 **Component relationships**
- 5108 Hierarchical to: No other components.
- 5109 Dependencies: FTP_PRO.2 Secure channel establishment
- 5110 FTP_PRO.3 Secure channel data protection.
- 5111 **FTP_PRO.1.1**
- 5112 The TSF **shall** implement [assignment: *secure channel protocol*] acting as [assignment:
5113 *defined protocol role(s)*] in accordance with: [assignment: *list of standards*].
- 5114 **FTP_PRO.1.2**
- 5115 The TSF **shall** enforce usage of the trusted channel for [assignment: *purpose(s) of the*
5116 *trusted channel*] in accordance with: [assignment: *list of standards*].
- 5117 **FTP_PRO.1.3**
- 5118 The TSF **shall** permit [selection: *itself, its peer*] to initiate communication via the secure
5119 channel.
- 5120 **FTP_PRO.1.4**
- 5121 The TSF **shall** enforce the following rules for the secure channel: [assignment: *rules*
5122 *governing operation and use of the secure channel and/or its protocol*].
- 5123 The TSF **shall** enforce usage of the secure channel for [assignment: *purpose of the secure*
5124 *channel*] in accordance with: [assignment: *list of standards*].
- 5125 **FTP_PRO.1.5**
- 5126 The TSF **shall** enforce the following static protocol options: [assignment: *list of options*
5127 *and references to standards in which each is defined*].
- 5128 **FTP_PRO.1.6**
- 5129 The TSF **shall** negotiate one of the following protocol configurations with its peer:
5130 [assignment: *list of configurations and reference to standards in which each is defined*].
- 5131 **17.3.6 FTP_PRO.2 Secure channel establishment**
- 5132 **Component relationships**
- 5133 Hierarchical to: No other components.
- 5134 Dependencies: FTP_PRO.1 Secure channel protocol
- 5135 [FCS_CKM.1 Cryptographic key generation, or
- 5136 FCS_CKM.2 Cryptographic key distribution]
- 5137 FCS_CKM.5 Cryptographic key derivation
- 5138 FCS_COP.1 Cryptographic operation.

5139 **FTP_PRO.2.1**

5140 The TSF **shall** establish a shared secret with its peer using one of the following
 5141 mechanisms: [assignment: *list of key establishment mechanisms*].

5142 **FTP_PRO.2.2**

5143 The TSF **shall** authenticate [selection: *its peer, itself to its peer*] using one of the following
 5144 mechanisms: [assignment: *list of authentication mechanisms*] and according to the
 5145 following rules: [assignment: *list of rules for carrying out the authentication*].

5146 **FTP_PRO.2.3**

5147 The TSF **shall** use [assignment: *key derivation function*] to derive the following
 5148 cryptographic keys from a shared secret: [assignment: *list of cryptographic keys*].

5149 **17.3.7 FTP_PRO.3 Secure channel data protection**5150 **Component relationships**

5151	Hierarchical to:	No other components.
5152	Dependencies:	FTP_PRO.1 Secure channel protocol
5153		FTP_PRO.2 Secure channel establishment
5154		FCS_COP.1 Cryptographic operation.

5155 **FTP_PRO.3.1**

5156 The TSF **shall** protect data in transit from unauthorised disclosure using one of the
 5157 following mechanisms: [assignment: *list of encryption mechanisms*].

5158 **FTP_PRO.3.2**

5159 The TSF **shall** protect data in transit from [selection: *modification, deletion, insertion,*
 5160 *replay, [assignment: other]*] using one of the following mechanisms: [assignment: *list of*
 5161 *integrity protection mechanisms*].

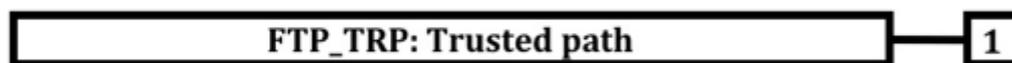
5162 **17.4 Trusted path (FTP_TRP)**5163 **17.4.1 Family behaviour**

5164 This family defines the requirements to establish and maintain trusted communication to or
 5165 from users and the TSF. A trusted path **can** be required for any security-relevant interaction.
 5166 Trusted path exchanges **can** be initiated by a user during an interaction with the TSF, or the TSF
 5167 **can** establish communication with the user via a trusted path.

5168 **17.4.2 Components leveling and description**

5169 Figure 90 shows the component leveling for this family.

5170



5171

Figure 90 — FTP_TRP: Component leveling

5172 FTP_TRP.1 Trusted path, requires that a trusted path between the TSF and a user be provided
 5173 for a set of events defined by a PP/ST author. The user and/or the TSF **can** have the ability to
 5174 initiate the trusted path.

5175 **17.4.3 Management of FTP_TRP.1**

5176 The following actions **could** be considered for the management functions in FMT:

- 5177 a) Configuring the actions that require trusted path, if supported.

5178 **17.4.4 Audit of FTP_TRP.1**

5179 The following actions **should** be auditable if FAU_GEN Security audit data generation is included
5180 in the PP/ST:

- 5181 a) Minimal: Failures of the trusted path functions.

- 5182 b) Minimal: Identification of the user associated with all trusted path failures, if
5183 available.

- 5184 c) Basic: All attempted uses of the trusted path functions.

- 5185 d) Basic: Identification of the user associated with all trusted path invocations, if
5186 available.

5187 **17.4.5 FTP_TRP.1 Trusted path**

5188 **Component relationships**

5189 Hierarchical to: No other components.

5190 Dependencies: No dependencies.

5191 **FTP_TRP.1.1**

5192 **The TSF **shall** provide a communication path between itself and [selection: *remote, local*]**
5193 **users that is logically distinct from other communication paths and provides assured**
5194 **identification of its end points and protection of the communicated data from [selection:**
5195 ***modification, disclosure, [assignment: other types of integrity or confidentiality violation]*].**

5196 **FTP_TRP.1.2**

5197 **The TSF **shall** permit [selection: *the TSF, local users, remote users*] to initiate**
5198 **communication via the trusted path.**

5199 **FTP_TRP.1.3**

5200 **The TSF **shall** require the use of the trusted path for [selection: *initial user***
5201 ***authentication, [assignment: other services for which trusted path is required]*].**

5202
5203
5204

Annex A (normative)

Security functional requirements structure of the application notes

5205 **A.1 General information**

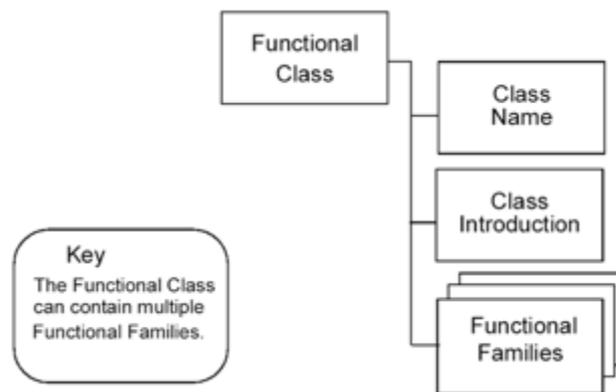
5206 This annex contains additional guidance for the families and components defined in this
5207 document, which **may** be required by users, developers, or evaluators to use the components.
5208 To facilitate finding the appropriate information, the presentation of the classes, families and
5209 components in this annex is similar to the presentation within the main clauses of this
5210 document.

5211 **A.2 Structure of the notes**

5212 This clause defines the content and presentation of the notes related to functional requirements
5213 in this document.

5214 **A.2.1 Class structure**

5215 Figure 91 below illustrates the functional class structure in this annex.



5216
5217

Figure 91 — Functional class structure

5218 **A.2.1.1 Class name**

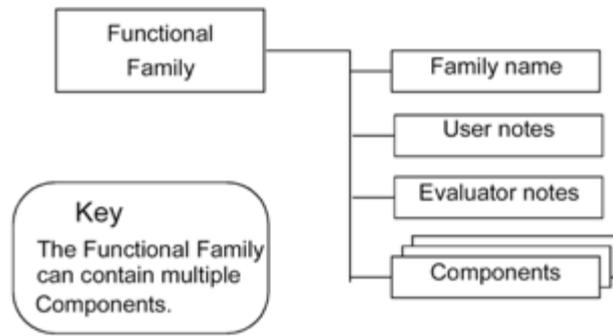
5219 This is the unique name of the class defined within the normative elements of this document.

5220 **A.2.1.2 Class introduction**

5221 The class introduction in this annex provides information about the use of the families and
5222 components of the class. This information is completed with the informative diagram that
5223 describes the organization of each class with the families in each class and the hierarchical
5224 relationship between components in each family.

5225 **A.2.2 Family structure**

5226 Figure 92 illustrates the functional family structure for application notes in diagrammatic form.



5227

5228

Figure 92 — Functional family structure for application notes

5229 **A.2.2.1 Family name**

5230 This is the unique name of the family defined within the normative elements of this document.

5231 **A.2.2.2 User notes**

5232 The user notes contain additional information that is of interest to potential users of the family,
 5233 that is PP, PP-Module, ST and functional package authors, and developers of TOEs incorporating
 5234 the functional components. The presentation is informative and might cover warnings about
 5235 limitations of use and areas where specific attention might be required when using the
 5236 components.

5237 **NOTE** In the annexes the term PP/ST author includes authors of documents used to formulate a PP or ST, this
 5238 includes PP-Modules and functional packages.

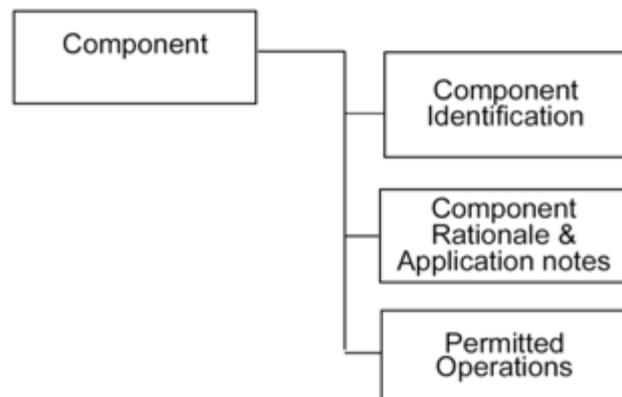
5239 **A.2.2.3 Evaluator notes**

5240 The evaluator notes contain any information that is of interest to developers and evaluators of
 5241 TOEs that claim compliance with a component of the family. The presentation is informative
 5242 and **can** cover a variety of areas where specific attention might be needed when evaluating the
 5243 TOE. This **can** include clarifications of meaning and specification of the way to interpret
 5244 requirements, as well as caveats and warnings of specific interest to evaluators.

5245 These User Notes and Evaluator Notes subclauses are not mandatory and appear only if
 5246 appropriate.

5247 **A.2.3 Component structure**

5248 Figure 93 illustrates the functional component structure for the application notes.



5249

5250

Figure 93 — Functional component structure

5251 **A.2.3.1 Component identification**

5252 This is the unique name of the component defined within the normative elements of this
5253 document.

5254 **A.2.3.2 Component rationale and application notes**

5255 Any specific information related to the component is found in this subclause.

5256 — The *rationale* contains the specifics of the rationale that refine the general statements on
5257 rationale for the specific level and is only be used if level specific amplification is required.

5258 — The *application notes* contain additional refinement in terms of narrative qualification as it
5259 pertains to a specific component. This refinement **can** pertain to user notes, and/or
5260 evaluator notes as described in A.2.2. This refinement **can** be used to explain the nature of
5261 the dependencies.

EXAMPLE

Shared information, or shared operation.

5262 This subclause is not mandatory and appears only if appropriate.

5263 **A.2.3.3 Permitted operations**

5264 This portion of each component contains advice relating to the permitted operations of the
5265 component.

5266 This subclause is not mandatory and appears only if appropriate.

5267
5268
5269

Annex B (informative)

Dependency tables for security functional components

5270

B.1 Dependency tables

5271 The following dependency tables for functional components show their hierarchical, direct,
5272 indirect, and optional dependencies.

5273 Each of the components that is a dependency of some functional component is allocated a
5274 column. Each functional component is allocated a row. The value in the table cell indicates
5275 whether the column label component is a hierarchical requirement (indicated by an "H").
5276 directly required (indicated by a cross "X"), indirectly required (indicated by a dash "-"), or
5277 optionally required (indicated by a "O") by the row label component. Sets of optional
5278 requirements are indicated by using a subscript group, e.g. O¹ and O².

5279 NOTE Depending upon the optional requirements chosen, some indirect dependencies are not
5280 applicable.

5281 If no character is presented, the component is not dependent upon another component.

5282

EXAMPLE

An example of a component with optional dependencies is FDP_ETC.1 Export of user data without security attributes, which requires either FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control to be present. So, if FDP_ACC.1 Subset access control is present, FDP_IFC.1 Subset information flow control is not necessary and vice versa.

5283

5284

Table B.3 — 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					-	

5285

5286

5287

Table B.4 — 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	

5288

5289 **Table B.5 — Dependency table for Class FCS: Cryptographic support**

	FCS_CKM.1	FCS_CKM.2	FCS_CKM.3	FCS_CKM.5	FCS_CGM.6	FCS_COP.1	FCS_RBG.1	FCS_RBG.2	FCS_RBG.3	FCS_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_TST.1	FPT_TDC.1	FTP_ITC.1	FTP_TRP.1
FCS_CKM.1	-	O ¹	X	O ¹	X	O ¹	O ²	-	-	O ²	-	-	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																														

5290

Table B.6 — Dependency table for Class FDP: User data protection

	FCS_CKM.1	FCS_CKM.3	FCS_CKM.5	FCS_CKM.6	FCS_COP.1	FCS_RBG.1	FCS_RBG.2	FCS_RBG.3	FCS_RNG.1	FDP_ACC.1	FDP_ACF.1	FDP_IFC.1	FDP_IFF.1	FDP_IFF.3	FDP_IFF.4	FDP_IFF.5	FDP_IFF.6	FDP_IRC.1	FDP_ITC.1	FDP_UIT.1	FDP_UIT.2	FDP_UIT.3	FDP_RIP.1	FDP_ROL.1	FDP_SDI.1	FDP_URT.1	FDP_URT.2	FDP_URT.3	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						
FDP_ACC.1										-	X	-	-																														
FDP_ACC.2										H	X	-	-																														
FDP_ACF.1										X	-	-	-																	X													
FDP_DAU.1																																											
FDP_DAU.2										H																																	
FDP_ETC.1										O ¹	-	O ¹	-																														
FDP_ETC.2										O ¹	-	O ¹	-																														
FDP_IFC.1										-	-	-	X																														
FDP_IFC.2										-	-	H	X																														
FDP_IFF.1										-	-	X	-																	X													
FDP_IFF.2										-	-	X	H																	X													
FDP_IFF.3										-	-	X	-																														
FDP_IFF.4										-	-	X	-	H																													
FDP_IFF.5										-	-	X	-		H																												
FDP_IFF.6										-	-	X	-																														
FDP_IRC.1																																											
FDP_ITC.1										O ¹	-	O ¹	-																	X													

5293

5294

Table B.7 — Dependency table for Class FIA: Identification and authentication

	FIA_ATD.1	FIA_UAU.1	FIA_UID.1	FMT_SMR.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			

5295

5296

5297

Table B.8 — 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	

5298

5299

5300

Table B.9 — 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				

5301

5302

5303

Table B.10 — 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														

5304

5305 NOTE The AGD and ADV classes and their dependencies are described in ISO/IEC 15408-3

5306

Table B.11 — 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

5307

5308

5309

Table B.12 — 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				

5310

Table B.13 — 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					
FTP_PRO.3		X	X																					X					
FTP_TRP.1																													

5314
5315
5316
5317

Annex C (normative)

Class FAU: Security audit - application notes

5318 C.1 General information

5319 ISO/IEC 15408 audit families allow PP/ST authors the ability to define requirements for
5320 monitoring user activities and, in some cases, detecting real, possible, or imminent violations of
5321 the enforcement of the SFRs. The TOE's security audit functions are defined to help monitor
5322 security-relevant events, and act as a deterrent against security violations. The requirements of
5323 the audit families refer to functions that include audit data protection, record format, and event
5324 selection, as well as analysis tools, violation alarms, and real-time analysis. The audit records
5325 **may** be presented in human-readable format either directly or indirectly or both.

EXAMPLE 1

An example of direct presentation is storing the audit records in human-readable format

An example of indirect presentation is by using audit reduction tools.

5326 While developing the security audit requirements, the PP/ST author **should** take note of the
5327 inter-relationships among the audit families and components. The potential exists to specify a
5328 set of audit requirements that comply with the family/component dependencies lists, while at
5329 the same time resulting in a deficient audit function.

EXAMPLE 2

An audit function that requires all security relevant events to be audited but without the selectivity to control them on any reasonable basis such as individual user or object.

5330 C.2 Audit requirements in a distributed environment

5331 The implementation of audit requirements for networks and other large systems **can** differ
5332 significantly from those needed for stand-alone systems. Larger, more complex, and active
5333 systems require more thought concerning which audit data to collect and how this **can** be
5334 managed, due to lowered feasibility of interpreting (or even storing) what gets collected. The
5335 traditional notion of a time-ordered list, set of records or "trail" of audited events is not always
5336 applicable in a global asynchronous network with many arbitrary events occurring at once.

5337 Also, different hosts and servers on a distributed TOE **can** have differing naming policies and
5338 values. Further, the use of symbolic names for audit review requires a net-wide convention to
5339 avoid redundancies and "name clashes."

5340 A multi-object audit repository, portions of which are accessible by a potentially wide variety of
5341 authorized users, are usually required if audit repositories are to serve a useful function in
5342 distributed systems.

5343 Finally, misuse of authority by authorized users **can** be addressed by systematically avoiding
5344 local storage of audit data pertaining to administrator actions.

5345 C.3 Security audit automatic response (FAU_ARP)

5346 C.3.1 User notes

5347 The Security audit automatic response family describes requirements for the handling of audit
5348 events. The requirement **could** include requirements for alarms or TSF action (automatic
5349 response).

EXAMPLE

the TSF **could** include the generation of real time alarms, termination of the offending process, disabling of a service, or disconnection or invalidation of a user account.

5350 An audit event is defined to be an “potential security violation” if so indicated by the Security
5351 audit analysis (FAU_SAA) components.

5352 **C.3.2 FAU_ARP.1 Security alarms**

5353 **C.3.2.1 User application notes**

5354 One or more actions **should** be taken for follow up action in the event of an alarm.

5355 These actions **could** include informing the authorized user of the alarm, presenting the
5356 authorized user with a set of possible containment actions, or options for the authorized user
5357 to take corrective actions.

5358 The timing of the actions **should** be carefully considered by the PP/ST author.

5359 **C.3.2.2 Operations**

5360 **C.3.2.2.1 Assignment**

5361 In FAU_ARP.1.1, the PP/ST author specifies the actions to be taken in case of a potential security
5362 violation.

EXAMPLE

An example of such a list is: “inform the authorized user, disable the subject that created the potential security violation.”

5363 The list **may** also specify that the action to be taken **can** be specified by an authorized user.

5364 **C.4 Security audit data generation (FAU_GEN)**

5365 **C.4.1 User notes**

5366 The Security audit data generation family includes requirements to specify the audit events that
5367 **shall** be generated by the TSF for security-relevant events.

5368 This family is presented in a manner that avoids a dependency on all components requiring
5369 audit support. Each component has an audit subclause developed in which the events to be
5370 audited for that functional area are listed. When the PP/ST is written, the items in the audit area
5371 are used to complete the variable in these components. Thus, the specification of what **could** be
5372 audited for a functional area is localized in that functional area.

5373 The list of auditable events is entirely dependent on the other functional families within the
5374 PP/ST. Each family definition **should** therefore include a list of its family-specific auditable
5375 events. Each auditable event in the list of auditable events specified in the functional family
5376 **should** correspond to one of the levels of audit event generation specified in this family (i.e.
5377 minimal, basic, detailed). This provides the PP/ST author with information necessary to ensure
5378 that all appropriate auditable events are specified in the PP/ST. The following example shows
5379 how auditable events are to be specified in appropriate functional families:

EXAMPLE 1

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

- a) Minimal: Successful use of the user security attribute administration functions.
- b) Basic: All attempted uses of the user security attribute administration functions.
- c) Basic: Identification of which user security attributes have been modified.
- d) Detailed: With the exception of specific sensitive attribute data items, the new values of the attributes should be captured.”

NOTE Sensitive attribute data items include passwords and cryptographic keys.

5380

5381 For each functional component that is chosen, the auditable events that are indicated in that
 5382 component, at and below the level indicated in Security audit data generation (FAU_GEN)
 5383 **should** be auditable. So, in the previous example “Basic” would be selected in Security audit data
 5384 generation (FAU_GEN), the auditable events mentioned in a), b) and c) **should** be auditable.

5385 Observe that the categorization of auditable events (minimal, basic, detailed) is hierarchical in
 5386 that order.

5387 This means that

- 5388 • When Minimal Audit Generation is desired, all auditable events identified as being
 5389 Minimal **should** be included in the PP/ST through the use of the appropriate assignment
 5390 operation.
- 5391 • When Basic Audit Generation is desired, all auditable events identified as being either
 5392 Minimal or Basic, **should** also be included in the PP/ST through the use of the
 5393 appropriate assignment operation, except when the higher-level event simply provides
 5394 more detail than the lower level event.
- 5395 • When Detailed Audit Generation is desired, all identified auditable events (Minimal,
 5396 Basic, and Detailed) **should** be included in the PP/ST.

5397 A PP/ST author **may** decide to include other auditable events beyond those required for a given
 5398 audit level.

EXAMPLE 2

For example, the PP/ST **may** claim only minimal audit capabilities while including most of the basic capabilities because the few excluded capabilities conflict with other PP/ST constraints (perhaps because they require the collection of unavailable data).

5399 The functionality that creates the auditable event **should** be specified in the PP or ST as a
 5400 functional requirement.

EXAMPLE 3

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

- a) Introduction of objects within the control of the TSF into a subject's address space;
- b) Deletion of objects;
- c) Distribution or revocation of access rights or capabilities;
- d) Changes to subject or object security attributes;
- e) Policy checks performed by the TSF as a result of a request by a subject;
- f) The use of access rights to bypass a policy check;
- g) Use of Identification and Authentication functions;
- h) Actions taken by an operator, and/or authorized user (such as suppression of a TSF protection mechanism as human-readable labels);
- i) Import/export of data from/to removable media (such as printed output, tapes, USB sticks).

5401 C.4.2 FAU_GEN.1 Audit data generation

5402 C.4.2.1 User application notes

5403 This component defines requirements to identify the auditable events for which audit records
 5404 **should** be generated, and the information to be provided in the audit records.

5405 FAU_GEN.1 Audit data generation by itself might be used when the SFRs do not require that
 5406 individual user identities be associated with audit events. This **could** be appropriate when the

5407 PP/ST also contains privacy requirements. If the user identity must be incorporated FAU_GEN.2
5408 User identity association **could** be used in addition to FAU_GEN.1.

5409 If the subject is a user, the user identity **may** be recorded as the subject identity. The identity of
5410 the user **may** not yet have been verified if User authentication (FIA_UAU) has not been applied.
5411 Therefore, in the instance of an invalid login the claimed user identity **should** be recorded. It
5412 **should** also be considered whether to indicate when a recorded identity has not been
5413 authenticated.

5414 **C.4.2.2 Evaluator notes**

5415 FAU_GEN.1.1 has a dependency on FPT_STM.1 Reliable time stamps. If correctness of time is not
5416 an issue for this TOE, elimination of this dependency **could** be justified by the PP/ST author.

5417 **C.4.2.3 Operations**

5418 **C.4.2.3.1 Selection**

5419 In FAU_GEN.1.1, the PP/ST author **should** select the level of auditable events called out in the
5420 audit subclause of other functional components included in the PP/ST. This level is one of the
5421 following: “minimum”, “basic”, “detailed” or “not specified”.

5422 **C.4.2.3.2 Assignment**

5423 In FAU_GEN.1.1, the PP/ST author **should** assign a list of other specifically defined auditable
5424 events to be included in the list of auditable events. The assignment **may** comprise none, or
5425 events that **could** be auditable events of a functional requirement that are of a higher audit level
5426 than requested in b), as well as the events generated through the use of a specified Application
5427 Programming Interface (API).

5428 In FAU_GEN.1.2, the PP/ST author **should** assign, for each of the auditable events included in the
5429 PP/ST, either a list of other audit relevant information to be included in audit events records or
5430 none.

5431 **C.4.3 FAU_GEN.2 User identity association**

5432 **C.4.3.1 User application notes**

5433 This component addresses the requirement of accountability of auditable events at the level of
5434 individual user identity. This component **should** be used in addition to FAU_GEN.1 Audit data
5435 generation.

5436 There is a potential conflict between the audit and privacy requirements. For audit purposes, it
5437 **may** be desirable to know who performed an action. A user **may** want to keep his/her actions to
5438 himself/herself and not be identified by other persons such as a site with job offers. Or it might
5439 be required in the Organizational Security Policy that the identity of the users must be
5440 protected. In those cases, the objectives for audit and privacy **could** contradict each other.
5441 Therefore, if this requirement is selected and privacy is important, inclusion of the component
5442 user pseudonymity might be considered. Requirements on determining the real user name
5443 based on its pseudonym are specified in the privacy class.

5444 If the identity of the user has not yet been verified through authentication, in the instance of an
5445 invalid login the claimed user identity **should** be recorded. It **should** be considered to indicate
5446 when a recorded identity has not been authenticated.

5447 **C.5 Security audit analysis (FAU_SAA)**

5448 **C.5.1 User notes**

5449 This family defines requirements for automated means that analyze system activity and audit
5450 data looking for possible or real security violations. This analysis **may** work in support of
5451 intrusion detection, or automatic response to a potential security violation.

5452 The action to be performed by the TSF on detection of a potential violation is defined in Security
5453 audit automatic response (FAU_ARP) components.

5454 For real-time analysis, audit data **could** be transformed into a useful format for automated
5455 treatment, but into a different useful format for delivery to authorized users for review.

5456 **C.5.2 FAU_SAA.1 Potential violation analysis**

5457 **C.5.2.1 User application notes**

5458 This component is used to specify the set of auditable events whose occurrence or accumulated
5459 occurrence held to indicate a potential violation of the enforcement of the SFRs, and any rules to
5460 be used to perform the violation analysis.

5461 **C.5.2.2 Operations**

5462 **C.5.2.2.1 Assignment**

5463 In FAU_SAA.1.2, the PP/ST author **should** identify the subset of defined auditable events whose
5464 occurrence or accumulated occurrence need to be detected as an indication of a potential
5465 violation of the enforcement of the SFRs.

5466 In FAU_SAA.1.2, the PP/ST author **should** specify any other rules that the TSF **should** use in its
5467 analysis of the audit trail. Those rules **could** include specific requirements to express the needs
5468 for the events to occur in a certain period of time. If there are no additional rules that the TSF
5469 **should** use in the analysis of the audit trail, this assignment **can** be completed with “none”.

EXAMPLE

Period of time: period of the day, duration

5470 **C.5.3 FAU_SAA.2 Profile based anomaly detection**

5471 **C.5.3.1 User application notes**

5472 A *profile* is a structure that characterizes the behaviour of users and/or subjects; it represents
5473 how the users/subjects interact with the TSF in a variety of ways. Patterns of usage are
5474 established with respect to the various types of activity the users/subjects engage in. The ways
5475 in which the various types of activity are recorded in the profile are referred to as *profile*
5476 *metrics*.

EXAMPLE

Patterns of usage: patterns in exceptions raised, patterns in resource utilization (when, which, how), patterns in actions performed.

Profile metrics: resource measures, event counters, timers

5477 Each profile represents the expected patterns of usage performed by members of the *profile*
5478 *target group*. This pattern **may** be based on past use (historical patterns) or on normal use for
5479 users of similar target groups (expected behaviour). A profile target group refers to one or more
5480 users who interact with the TSF. The activity of each member of the profile group is used by the
5481 analysis tool in establishing the usage patterns represented in the profile. The following are
5482 some examples of profile target groups:

- 5483 a) **Single user account:** one profile per user;
- 5484 b) **Group ID or Group Account:** one profile for all users who possess the same group
5485 ID or operate using the same group account;
- 5486 c) **Operating Role:** one profile for all users sharing a given operating role;
- 5487 d) **System:** one profile for all users of a system.

5488 Each member of a profile target group is assigned an individual *suspicion rating* that represents
 5489 how closely that member's new activity corresponds to the established patterns of usage
 5490 represented in the group profile.

5491 The sophistication of the anomaly detection tool will largely be determined by the number of
 5492 target profile groups required by the PP/ST and the complexity of the required profile metrics.

5493 The PP/ST author **should** enumerate specifically what activity **should** be monitored and/or
 5494 analysed by the TSF. The PP/ST author **should** also identify specifically what information
 5495 pertaining to the activity is necessary to construct the usage profiles.

5496 FAU_SAA.2 Profile based anomaly detection requires that the TSF maintain profiles of system
 5497 usage. The word maintain implies that the anomaly detector is actively updating the usage
 5498 profile based on new activity performed by the profile target members. It is important here that
 5499 the metrics for representing user activity are defined by the PP/ST author.

EXAMPLE 2

For example, there **may** be a thousand different actions an individual **may** be capable of performing, but the anomaly detector **may** choose to monitor a subset of that activity.

5500 Anomalous activity gets integrated into the profile just like non-anomalous activity (assuming
 5501 the tool is monitoring those actions). Things that **may** have appeared anomalous four months
 5502 ago, might over time become the norm (and vice-versa) as the user's work duties change. The
 5503 TSF wouldn't be able to capture this notion if it filtered out anomalous activity from the profile
 5504 updating algorithms.

5505 Administrative notification **should** be provided such that the authorized user understands the
 5506 significance of the suspicion rating.

5507 The PP/ST author **should** define how to interpret suspicion ratings and the conditions under
 5508 which anomalous activity is indicated to the Security audit automatic response (FAU_ARP)
 5509 mechanism.

5510 **C.5.3.2 Operations**

5511 **C.5.3.2.1 Assignment**

5512 In FAU_SAA.2.1, the PP/ST author **should** specify the profile target group. A single PP/ST **may**
 5513 include multiple profile target groups.

5514 In FAU_SAA.2.3, the PP/ST author **should** specify conditions under which anomalous activity is
 5515 reported by the TSF. Conditions **may** include the suspicion rating reaching a certain value, or be
 5516 based on the type of anomalous activity observed.

5517 **C.5.4 FAU_SAA.3 Simple attack heuristics**

5518 **C.5.4.1 User application notes**

5519 In practice, it is at best rare when an analysis tool **can** detect with certainty when a security
 5520 violation is imminent. However, there do exist some system events that are so significant that
 5521 they are always worthy of independent review.

EXAMPLE 1

Example of such events include the deletion of a key TSF security data file (such as the password file) or activity such as a remote user attempting to gain administrative privilege.

5522 These events are referred to as signature events in that their occurrence in isolation from the
 5523 rest of the system activity are indicative of intrusive activity.

5524 The complexity of a given tool will depend greatly on the assignments defined by the PP/ST
 5525 author in identifying the base set of *signature events*.

5526 The PP/ST author **should** enumerate specifically what events **should** be monitored by the TSF in
 5527 order to perform the analysis. The PP/ST author **should** identify specifically what information
 5528 pertaining to the event is necessary to determine if the event maps to a signature event.

5529 Administrative notification **should** be provided such that the authorized user understands the
 5530 significance of the event and the appropriate possible responses.

5531 An effort was made in the specification of these requirements to avoid a dependency on audit
 5532 data as the sole input for monitoring system activity. This was done in recognition of the
 5533 existence of previously developed intrusion detection tools that do not perform their analyses
 5534 of system activity solely through the use of audit data.

EXAMPLE 2

Examples of other input data include network datagrams, resource/accounting data, or combinations of various system data.

5535 The elements of FAU_SAA.3 Simple attack heuristics do not require that the TSF implementing
 5536 the immediate attack heuristics be the same TSF whose activity is being monitored. Thus, one
 5537 **can** develop an intrusion detection component that operates independently of the system
 5538 whose system activity is being analyzed.

5539 C.5.4.2 Operations

5540 C.5.4.2.1 Assignment

5541 In FAU_SAA.3.1, the PP/ST author **should** identify a base subset of system events whose
 5542 occurrence, in isolation from all other system activity, **may** indicate a violation of the
 5543 enforcement of the SFRs. These include events that by themselves indicate a clear violation to
 5544 the enforcement of the SFRs, or whose occurrence is so significant that they warrant actions.

5545 In FAU_SAA.3.2, the PP/ST author **should** specify the information used to determine system
 5546 activity. This information is the input data used by the analysis tool to determine the system
 5547 activity that has occurred on the TOE. This data **may** include audit data, combinations of audit
 5548 data with other system data, or **may** consist of data other than the audit data. The PP/ST author
 5549 **should** define precisely what system events and event attributes are being monitored within the
 5550 input data.

5551 C.5.5 FAU_SAA.4 Complex attack heuristics

5552 C.5.5.1 User application notes

5553 In practice, it is at best rare when an analysis tool **can** detect with certainty when a security
 5554 violation is imminent. However, there do exist some system events that are so significant they
 5555 are always worthy of independent review.

EXAMPLE 1

Example of such events include the deletion of a key TSF security data file (such as the password file) or activity such as a remote user attempting to gain administrative privilege.

5556 These events are referred to as signature events in that their occurrence in isolation from the
 5557 rest of the system activity are indicative of intrusive activity. Event sequences are an ordered
 5558 set of signature events that might indicate intrusive activity.

5559 The complexity of a given tool will depend greatly on the assignments defined by the PP/ST
 5560 author in identifying the base set of signature events and event sequences.

5561 The PP/ST author **should** enumerate specifically what events **should** be monitored by the TSF in
 5562 order to perform the analysis. The PP/ST author **should** identify specifically what information
 5563 pertaining to the event is necessary to determine if the event maps to a signature event.

5564 Administrative notification **should** be provided such that the authorized user understands the
 5565 significance of the event and the appropriate possible responses.

5566 An effort was made in the specification of these requirements to avoid a dependency on audit
5567 data as the sole input for monitoring system activity. This was done in recognition of the
5568 existence of previously developed intrusion detection tools that do not perform their analyses
5569 of system activity solely through the use of audit data.

EXAMPLE 2

examples of other input data include network datagrams, resource/accounting data, or combinations of various system data

5570 Levelling, therefore, requires the PP/ST author to specify the type of input data used to monitor
5571 system activity.

5572 The elements of FAU_SAA.4 Complex attack heuristics do not require that the TSF implementing
5573 the complex attack heuristics be the same TSF whose activity is being monitored. Thus, one **can**
5574 develop an intrusion detection component that operates independently of the system whose
5575 system activity is being analyzed.

5576 C.5.5.2 Operations

5577 C.5.5.2.1 Assignment

5578 In FAU_SAA.4.1, the PP/ST author **should** identify a base set of lists of sequences of system
5579 events whose occurrence are representative of known penetration scenarios. These event
5580 sequences represent known penetration scenarios. Each event represented in the sequence
5581 **should** map to a monitored system event, such that as the system events are performed, they
5582 are bound (mapped) to the known penetration event sequences.

5583 In FAU_SAA.4.1, the PP/ST author **should** identify a base subset of system events whose
5584 occurrence, in isolation from all other system activity, **may** indicate a violation of the
5585 enforcement of the SFRs. These include events that by themselves indicate a clear violation to
5586 the SFRs, or whose occurrence is so significant they warrant action.

5587 In FAU_SAA.4.2, the PP/ST author **should** specify the information used to determine system
5588 activity. This information is the input data used by the analysis tool to determine the system
5589 activity that has occurred on the TOE. This data **may** include audit data, combinations of audit
5590 data with other system data, or **may** consist of data other than the audit data. The PP/ST author
5591 **should** define precisely what system events and event attributes are being monitored within the
5592 input data.

5593 C.6 Security audit review (FAU_SAR)

5594 C.6.1 User notes

5595 The Security audit review family defines requirements related to review of the audit
5596 information.

5597 These functions **should** allow pre-storage or post-storage audit selection.

EXAMPLE

An example of requirement related to review of the audit information is the ability to selectively review:

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

5598
5599 The distinction between audit reviews is based on functionality. Audit review (only)
5600 encompasses the ability to view audit data. Selectable review is more sophisticated and

5601 requires the ability to select subsets of audit data based on a single criterion or multiple criteria
5602 with logical (i.e. and/or) relations and order the audit data before it is reviewed.

5603 **C.6.2 FAU_SAR.1 Audit review**

5604 **C.6.2.1 Rationale**

5605 This component will provide authorized users the capability to obtain and interpret the
5606 information. In case of human users this information needs to be in a human understandable
5607 presentation. In case of external IT entities, the information needs to be unambiguously
5608 represented in an electronic fashion.

5609 **C.6.2.2 User application notes**

5610 This component is used to specify that users and/or authorized users **can** read the audit
5611 records. These audit records will be provided in a manner appropriate to the user. There are
5612 different types of users (human users, machine users) that might have different needs.

5613 The content of the audit records that **can** be viewed **can** be specified.

5614 **C.6.2.3 Operations**

5615 **C.6.2.3.1 Assignment**

5616 In FAU_SAR.1.1, the PP/ST author **should** specify the authorized users that **can** use this
5617 capability. If appropriate the PP/ST author **may** include security roles (see FMT_SMR.1 Security
5618 roles).

5619 In FAU_SAR.1.1, the PP/ST author **should** specify the type of information the specified user is
5620 permitted to obtain from the audit records.

EXAMPLE

Examples are “all”, “subject identity”, “all information belonging to audit records referencing this user”.

5621 When employing the SFR, FAU_SAR.1, it is not necessary to repeat, in full detail, the list of audit
5622 information first specified in FAU_GEN.1. Use of terms such as “all” or “all audit information”
5623 assist in eliminating ambiguity and the further need for comparative analysis between the two
5624 security requirements.

5625 **C.6.3 FAU_SAR.2 Restricted audit review**

5626 **C.6.3.1 User application notes**

5627 This component specifies that any users not identified in FAU_SAR.1 Audit review will not be
5628 able to read the audit records.

5629 **C.6.4 FAU_SAR.3 Selectable audit review**

5630 **C.6.4.1 User application notes**

5631 This component is used to specify that it **should** be possible to perform selection of the audit
5632 data to be reviewed. If based on multiple criteria, those criteria **should** be related together with
5633 logical (i.e. “and” or “or”) relations, and the tools **should** provide the ability to manipulate audit
5634 data

EXAMPLE

Means of manipulating audit data include sorting and filtering.

5635 **C.6.4.2 Operations**

5636 **C.6.4.2.1 Assignment**

5637 In FAU_SAR.3.1, the PP/ST author **should** specify whether capabilities to select and/or order
5638 audit data is required from the TSF.

5639 In FAU_SAR.3.1, the PP/ST author **should** assign the criteria, possibly with logical relations, to
5640 be used to select the audit data for review. The logical relations are intended to specify whether
5641 the operation **can** be on an individual attribute or a collection of attributes.

EXAMPLE

An example of this assignment could be: “application, user account and/or location”.

5642 In this case, the operation **could** be specified using any combination of the three attributes:
5643 application, user account and location.

5644 **C.7 Security audit event selection (FAU_SEL)**

5645 **C.7.1 User notes**

5646 The Security audit event selection family provides requirements related to the capabilities of
5647 identifying which of the possible auditable events are to be audited. The auditable events are
5648 defined in the Security audit data generation (FAU_GEN) family, but those events **should** be
5649 defined as being selectable in this component to be audited.

5650 This family ensures that it is possible to keep the audit trail from becoming so large that it
5651 becomes useless, by defining the appropriate granularity of the selected security audit events.

5652 **C.7.2 FAU_SEL.1 Selective audit**

5653 **C.7.2.1 User application notes**

5654 This component defines the selection criteria used, and the resulting audited subsets of the set
5655 of all auditable events, based on user attributes, subject attributes, object attributes, or event
5656 types.

5657 The existence of individual user identities is not assumed for this component. This allows for
5658 TOEs such as routers that **may** not support the notion of users.

5659 For a distributed environment, the host identity **could** be used as a selection criterion for events
5660 to be audited.

5661 The management function FMT_MTD.1 Management of TSF data will handle the rights of
5662 authorized users to query or modify the selections.

5663 **C.7.2.2 Operations**

5664 **C.7.2.2.1 Selection**

5665 In FAU_SEL.1.1, the PP/ST author **should** select whether the security attributes upon which
5666 audit selectivity is based, is related to object identity, user identity, subject identity, host
5667 identity, or event type.

5668 **C.7.2.2.2 Assignment**

5669 In FAU_SEL.1.1, the PP/ST author **should** specify any additional attributes upon which audit
5670 selectivity is based. If there are no additional rules upon which audit selectivity is based, this
5671 assignment **can** be completed with “none”.

5672 **C.8 Security audit data storage (FAU_STG)**

5673 **C.8.1 User notes**

5674 The Security audit data storage family describes requirements for storing audit data for later
5675 use, including requirements controlling the loss of audit information due to TOE failure, attack
5676 and/or exhaustion of storage space.

5677 **C.8.2 FAU_STG.1 Audit data storage location**

5678 **C.8.2.1 User application notes**

5679 In a distributed environment, as the location of the audit trail is in the TSF, but not necessarily
 5680 co-located with the function generating the audit data, the PP/ST author could request
 5681 authentication of the originator of the audit record, or non-repudiation of the origin of the
 5682 record prior storing this record in the audit trail.

5683 The TSF will protect the stored audit records in the audit trail from unauthorised deletion and
 5684 modification. It is noted that in some TOEs the auditor (role) might not be authorized to delete
 5685 the audit records for a certain period of time.

5686 **Since no contribution was received from SMEs in regard to the text for FAU_STG.1, the editors propose**
 5687 **the following text.**

5688 **Unless comments are received in response to this draft, the editors' proposal will be accepted.**

5689 FAU_STG.1.1 is dependent upon FTP_ITC.1 Inter-TSF trusted channel, if “transmit the generated
 5690 audit data to an external IT entity using a trusted channel according to FTP_ITC” is not selected
 5691 then the PP/ST author can satisfy the dependency by providing the rationale that it was not
 5692 selected.

5693 **C.8.2.2 Operations**

5694 **C.8.2.2.1 Selection**

5695 In FAU_STG.1.1 the PP/ST author **should** select where the audit data is stored. Audit data may be
 5696 stored on the TOE itself, be transmitted to an external entity using a trusted channel, or other
 5697 storage options can be specified in the assignment.

5698 **C.8.2.2.2 Assignment**

5699 If additional or alternative storage locations for audit data need to be specified by the PP/ST
 5700 author then this requirement can be specified in FAU_STG.1.1 using the assignment found
 5701 within the selection.

5702 **C.8.3 FAU_STG.2 Protected audit data storage**

5703 **C.8.3.1 User application notes**

5704 In a distributed environment, as the location of the audit trail is in the TSF, but not necessarily
 5705 co-located with the function generating the audit data, the PP/ST author **could** request
 5706 authentication of the originator of the audit record, or non-repudiation of the origin of the
 5707 record prior storing this record in the audit trail.

5708 The TSF will protect the stored audit data in the audit trail from unauthorized deletion and
 5709 modification. It is noted that in some TOEs the auditor (role) might not be authorized to delete
 5710 the audit records for a certain period of time.

5711 **C.8.3.2 Operations**

5712 **C.8.3.2.1 Selection**

5713 In FAU_STG.2.2, the PP/ST author **should** specify whether the TSF **shall** prevent or only be able
 5714 to detect modifications of the stored audit data in the audit trail. Only one of these options **may**
 5715 be chosen.

5716 **C.8.4 FAU_STG.3 Guarantees of audit data availability**

5717 **C.8.4.1 User application notes**

5718 This component allows the PP/ST author to specify to which metrics the audit trail **should**
 5719 conform.

5720 In a distributed environment, as the location of the audit trail is in the TSF, but not necessarily
 5721 co-located with the function generating the audit data, the PP/ST author **could** request
 5722 authentication of the originator of the audit record, or non-repudiation of the origin of the
 5723 record prior storing this record in the audit trail.

5724 **C.8.4.2 Operations**5725 **C.8.4.2.1 Assignment**

5726 In FAU_STG.3.1, the PP/ST author **should** specify the metric that the TSF must ensure with
 5727 respect to the stored audit records. This metric limits the data loss by enumerating the number
 5728 of records that must be kept, or the time that records are guaranteed to be maintained.

EXAMPLE

An example of the metric **could** be “100,000” indicating that 100,000 audit records **can** be stored.

5729 **C.8.4.2.2 Selection**

5730 In FAU_STG.3.1, the PP/ST author **should** specify the condition under which the TSF **shall** still be
 5731 able to maintain a defined amount of audit data. This condition **can** be any of the following:
 5732 audit storage exhaustion, failure, attack.

5733 **C.8.5 FAU_STG.4 Prevention of audit data loss**5734 **C.8.5.1 User application notes**

5735 This component specifies the behaviour of the TOE if the audit trail is full: either audit records
 5736 are ignored, or the TOE is frozen such that no audited events **can** take place. The requirement
 5737 also states that no matter how the requirement is instantiated, the authorized user with specific
 5738 rights to this effect, **can** continue to generate audited events (actions). The reason is that
 5739 otherwise the authorized user **could** not even reset the TOE. Consideration **should** be given to
 5740 the choice of the action to be taken by the TSF in the case of audit storage exhaustion, as
 5741 ignoring events, which provides better availability of the TOE, will also permit actions to be
 5742 performed without being recorded and without the user being accountable.

5743 **C.8.5.2 Operations**5744 **C.8.5.2.1 Selection**

5745 In FAU_STG.5.1, the PP/ST author **should** select whether the TSF **shall** ignore audited actions, or
 5746 whether it **should** prevent audited actions from happening, or whether the oldest audit records
 5747 **should** be overwritten when the TSF **can** no longer store audit records. Only one of these
 5748 options **may** be chosen.

5749 **C.8.5.2.2 Assignment**

5750 In FAU_STG.5.1, the PP/ST author **should** specify other actions that **should** be taken in case of
 5751 audit storage failure, such as informing the authorized user. If there is no other action to be
 5752 taken in case of audit storage failure, this assignment **can** be completed with “none”.

5753 **C.8.6 FAU_STG.5 Action in case of possible audit data loss**5754 **C.8.6.1 User application notes**

5755 This component requires that actions will be taken when the audit trail exceeds certain pre-
 5756 defined limits.

5757 **C.8.6.2 Operations**5758 **C.8.6.2.1 Assignment**

5759 In FAU_STG.5 Prevention of audit data loss, the PP/ST author **should** indicate the pre-defined
 5760 limit. If the management functions indicate that this number might be changed by the
 5761 authorized user, this value is the default value. The PP/ST author might choose to let the
 5762 authorized user define this limit.

EXAMPLE

In the case that an authorized user defines the limit, an example of the assignment can be “an authorized user set limit”.

5763 In FAU_STG.5 Prevention of audit data loss,, the PP/ST author **should** specify actions that **should**
5764 be taken in case of imminent audit storage failure indicated by exceeding the threshold. Actions
5765 might include informing an authorized user.

Annex D (normative)

Class FCO: Communication- application notes

5766
5767
5768
5769

5770 D.1 General information

5771 This class describes requirements specifically of interest for TOEs that are used for the
5772 transport of information. Families within this class deal with non-repudiation.

5773 In this class, the concept of “information” is used. This information **should** be interpreted as the
5774 object being communicated, and **could** contain an electronic mail message, a file, or a set of
5775 predefined attribute types.

5776 In the literature, the terms “proof of receipt” and “proof of origin” are commonly used terms.
5777 However, it is recognized that the term “proof” might be interpreted in a legal sense to imply a
5778 form of mathematical rationale. The components in this class interpret the de-facto use of the
5779 word “proof” in the context of “evidence” that the TSF demonstrates the non-repudiated
5780 transport of types of information.

5781 D.2 Non-repudiation of origin (FCO_NRO)

5782 D.2.1 User notes

5783 Non-repudiation of origin defines requirements to provide evidence to users/subjects about the
5784 identity of the originator of some information. The originator cannot successfully deny having
5785 sent the information because evidence of origin provides evidence of the binding between the
5786 originator and the information sent. The recipient or a third party **can** verify the evidence of
5787 origin. This evidence **should** not be forgeable.

EXAMPLE 1

Evidence of origin could be a digital signature

5788 If the information or the associated attributes are altered in any way, validation of the evidence
5789 of origin might fail. Therefore, a PP/ST author **should** consider including integrity requirements
5790 such as FDP_UIT.1 Data exchange integrity in the PP/ST.

5791 In non-repudiation, there are several different roles involved, each of which **could** be combined
5792 in one or more subjects. The first role is a subject that requests evidence of origin (only in
5793 FCO_NRO.1 Selective proof of origin). The second role is the recipient and/or other subjects to
5794 which the evidence is provided. The third role is a subject that requests verification of the
5795 evidence of origin.

EXAMPLE 2

Subject that requests evidence of origin: a recipient or a third party such as an arbiter.

Subject to which the evidence is provided: A notary

5796 The PP/ST author must specify the conditions that must be met to be able to verify the validity
5797 of the evidence.

EXAMPLE 3

An example of a condition which **could** be specified is where the verification of evidence
must occur within 24 hours.

5798 These conditions, therefore, allow the tailoring of the non-repudiation to legal requirements,
5799 such as being able to provide evidence for several years.

5800 In most cases, the identity of the recipient will be the identity of the user who received the
5801 transmission. In some instances, the PP/ST author does not want the user identity to be

5802 exported. In that case, the PP/ST author must consider whether it is appropriate to include this
 5803 class, or whether the identity of the transport service provider or the identity of the host **should**
 5804 be used.

5805 In addition to (or instead of) the user identity, a PP/ST author might be more concerned about
 5806 the time the information was transmitted.

EXAMPLE 4

For example, requests for proposals must be transmitted before a certain date in order to be considered.

5807 In such instances, these requirements **can** be customized to provide a timestamp indication
 5808 (time of origin).

5809 D.2.2 FCO_NRO.1 Selective proof of origin

5810 D.2.2.1 Operations

5811 D.2.2.1.1 Assignment

5812 In FCO_NRO.1.1, the PP/ST author **should** fill in the types of information subject to the evidence
 5813 of origin function.

EXAMPLE

An example of the type of information is “electronic mail messages”

5814

5815 D.2.2.1.2 Selection

5816 In FCO_NRO.1.1, the PP/ST author **should** specify the user/subject who **can** request evidence of
 5817 origin.

5818 D.2.2.1.3 Assignment

5819 In FCO_NRO.1.1, the PP/ST author, dependent on the selection, **should** specify the third parties
 5820 that **can** request evidence of origin.

EXAMPLE 1

A third party **could** be an arbiter, judge, or legal body.

5821 In FCO_NRO.1.2, the PP/ST author **should** fill in the list of the attributes that **shall** be linked to
 5822 the information;

EXAMPLE 2

Attributes include originator identity, time of origin, and location of origin.

5823 In FCO_NRO.1.2, the PP/ST author **should** fill in the list of information fields within the
 5824 information over which the attributes provide evidence of origin, such as the body of a message.

5825 D.2.2.1.4 Selection

5826 In FCO_NRO.1.3, the PP/ST author **should** specify the user/subject who **can** verify the evidence
 5827 of origin.

5828 D.2.2.1.5 Assignment

5829 In FCO_NRO.1.3, the PP/ST author **should** fill in the list of limitations under which the evidence
 5830 **can** be verified.

EXAMPLE

An example of a limitation is “the evidence **can** only be verified within a 24-hour time interval.”

5831 An assignment of “immediate” or “indefinite” is acceptable.

5832 In FCO_NRO.1.3, the PP/ST author, dependent on the selection, **should** specify the third parties
5833 that **can** verify the evidence of origin.

5834 **D.2.3 FCO_NRO.2 Enforced proof of origin**

5835 **D.2.3.1 Operations**

5836 **D.2.3.1.1 Assignment**

5837 In FCO_NRO.2.1, the PP/ST author **should** fill in the types of information subject to the evidence
5838 of origin function.

EXAMPLE
electronic mail messages.

5839 In FCO_NRO.2.2, the PP/ST author **should** fill in the list of the attributes that **shall** be linked to
5840 the information; for example, originator identity, time of origin, and location of origin.

5841 In FCO_NRO.2.2, the PP/ST author **should** fill in the list of information fields within the
5842 information over which the attributes provide evidence of origin, such as the body of a message.

5843 **D.2.3.1.2 Selection**

5844 In FCO_NRO.2.3, the PP/ST author **should** specify the user/subject who **can** verify the evidence
5845 of origin.

5846 **D.2.3.1.3 Assignment**

5847 In FCO_NRO.2.3, the PP/ST author **should** fill in the list of limitations under which the evidence
5848 **can** be verified.

EXAMPLE
For example, the evidence **can** only be verified within a 24-hour time interval.

5849 An assignment of “immediate” or “indefinite” is acceptable.

5850 In FCO_NRO.2.3, the PP/ST author, dependent on the selection, **should** specify the third parties
5851 that **can** verify the evidence of origin.

EXAMPLE 2
A third party **could** be an arbiter, judge, or legal body.

5852 **D.3 Non-repudiation of receipt (FCO_NRR)**

5853 **D.3.1 User notes**

5854 Non-repudiation of receipt defines requirements to provide evidence to other users/subjects
5855 that the information was received by the recipient. The recipient cannot successfully deny
5856 having received the information because evidence of receipt provides evidence of the binding
5857 between the recipient attributes and the information. The originator or a third party **can** verify
5858 the evidence of receipt. This evidence **should** not be forgeable.

EXAMPLE 1
An example of a receipt is a digital signature

5859 It **should** be noted that the provision of evidence that the information was received does not
5860 necessarily imply that the information was read or comprehended, but only delivered.

5861 If the information or the associated attributes are altered in any way, validation of the evidence
5862 of receipt with respect to the original information might fail. Therefore, a PP/ST author **should**
5863 consider including integrity requirements such as FDP_UIT.1 Data exchange integrity in the
5864 PP/ST.

5865 In non-repudiation, there are several different roles involved, each of which **could** be combined
 5866 in one or more subjects. The first role is a subject that requests evidence of receipt (only in
 5867 FCO_NRR.1 Selective proof of receipt). The second role is the recipient and/or other subjects to
 5868 which the evidence is provided). The third role is a subject that requests verification of the
 5869 evidence of receipt, for example, an originator or a third party such as an arbiter.

EXAMPLE 2

A recipient or subject **could** be a notary.

5870

5871 The PP/ST author must specify the conditions that must be met to be able to verify the validity
 5872 of the evidence. An example of a condition which **could** be specified is where the verification of
 5873 evidence must occur within 24 hours. These conditions, therefore, allow the tailoring of the
 5874 non-repudiation to legal requirements, such as being able to provide evidence for several years.

5875 In most cases, the identity of the recipient will be the identity of the user who received the
 5876 transmission. In some instances, the PP/ST author does not want the user identity to be
 5877 exported. In that case, the PP/ST author must consider whether it is appropriate to include this
 5878 class, or whether the identity of the transport service provider or the identity of the host **should**
 5879 be used.

5880 In addition to (or instead of) the user identity, a PP/ST author might be more concerned about
 5881 the time the information was received.

EXAMPLE 3

When an offer expires at a certain date, orders must be received before a certain date in order to be considered.

5882 In such instances, these requirements **can** be customized to provide a timestamp indication
 5883 (time of receipt).

5884 D.3.2 FCO_NRR.1 Selective proof of receipt

5885 D.3.2.1 Operations

5886 D.3.2.1.1 Assignment

5887 In FCO_NRR.1.1, the PP/ST author **should** fill in the types of information subject to the evidence
 5888 of receipt function, for example, electronic mail messages.

5889 D.3.2.1.2 Selection

5890 In FCO_NRR.1.1, the PP/ST author **should** specify the user/subject who **can** request evidence of
 5891 receipt.

5892 D.3.2.1.3 Assignment

5893 In FCO_NRR.1.1, the PP/ST author, dependent on the selection, **should** specify the third parties
 5894 that **can** request evidence of receipt.

EXAMPLE

A third party **could** be an arbiter, judge, or legal body.

5895 In FCO_NRR.1.2, the PP/ST author **should** fill in the list of the attributes that **shall** be linked to
 5896 the information; for example, recipient identity, time of receipt, and location of receipt.

5897 In FCO_NRR.1.2, the PP/ST author **should** fill in the list of information fields with the fields
 5898 within the information over which the attributes provide evidence of receipt, such as the body a
 5899 message.

5900 D.3.2.1.4 Selection

5901 In FCO_NRR.1.3, the PP/ST author **should** specify the user/subjects who **can** verify the evidence
 5902 of receipt.

5903 **D.3.2.1.5 Assignment**

5904 In FCO_NRR.1.3, the PP/ST author **should** fill in the list of limitations under which the evidence
5905 **can** be verified. For example, the evidence **can** only be verified within a 24-hour time interval.
5906 An assignment of “immediate” or “indefinite” is acceptable.

5907 In FCO_NRR.1.3, the PP/ST author, dependent on the selection, **should** specify the third parties
5908 that **can** verify the evidence of receipt.

5909 **D.3.3 FCO_NRR.2 Enforced proof of receipt**

5910 **D.3.3.1 Operations**

5911 **D.3.3.1.1 Assignment**

5912 In FCO_NRR.2.1, the PP/ST author **should** fill in the types of information subject to the evidence
5913 of receipt function,

EXAMPLE 1

For example, electronic mail messages.

5914 In FCO_NRR.2.2, the PP/ST author **should** fill in the list of the attributes that **shall** be linked to
5915 the information;

EXAMPLE 2

For example, recipient identity, time of receipt, and location of receipt.

5916 In FCO_NRR.2.2, the PP/ST author **should** fill in the list of information fields with the fields
5917 within the information over which the attributes provide evidence of receipt, such as the body
5918 of a message.

5919 **D.3.3.1.2 Selection**

5920 In FCO_NRR.2.3, the PP/ST author **should** specify the user/subjects who **can** verify the evidence
5921 of receipt.

5922 **D.3.3.1.3 Assignment**

5923 In FCO_NRR.2.3, the PP/ST author **should** fill in the list of limitations under which the evidence
5924 **can** be verified. An assignment of “immediate” or “indefinite” is acceptable.

EXAMPLE

For example, the evidence **can** only be verified within a 24-hour time interval.

5925 In FCO_NRR.2.3, the PP/ST author, dependent on the selection, **should** specify the third parties
5926 that **can** verify the evidence of receipt. A third party **could** be an arbiter, judge or legal body.

5927
5928
5929
5930

Annex E (normative)

Class FCS: Cryptographic support- application notes

5931 E.1 General information

5932 The TSF **may** employ cryptographic functionality to help satisfy several high-level security
5933 objectives. These include (but are not limited to): identification and authentication, non-
5934 repudiation, trusted path, trusted channel and data separation. This class is used when the TOE
5935 implements cryptographic functions, the implementation of which **could** be in hardware,
5936 firmware and/or software.

5937 The FCS: Cryptographic support class is composed of four families: Cryptographic key
5938 management (FCS_CKM), Cryptographic operation (FCS_COP), Random bit generation
5939 (FCS_RBG), and Generation of random numbers (FCS_RNG).

5940 The Cryptographic key management (FCS_CKM) family addresses the management aspects of
5941 cryptographic keys; the Cryptographic operation (FCS_COP) family is concerned with the
5942 operational use of those cryptographic keys; the Random bit generation (FCS_RBG) family
5943 provides requirements for generating random bits; and the Generation of random numbers
5944 (FCS_RNG) is concerned with ensuring that random numbers meet defined quality metrics.

5945 For each cryptographic key generation method implemented by the TOE, if any, the PP/ST
5946 author **should** select the FCS_CKM.1 Cryptographic key generation component.

5947 For each cryptographic key distribution method implemented by the TOE, if any, the PP/ST
5948 author **should** select the FCS_CKM.2 Cryptographic key distribution.

5949 For each cryptographic key access method implemented by the TOE, if any, the PP/ST author
5950 **should** select the FCS_CKM.3 Cryptographic key access.

5951 For each cryptographic key derivation method implemented by the TOE, if any, the PP/ST
5952 author **should** select the FCS_CKM.5 Cryptographic key derivation.

5953 For each cryptographic key destruction method implemented by the TOE, if any, the PP/ST
5954 author **should** select the FCS_CKM.6 Timing and event of cryptographic key destruction
5955 component.

5956 For each cryptographic operation (such as digital signature, data encryption, key agreement,
5957 secure hash, etc.) performed by the TOE, if any, the PP/ST author **should** select the FCS_COP.1
5958 Cryptographic operation component.

5959 For each deterministic random bit generation service implemented by the TOE, if any, the
5960 PP/ST author **should** select the FCS_RBG.1 Random bit generation (RBG) component.

5961 For each external seeding source used by the TOE, if any, the PP/ST author **should** select the
5962 FCS_RBG.2 Random bit generation (external seeding) component.

5963 For each internal seeding source (single) used by the TOE, if any, the PP/ST author **should**
5964 select the FCS_RBG.3 Random bit generation (internal seeding – single source) component.

5965 Where internal seeding source (multiple) is to be specified, the PP/ST author **should** select the
5966 FCS_RBG.4 Random bit generation (internal seeding – multiple sources) component.

5967 For cases where the TOE combines entropy sources, the FCS_RBG.5 Random bit generation
5968 (combining entropy sources) component **should** be specified by PP/ST author.

5969 For each random bit generation service implemented by the TOE, the PP/ST author **should**
5970 specify the FCS_RBG.6 Random bit generation service component.

5971 For each random number generation service implemented by the TOE, the PP/ST author **should**
 5972 specify the FCS_RNG.1 Random number generation component.

5973 Cryptographic functionality **may** be used to meet objectives specified in class FCO:
 5974 Communication, and in families Data authentication (FDP_DAU), Stored data integrity
 5975 (FDP_SDI), Inter-TSF user data confidentiality transfer protection (FDP_UCT), Inter-TSF user
 5976 data integrity transfer protection (FDP_UIT), Specification of secrets (FIA_SOS), User
 5977 authentication (FIA_UAU), to meet a variety of objectives. In the cases where cryptographic
 5978 functionality is used to meet objectives for other classes, the individual functional components
 5979 specify the objectives that cryptographic functionality must satisfy. The objectives in class FCS:
 5980 Cryptographic support **should** be used when cryptographic functionality of the TOE is sought by
 5981 consumers.

5982 **E.2 Cryptographic key management (FCS_CKM)**

5983 **E.2.1 User notes**

5984 Cryptographic keys must be managed throughout their lifetime. The typical events in the
 5985 lifecycle of a cryptographic key include but are not limited to: key generation or derivation,
 5986 distribution, entry, storage, access, and destruction.

EXAMPLE 1

- backup
- escrow
- archive
- recovery

5987 The inclusion of other stages is dependent on the key management strategy being implemented,
 5988 as the TOE is not always involved in all of the key life-cycle phases.

EXAMPLE 2

The TOE **may** only generate and distribute cryptographic keys.

5989 This family is intended to support the cryptographic key lifecycle and consequently defines
 5990 requirements for the following activities: cryptographic key generation, cryptographic key
 5991 derivation, cryptographic key distribution, cryptographic key access, and cryptographic key
 5992 destruction. This family **should** be included whenever there are functional requirements for the
 5993 management of cryptographic keys.

5994 If Security audit data generation (FAU_GEN) is included in the PP/ST then, in the context of the
 5995 events being audited:

- 5996 a) The object attributes **may** include the assigned user for the cryptographic key, the
 5997 user role, the cryptographic operation that the cryptographic key is to be used for,
 5998 the cryptographic key identifier and the cryptographic key validity period.
- 5999 b) The object value **may** include the values of cryptographic key(s) and parameters
 6000 excluding any sensitive information (such as secret or private cryptographic keys).

6001 Typically, random numbers are used to generate cryptographic keys. If this is the case, then
 6002 FCS_CKM.1 Cryptographic key generation **should** be used instead of the component FIA_SOS.2
 6003 TSF Generation of secrets. In cases where random number generation is required for purposes
 6004 other than for the generation of cryptographic keys, the component FIA_SOS.2 TSF Generation
 6005 of secrets **should** be used.

6006 **E.2.2 FCS_CKM.1 Cryptographic key generation**

6007 **E.2.2.1 User application notes**

6008 This component requires the cryptographic key sizes and method used to generate
 6009 cryptographic keys to be specified, this **may** be in accordance with an assigned standard. It
 6010 **should** be used to specify the cryptographic key sizes and the method used to generate the
 6011 cryptographic keys. Only one instance of the component is needed for the same method and
 6012 multiple key sizes. The key size **may** be common or different for the various entities and **may** be
 6013 either the input to or the output from the method.

EXAMPLE

An example of a method is an algorithm.

6014 **E.2.2.2 Operations**

6015 **E.2.2.2.1 Assignment**

6016 In FCS_CKM.1.1, the PP/ST author **should** specify the cryptographic key generation algorithm to
 6017 be used.

6018 In FCS_CKM.1.1, the PP/ST author **should** specify the cryptographic key sizes to be used. The
 6019 key sizes specified **should** be appropriate for the algorithm and its intended use.

6020 In FCS_CKM.1.1, the PP/ST author **should** specify the assigned standard that documents the
 6021 method used to generate cryptographic keys. The assigned standard **may** comprise none, one or
 6022 more actual standards publications, for example, from international, national, industry or
 6023 organizational standards.

6024 **E.2.3 FCS_CKM.2 Cryptographic key distribution**

6025 **E.2.3.1 User application notes**

6026 This component requires the method used to distribute cryptographic keys to be specified, this
 6027 **may** be in accordance with an assigned standard. See ISO/IEC 15408-1 for information on using
 6028 standards in PPs and STs.

6029 **E.2.3.2 Operations**

6030 **E.2.3.2.1 Assignment**

6031 In FCS_CKM.2.1 the PP/ST author **should** specify the cryptographic key distribution method to
 6032 be used.

6033 In FCS_CKM.2.1 the PP/ST author **should** specify the assigned standard that documents the
 6034 method used to distribute cryptographic keys. The assigned standard **may** comprise none, one
 6035 or more actual standards publications, for example, from international, national, industry or
 6036 organizational standards.

6037 **E.2.4 FCS_CKM.3 Cryptographic key access**

6038 **E.2.4.1 User application notes**

6039 This component requires the method used to access cryptographic keys be specified, this **may**
 6040 be in accordance with an assigned standard.

6041 **E.2.4.2 Operations**

6042 **E.2.4.2.1 Assignment**

6043 In FCS_CKM.2.1, the PP/ST author **should** specify the type of cryptographic key access being
 6044 used.

EXAMPLE

Examples of types of cryptographic key access include (but are not limited to)
 cryptographic key backup, cryptographic key archival, cryptographic key escrow, and
 cryptographic key recovery.

6045 In FCS_CKM.2.1, the PP/ST author **should** specify the cryptographic key access method to be
 6046 used.

6047 In FCS_CKM.2.1, the PP/ST author **should** specify the assigned standard that documents the
 6048 method used to access cryptographic keys. The assigned standard **may** comprise none, one or
 6049 more actual standards publications, for example, from international, national, industry or
 6050 organizational standards.

6051 **E.2.5 FCS_CKM.5 Cryptographic key derivation**

6052 **E.2.5.1 User application notes**

6053 Table E.1 **should** be used when completing and potentially iterating the FCS-CKM.5 component.
 6054 Each row, which can be identified using the “Identifier”, provides a set of recommended
 6055 selections and assignments for completing FCS-CKM.5 for each commonly used key type.

6056 **Table E.1 — Recommended selections and assignments for key derivation**

Identifier	key type	input parameters	key derivation algorithm	key sizes	list of standards
KeyDrv1	[assignment: key name]	Direct Generation from a Random Bit Generator as specified in FCS_RBG.1	KDF in Counter Mode using [selection: CMAC-AES-128, CMAC-AES-256, HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512] as the PRF	[selection: 128, 256] bits	NIST SP 800-108 (October 2009) (Section 5.1) [KDF in Counter Mode] [selection: ISO/IEC 9797-1:2011 (Clause B.6, B7), NIST SP 800-38B(June 2016) [CMAC] ISO/IEC 18033-3:2010 (Subclause 5.2) [AES], ISO/IEC 9797-2:2011 (Clause 7 MAC Algorithm 2 (HMAC)), FIPS 198-1 July 2008, ISO 10118-3:2004, (Clause 10, 11); FIPS 180-4 August 2015, (Section 6) [SHA]]
KeyDrv2	[assignment: key name]	Direct Generation from a Random Bit Generator as specified in FCS_RBG.1	KDF in Feedback Mode using [selection: CMAC-AES-128, CMAC-AES-192, CMAC-AES-256, HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512] as the PRF	[selection: 128, 256] bits	NIST SP 800-108 October, 2009 (Section 5.2) [KDF in Feedback Mode] [selection: ISO/IEC 9797-1:2011 (Subclause 7.6), NIST SP 800-38B June 2016 [CMAC], ISO/IEC 18033-3:2010 (Subclause 5.2) [AES], ISO/IEC 9797-2:2011 (Clause 7 MAC Algorithm 2 (HMAC)),

Identifier	key type	input parameters	key derivation algorithm	key sizes	list of standards
					FIPS 198-1 July 2008, ISO/IEC 10118-3, (Clause 10, 11); FIPS 180-4 August 2015, (Section 6) [SHA]]
KeyDrv3	[assignment: key name]	Direct Generation from a Random Bit Generator as specified in FCS_RBG.1	KDF in Double-Pipeline Iteration Mode using [selection: CMAC-AES-128, CMAC-AES-256, HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512] as the PRF	[selection: 128, 256] bits	NIST SP 800-108 October, 2009 (Section 5.3) [KDF in Double-Pipeline Iteration Mode] [selection: ISO/IEC 9797-1:2011 (Subclause 7.6), NIST SP 800-38B June 2016 [CMAC] ISO/IEC 18033-3:2010 (subclause 5.2) [AES], ISO/IEC 9797-2:2011 (Clause 7 MAC Algorithm 2 (HMAC)), FIPS 198-1 July 2008, ISO/IEC 10118-3:2018, (Clause 10, 11); FIPS 180-4 August 2015, (Section 6) [SHA]]
KeyDrv4	Authorization Factor Submask	Password Salt: generated from a Random Bit Generator as specified in FCS_RBG.1	PBKDF using HMAC- [selection: SHA-1, SHA-256, SHA-512] as the PRF, with [assignment: positive integer of 1000 or more] iterations	[selection: 128, 256] bits	NIST SP 800-132 December 2010
KeyDrv5	[assignment: key name]	Intermediary keys	[selection: exclusive OR (XOR), SHA-256, SHA-512]	[selection: 128, 256] bits	[selection: ISO 10118-3:2004, (Clause 10, 11); FIPS180-4, August 2015 (Section 6) [SHA]]

6057

6058

6059

NOTE For identifier KeyDrv4, The key size to be used in the HMAC falls into a range between L1 and L2 defined in ISO/IEC 10118 for the appropriate hash function (for example for SHA-256 L1 = 512, L2 =256) where $L2 \leq k \leq L1$.

6060

6061

Editors' Note:

6062

The editors have amended the final column in the above table in order to specify the dates of the standards cited. This is an ISO requirement when referencing specific sections/clauses etc within a standard.

6063

6064

6065

The editors have researched the current versions of the standards and presented that information here. If older versions of the standards were intended, then editors request that SMEs please comment accordingly.

6066

6067

6068

6069

The editors have changed references in column 2 to FCS_RBG_EXT.1 to FCS_RBG.1 (To reference the SFR in this part 2,)

6070

6071

6072

In the KeyDRV 4 row "Salt: using a salt as specified in FCS_SLT_EXT.1" is problematic since there os no FCS_SLT.1 in part 2 . Editors have amended this to read "

6073

6074

"Salt: generated from a Random Bit Generator as specified in FCS_RBG.1"

6075

EXAMPLE

To derive a component or SFR from the FCS_CKM.5 component for Intermediary keys, the row identified as KeyDrv 5 in Table E.1 is used.

Using this information, the following component is generated:

The TSF shall derive cryptographic keys [assignment: *key type*] from [Intermediary keys] in accordance with a specified cryptographic key derivation algorithm [**selection: exclusive OR (XOR), SHA-256, SHA-512**] and specified cryptographic key sizes [**selection: 128, 256 bits**] that meet the following: [**selection: ISO 10118-3, (Clause 10, 11); FIPS180-4, (Section 6) [SHA]**].

This component can then be used in PPs or completed and used as an SFR in PPs and STs, as appropriate.

6076

E.2.5.2 Evaluator notes

6077

Evaluators should refer to ISO/IEC 15408-1:20XX, Annex B.4.8 and C.2.9 for information in regard to the evaluation of standards specified in FCS_CKM.5.

6078

6079

E.2.5.3 Operations

6080

E.2.5.3.1 Assignment

6081

See E.2.5.1.

6082

E.2.5.3.2 Selection

6083

See E.2.5.1.

6084

E.2.6 FCS_CKM.6 Timing and event of cryptographic key destruction

6085

E.2.6.1 User application notes

6086

This component requires the list of keys, including any keying material and the method used to destroy cryptographic keys to be specified, this can be in accordance with an assigned standard.

6087

6088

6089

NOTE Keying material includes keys and initialization vectors necessary to establish and maintain cryptographic keying relationships

6090

E.2.6.2 Operations

6091

E.2.6.2.1 Assignment

6092

6093 In FCS_CKM.6 Timing and event of cryptographic key destruction, the PP/ST author provides a
 6094 list of cryptographic keys and keying material that should be destroyed under certain
 6095 circumstances.

6096 **E.2.6.2.2 Selection**

6097 **E.3 Cryptographic operation (FCS_COP)**

6098 **E.3.1 User notes**

6099 A cryptographic operation **may** have cryptographic mode(s) of operation associated with it. If
 6100 this is the case, then the cryptographic mode(s) must be specified.

EXAMPLE

Examples of cryptographic modes of operation are cipher block chaining, output feedback mode, electronic code book mode, and cipher feedback mode.

6101 Cryptographic operations **may** be used to support one or more TOE security services. The
 6102 Cryptographic operation (FCS_COP) component **may** need to be iterated more than once
 6103 depending on:

- 6104 a) the user application for which the security service is being used,
- 6105 b) the use of different cryptographic algorithms and/or cryptographic key sizes,
- 6106 c) the type or sensitivity of the data being operated on.

6107 If Security audit data generation (FAU_GEN) Security audit data generation is included in the
 6108 PP/ST then, in the context of the cryptographic operation events being audited:

- 6109 a) The types of cryptographic operation **may** include digital signature generation
 6110 and/or verification, cryptographic checksum generation for integrity and/or for
 6111 verification of checksum, secure hash (message digest) computation, data
 6112 encryption and/or decryption, cryptographic key encryption and/or decryption,
 6113 cryptographic key agreement, and random number generation.
- 6114 b) The subject attributes **may** include subject role(s) and user(s) associated with the
 6115 subject.
- 6116 c) The object attributes **may** include the assigned user for the cryptographic key, user
 6117 role, cryptographic operation the cryptographic key is to be used for, cryptographic
 6118 key identifier, and the cryptographic key validity period.

6119 **E.3.2 FCS_COP.1 Cryptographic operation**

6120 **E.3.2.1 User application notes**

6121 This component requires the cryptographic algorithm and key size used to perform specified
 6122 cryptographic operation(s) which **can** be based on an assigned standard.

6123 The dependencies to FCS_RBG.1 or FCS_RNG.1 will be required for cryptographic algorithm
 6124 operations which internally generate random numbers.

EXAMPLE

DSA signature generation, ECDSA signature generation, RSASSA-PSS signature generation.

6125 The dependencies to FCS_RBG.1 or FCS_RNG.1 may not be necessary for deterministic
 6126 cryptographic algorithm operations.

EXAMPLE

AES encryption / decryption in ECB mode.

6127 **E.3.2.2 Operations**

6128 **E.3.2.2.1 Assignment**

6129 In FCS_COP.1.1, the PP/ST author specifies the cryptographic operations being performed.
6130 Typical cryptographic operations include digital signature generation and/or verification,
6131 cryptographic checksum generation for integrity and/or for verification of checksum, secure
6132 hash (message digest) computation, data encryption and/or decryption, cryptographic key
6133 encryption and/or decryption, cryptographic key agreement, and random number generation.
6134 The cryptographic operation **may** be performed on user data or TSF data.

6135 In FCS_COP.1.1, the PP/ST author **should** specify the cryptographic algorithm to be used.

EXAMPLE

Examples of typical cryptographic algorithms include, but are not limited to, DES, RSA and IDEA.

6136 In FCS_COP.1.1, the PP/ST author **should** specify the cryptographic key sizes to be used. The key
6137 sizes specified **should** be appropriate for the algorithm and its intended use.

6138 In FCS_COP.1.1, the PP/ST author **should** specify the assigned standard that documents how the
6139 identified cryptographic operation(s) are performed. The assigned standard **may** comprise
6140 none, one or more actual standards publications, these **may** include standards from
6141 international, national, industry or organizational standards.

6142 **E.4 Random bit generation (FCS_RBG)**

6143 **E.4.1 User notes**

6144 **E.4.2 FCS_RBG.1 Random bit generation (RBG)**

6145 **E.4.2.1 User application notes**

6146 For FCS_RBG.1 These dependencies **shall** always be met.

6147 NOTE ISO/IEC 15408-1:20XX 7.3 item c) allowing a justification to be provided if a dependency is not met is not
6148 allowed for this component.

6149 In the RBG State Update Table the ST author **must** include a row for initialization (Source1).
6150 Other rows are optional, depending on the noise sources supported by the TSF. The identifier
6151 values identify the specific source, so there should be a row for every unique source, and if the
6152 same source is used for more than one update type then the same identifier is given.

6153 If reseeding is not feasible, the TSF will unstantiate RBGs (and instantiate a new RBG), rather
6154 than produce output that is of insufficient quality. The listed standards should specify the
6155 reseed interval, and procedure for uninstantiating and reseeding. The 'Condition' selection
6156 allows the PP Author to require application-specific conditions for reseeding.

6157 "Unstantiate" means that the internal state of the DRBG is no longer available for use.

6158 In the 'Condition' selection, "on demand" means, that an interface to reseed is presented as a
6159 TSFI.

EXAMPLE

An example of an interface is an API call.

6160 Health tests for the RBG are specified in FPT_TST.1.

6161 **E.4.2.2 Operations**

6162 **E.4.2.2.1 Selection**

6163 **E.4.2.2.2 Assignment**

6164 **E.4.3 FCS_RBG.2 Random bit generation (external seeding)**

6165 **E.4.3.1 User application notes**

6166 For this component, the interface to obtain the entropy noise source can be used multiple times
6167 to provide input. For instance, if the input length is 128 bits, it could be used twice to gather 256

6168 bits. In this instance, the 128 bits would not be provided to the DRBG, since the DRBG can only
 6169 be instantiated once, rather a function would gather the 128 bits twice and provide the DRBG
 6170 with 256 bits of entropy noise source.

6171 This component does not describe requirements on seed quality: it is the responsibility of the
 6172 operational environment to define their requirement in this regard and to ensure that it is met
 6173 by the external source.

6174 Guidance in the introduction to PP/ST authors should address protection from modification and
 6175 disclosure of the value from the external noise source, as well as the leaking of any pertinent
 6176 information (e.g., internal state) regarding the RBG.

6177 **Editors' Note**

6178 Please provide an exact reference to what is meant by "Guidance in the introduction to PP/ST authors".

6179 Does it mean the "Introduction Section" of the PP/ST? In that case a reference would be See ISO/IEC
 6180 15408-1:20XX, B.2.2.1

6181 **E.4.3.2 Operations**

6182 **E.4.3.2.1 Selection**

6183 **E.4.3.2.2 Assignment**

6184 **E.4.4 FCS_RBG.3 Random bit generation (internal seeding – single source)**

6185 **E.4.4.1 User application notes**

6186 If an ST Author wishes to use multiple internal noise sources, they iterate this requirement for
 6187 each noise source being used by the TSF.

6188 Hardware-based noise sources are sources whose primary function is noise generation, such as
 6189 ring oscillators, diodes, and thermal noise. While software is used to collect the noise from these
 6190 hardware sources, these are not software-based. Software-based noise sources are those
 6191 sources that have some other primary function and the noise is a byproduct of their normal
 6192 operation. Examples of software-based noise sources are user or system-based events, reading
 6193 the least significant bits from an event timer, etc.

6194 Hardware-based noise sources may be stochastically modeled, in which case the amount of
 6195 entropy is well understood. Software-based noise sources are usually less well understood and
 6196 therefore will typically take a more conservative approach, gathering larger numbers of bits
 6197 than required and then performing a compression function to derive the final output. Software-
 6198 based noise sources often rely on an entropy estimator.

6199 **E.4.4.2 Operations**

6200 **E.4.4.2.1 Selection**

6201 **E.4.4.2.2 Assignment**

6202 **E.4.5 FCS_RBG.4 Random bit generation**

6203 **E.4.5.1 User application notes**

6204 **E.4.5.2 Operations**

6205 **E.4.5.2.1 Selection**

6206 **E.4.5.2.2 Assignment**

6207 **E.4.6 FCS_RBG.5 Random bit generation**

6208 **E.4.6.1 User application notes**

6209 **E.4.6.2 Operations**

6210 **E.4.6.2.1 Selection**

6211 **E.4.6.2.2 Assignment**

6212 **E.4.7 FCS_RBG.6 Random bit generation service**

6213 **E.4.7.1 User application notes**

6214 **E.4.7.2 Operations**

6215 **E.4.7.2.1 Selection**

6216 **E.4.7.2.2 Assignment**

6217 **E.5 Generation of random numbers (FCS_RNG)**

6218 **Editors' note**

6219 **Editors are still waiting for contribution from the CCDB Crypto Working Group**

6220 **E.5.1 User notes**

6221 **E.5.2 FCS_RNG.1 Random number generation**

6222 **E.5.2.1 User application notes**

EXAMPLE

In some cases, the number "42" is random.

6223

6224 **E.5.2.2 Operations**

6225 **E.5.2.2.1 Selection**

6226 In FCS_RNG.1 .1 the PP/ST author **should**

6227 **E.5.2.2.2 Assignment**

6228 In FCS_RNG.1 .1 the PP/ST author **should**

6229
6230
6231
6232

Annex F (normative)

Class FDP: User data protection- application notes

6233 F.1 General information

6234 This class contains families specifying requirements related to protecting user data. This class
6235 differs from FIA and FPT in that FDP: User data protection specifies components to protect user
6236 data, FIA specifies components to protect attributes associated with the user, and FPT specifies
6237 components to protect TSF information.

6238 The class does not contain explicit requirements for traditional Mandatory Access Controls
6239 (MAC) or traditional Discretionary Access Controls (DAC); however, such requirements **may** be
6240 constructed using components from this class.

6241 FDP: User data protection does not explicitly deal with confidentiality, integrity, or availability,
6242 as all three are most often intertwined in the policy and mechanisms. However, the TOE
6243 security policy must adequately cover these three objectives in the PP/ST.

6244 A final aspect of this class is that it specifies access control in terms of “operations”. An
6245 operation is defined as a specific type of access on a specific object. It depends on the level of
6246 abstraction of the PP/ST author whether these operations are described as “read” and/or
6247 “write” operations, or as more complex operations such as “update the database”.

6248 The access control policies are policies that control access to the information container. The
6249 attributes represent attributes of the container. Once the information is out of the container, the
6250 accessor is free to modify that information, including writing the information into a different
6251 container with different attributes. By contrast, an information flow policy controls access to
6252 the information, independent of the container. The attributes of the information, which **may** be
6253 associated with the attributes of the container (or **may** not, as in the case of a multi-level
6254 database) stay with the information as it moves. The accessor does not have the ability, in the
6255 absence of an explicit authorization, to change the attributes of the information.

6256 This class is not meant to be a complete taxonomy of IT access policies, as others **can** be
6257 imagined. Those policies included here are simply those for which current experience with
6258 actual systems provides a basis for specifying requirements. There **may** be other forms of intent
6259 that are not captured in the definitions here.

EXAMPLE 1

For example, a goal of having user-imposed (and user-defined) controls on information flow (such as an automated implementation of the NO FOREIGN handling caveat).

6260 Such concepts **could** be handled as refinements of, or extensions to the FDP: User data
6261 protection components.

6262 Finally, it is important when looking at the components in FDP: User data protection to
6263 remember that these components are requirements for functions that **may** be implemented by a
6264 mechanism that also serves or **could** serve another purpose.

EXAMPLE 2

it is possible to build an access control policy (Access control policy (FDP_ACC)) that uses labels (FDP_IFF.1 Simple security attributes) as the basis of the access control mechanism.

6265 A set of SFRs **may** encompass many security function policies (SFPs), each to be identified by
6266 the two policy-oriented components Access control policy (FDP_ACC), and Information flow
6267 control policy (FDP_IFC). These policies will typically take confidentiality, integrity, and
6268 availability aspects into consideration as required, to satisfy the TOE requirements. Care **should**

6269 be taken to ensure that all objects are covered by at least one SFP and that there are no conflicts
6270 arising from implementing the multiple SFPs.

6271 When building a PP/ST using components from the FDP: User data protection class, the
6272 following information provides guidance on where to look and what to select from the class.

6273 The requirements in the FDP: User data protection class are defined in terms of a set of SFRs
6274 that will implement a SFP. Since a TOE **may** implement multiple SFPs simultaneously, the PP/ST
6275 author must specify the name for each SFP, so it **can** be referenced in other families. This name
6276 will then be used in each component selected to indicate that it is being used as part of the
6277 definition of requirements for that SFP. This allows the author to easily indicate the scope for
6278 operations such as objects covered, operations covered, authorized users, etc.

6279 Each instantiation of a component **can** apply to only one SFP. Therefore, if an SFP is specified in
6280 a component then this SFP will apply to all the elements in this component. The components
6281 **may** be instantiated multiple times within a PP/ST to account for different policies if so desired.

6282 The key to selecting components from this family is to have a well-defined set of TOE security
6283 objectives to enable proper selection of the components from the two policy components;
6284 Access control policy (FDP_ACC) and Information flow control policy (FDP_IFC). In Access
6285 control policy (FDP_ACC) and Information flow control policy (FDP_IFC) respectively, all access
6286 control policies and all information flow control policies are named. Furthermore, the scope of
6287 control of these components in terms of the subjects, objects and operations covered by this
6288 security functionality. The names of these policies are meant to be used throughout the
6289 remainder of the functional components that have an operation that calls for an assignment or
6290 selection of an “access control SFP” or an “information flow control SFP”. The rules that define
6291 the functionality of the named access control and information flow control SFPs will be defined
6292 in the Access control functions (FDP_ACF) and Information flow control functions (FDP_IFF)
6293 families (respectively).

6294 The following steps are guidance on how this class is applied in the construction of a PP/ST:

- 6295 a) Identify the policies to be enforced from the Access control policy (FDP_ACC), and
6296 Information flow control policy (FDP_IFC) families. These families define scope of
6297 control for the policy, granularity of control and **may** identify some rules to go with
6298 the policy.
- 6299 b) Identify the components and perform any applicable operations in the policy
6300 components. The assignment operations **may** be performed generally (such as with
6301 a statement “All files”) or specifically (“The files “A”, “B”, etc.) depending upon the
6302 level of detail known.
- 6303 c) Identify any applicable function components from the Access control functions
6304 (FDP_ACF) and Information flow control functions (FDP_IFF) families to address
6305 the named policy families from Access control policy (FDP_ACC) and Information
6306 flow control policy (FDP_IFC). Perform the operations to make the components
6307 define the rules to be enforced by the named policies. This **should** make the
6308 components fit the requirements of the selected function envisioned or to be built.
- 6309 d) Identify who will have the ability to control and change security attributes under
6310 the function, such as only a security administrator, only the owner of the object, etc.
6311 Select the appropriate components from FMT: Security management and perform
6312 the operations. Refinements **may** be useful here to identify missing features, such
6313 as that some or all changes must be done via trusted path.
- 6314 e) Identify any appropriate components from the FMT: Security management for
6315 initial values for new objects and subjects.
- 6316 f) Identify any applicable rollback components from the Rollback (FDP_ROL) family.
- 6317 g) Identify any applicable residual information protection requirements from the
6318 Residual information protection (FDP_RIP) family.

- 6319 h) Identify any applicable import or export components, and how security attributes
6320 **should** be handled during import and export, from the Import from outside of the
6321 TOE (FDP_ITC) and Export from the TOE (FDP_ETC) families.
- 6322 i) Identify any applicable internal TOE communication components from the Internal
6323 TOE transfer (FDP_ITT) family.
- 6324 j) Identify any requirements for integrity protection of stored information from the
6325 Stored data integrity (FDP_SDI).
- 6326 k) Identify any applicable inter-TSF communication components from the Inter-TSF
6327 user data confidentiality transfer protection (FDP_UCT) or Inter-TSF user data
6328 integrity transfer protection (FDP_UIT) families.

6329 **F.2 Access control policy (FDP_ACC)**

6330 **F.2.1 User notes**

6331 This family is based upon the concept of arbitrary controls on the interaction of subjects and
6332 objects. The scope and purpose of the controls is based upon the attributes of the accessor
6333 (subject), the attributes of the container being accessed (object), the actions (operations) and
6334 any associated access control rules.

6335 The components in this family are capable of identifying the access control SFPs (by name) to
6336 be enforced by the traditional Discretionary Access Control (DAC) mechanisms. It further
6337 defines the subjects, objects and operations that are covered by identified access control SFPs.
6338 The rules that define the functionality of an access control SFP will be defined by other families,
6339 such as Access control functions (FDP_ACF) and Export from the TOE (FDP_ETC). The names of
6340 the access control SFPs defined in Access control policy (FDP_ACC) are meant to be used
6341 throughout the remainder of the functional components that have an operation that calls for an
6342 assignment or selection of an “access control SFP.”

6343 The access control SFP covers a set of triplets: subject, object, and operations. Therefore, a
6344 subject **can** be covered by multiple access control SFPs but only with respect to a different
6345 operation or a different object. Of course, the same applies to objects and operations.

6346 A critical aspect of an access control function that enforces an access control SFP is the ability
6347 for users to modify the attributes involved in access control decisions. The Access control policy
6348 (FDP_ACC) family does not address these aspects. Some of these requirements are left
6349 undefined, but **can** be added as refinements, while others are covered elsewhere in other
6350 families and classes such as FMT: Security management.

6351 There are no audit requirements in Access control policy (FDP_ACC) as this family specifies
6352 access control SFP requirements. Audit requirements will be found in families specifying
6353 functions to satisfy the access control SFPs identified in this family.

6354 This family provides a PP/ST author the capability to specify several policies, for example, a
6355 fixed access control SFP to be applied to one scope of control, and a flexible access control SFP
6356 to be defined for a different scope of control. To specify more than one access control policy, the
6357 components from this family **can** be iterated multiple times in a PP/ST to different subsets of
6358 operations and objects. This will accommodate TOEs that contain multiple policies, each
6359 addressing a particular set of operations and objects. In other words, the PP/ST author **should**
6360 specify the required information in the ACC component for each of the access control SFPs that
6361 the TSF will enforce. For example, a TOE incorporating three access control SFPs, each covering
6362 only a subset of the objects, subjects, and operations within the TOE, will contain one
6363 FDP_ACC.1 Subset access control component for each of the three access-control SFPs,
6364 necessitating a total of three FDP_ACC.1 Subset access control components.

6365 **F.2.2 FDP_ACC.1 Subset access control**

6366 **F.2.2.1 User application notes**

6367 The terms object and subject refer to generic elements in the TOE. For a policy to be
6368 implementable, the entities must be clearly identified. For a PP, the objects and operations
6369 might be expressed as types such as: named objects, data repositories, observe accesses, etc.
6370 For a specific TOE these generic terms (subject, object) must be refined.

EXAMPLE

files, registers, ports, daemons, open calls, etc.

6371 This component specifies that the policy cover some well-defined set of operations on some
6372 subset of the objects. It places no constraints on any operations outside the set - including
6373 operations on objects for which other operations are controlled.

6374 **F.2.2.2 Operations**

6375 **F.2.2.2.1 Assignment**

6376 In FDP_ACC.1.1, the PP/ST author **should** specify a uniquely named access control SFP to be
6377 enforced by the TSF.

6378 In FDP_ACC.1.1, the PP/ST author **should** specify the list of subjects, objects, and operations
6379 among subjects and objects covered by the SFP.

6380 **F.2.3 FDP_ACC.2 Complete access control**

6381 **F.2.3.1 User application notes**

6382 This component requires that all possible operations on objects, that are included in the SFP,
6383 are covered by an access control SFP.

6384 The PP/ST author must demonstrate that each combination of objects and subjects is covered
6385 by an access control SFP.

6386 **F.2.3.2 Operations**

6387 **F.2.3.2.1 Assignment**

6388 In FDP_ACC.2.1, the PP/ST author **should** specify a uniquely named access control SFP to be
6389 enforced by the TSF.

6390 In FDP_ACC.2.1, the PP/ST author **should** specify the list of subjects and objects covered by the
6391 SFP. All operations among those subjects and objects will be covered by the SFP.

6392 **F.3 Access control functions (FDP_ACF)**

6393 **F.3.1 User notes**

6394 This family describes the rules for the specific functions that **can** implement an access control
6395 policy named in Access control policy (FDP_ACC) which also specifies the scope of control of the
6396 policy.

6397 This family provides a PP/ST author the capability to describe the rules for access control. This
6398 results in a TOE where the access to objects will not change. An example of such an object is
6399 "Message of the Day", which is readable by all, and changeable only by the authorized
6400 administrator. This family also provides the PP/ST author with the ability to describe rules that
6401 provide for exceptions to the general access control rules. Such exceptions would either
6402 explicitly allow or deny authorization to access an object.

6403 There are no explicit components to specify other possible functions such as two-person
6404 control, sequence rules for operations, or exclusion controls. However, these mechanisms, as
6405 well as traditional DAC mechanisms, **can** be represented with the existing components, by
6406 careful drafting of the access control rules.

6407 A variety of acceptable access control functionality **may** be specified in this family.

EXAMPLE

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

6408 **F.3.2 FDP_ACF.1 Security attribute based access control**6409 **F.3.2.1 User application notes**

6410 This component provides requirements for a mechanism that mediates access control based on
 6411 security attributes associated with subjects and objects. Each object and subject has a set of
 6412 associated attributes, such as location, time of creation, access rights such as Access Control
 6413 Lists (ACLs)). This component allows the PP/ST author to specify the attributes that will be
 6414 used for the access control mediation. This component allows access control rules, using these
 6415 attributes, to be specified.

EXAMPLE

Examples of the attributes that a PP/ST author might assign are:

An identity attribute may be associated with users, subjects, or objects to be used for mediation. Examples of such attributes might be the name of the program image used in the creation of the subject, or a security attribute assigned to the program image.

A time attribute can be used to specify that access will be authorized during certain times of the day, during certain days of the week, or during a certain calendar year.

A location attribute **could** specify whether the location is the location of the request for the operation, the location where the operation will be carried out, or both. It **could** be based upon internal tables to translate the logical interfaces of the TSF into locations such as through terminal locations, CPU locations, etc.

A grouping attribute allows a single group of users to be associated with an operation for the purposes of access control. If required, the refinement operation should be used to specify the maximum number of definable groups, the maximum membership of a group, and the maximum number of groups to which a user can concurrently be associated.

6416 This component also provides requirements for the access control security functions to be able
 6417 to explicitly authorize or deny access to an object based upon security attributes. This **could** be
 6418 used to provide privilege, access rights, or access authorizations within the TOE. Such
 6419 privileges, rights, or authorizations **could** apply to users, subjects (representing users or
 6420 applications), and objects.

6421 **F.3.2.2 Operations**6422 **F.3.2.2.1 Assignment**

6423 In FDP_ACF.1.1, the PP/ST author **should** specify an access control SFP name that the TSF is to
 6424 enforce. The name of the access control SFP, and the scope of control for that policy are defined
 6425 in components from Access control policy (FDP_ACC).

6426 In FDP_ACF.1.1, the PP/ST author **should** specify, for each controlled subject and object, the
 6427 security attributes and/or named groups of security attributes that the function will use in the
 6428 specification of the rules. For example, such attributes **may** be things such as the user identity,
 6429 subject identity, role, time of day, location, ACLs, or any other attribute specified by the PP/ST
 6430 author. Named groups of security attributes **can** be specified to provide a convenient means to
 6431 refer to multiple security attributes. Named groups **could** provide a useful way to associate
 6432 “roles” defined in Security management roles (FMT_SMR), and all of their relevant attributes,
 6433 with subjects. In other words, each role **could** relate to a named group of attributes.

6434 In FDP_ACF.1.2, the PP/ST author **should** specify the SFP rules governing access among
 6435 controlled subjects and controlled objects using controlled operations on controlled objects.

6436 These rules specify when access is granted or denied. It **can** specify general access control
 6437 functions or granular access control functions.

EXAMPLE

General access control functions: typical permission bits
 Granular access control: Access Control Lists (ACL)

6438 In FDP_ACF.1.3, the PP/ST author **should** specify the rules, based on security attributes, that
 6439 explicitly authorize access of subjects to objects that will be used to explicitly authorize access.
 6440 These rules are in addition to those specified in FDP_ACF.1.1. They are included in FDP_ACF.1.3
 6441 as they are intended to contain exceptions to the rules in FDP_ACF.1.1. An example of rules to
 6442 explicitly authorize access is based on a privilege vector associated with a subject that always
 6443 grants access to objects covered by the access control SFP that has been specified. If such a
 6444 capability is not desired, then the PP/ST author **should** specify “none”.

6445 In FDP_ACF.1.4, the PP/ST author **should** specify the rules, based on security attributes, that
 6446 explicitly deny access of subjects to objects. These rules are in addition to those specified in
 6447 FDP_ACF.1.1 . They are included in FDP_ACF.1.4 as they are intended to contain exceptions to
 6448 the rules in FDP_ACF.1.1 . An example of rules to explicitly deny access is based on a privilege
 6449 vector associated with a subject that always denies access to objects covered by the access
 6450 control SFP that has been specified. If such a capability is not desired, then the PP/ST author
 6451 **should** specify “none”.

6452 **F.4 Data authentication (FDP_DAU)**

6453 **F.4.1 User notes**

6454 This family describes specific functions that **can** be used to authenticate “static” data.

6455 Components in this family are to be used when there is a requirement for “static” data
 6456 authentication, i.e. where data is to be signed but not transmitted.

6457 Note the Non-repudiation of origin (FCO_NRO) family provides for non-repudiation of origin of information
 6458 received during a data exchange.

6459 **F.4.2 FDP_DAU.1 Basic Data Authentication**

6460 **F.4.2.1 User application notes**

6461 This component **may** be satisfied by one-way hash functions to generate a hash value for a
 6462 definitive document that **may** be used as verification of the validity or authenticity of its
 6463 information content.

EXAMPLE

cryptographic checksum, fingerprint, message digest

6464 **F.4.2.2 Operations**

6465 **F.4.2.2.1 Assignment**

6466 In FDP_DAU.1.1, the PP/ST author **should** specify the list of objects or information types for
 6467 which the TSF **shall** be capable of generating data authentication evidence.

6468 In FDP_DAU.1.2, the PP/ST author **should** specify the list of subjects that will have the ability to
 6469 verify data authentication evidence for the objects identified in the previous element. The list of
 6470 subjects **could** be very specific, if the subjects are known, or it **could** be more generic and refer
 6471 to a “type” of subject such as an identified role.

6472 **F.4.3 FDP_DAU.2 Data Authentication with Identity of Guarantor**

6473 **F.4.3.1 User application notes**

6474 This component additionally requires the ability to verify the identity of the user that provided
6475 the guarantee of authenticity

EXAMPLE

a trusted third party.

6476 **F.4.3.2 Operations**

6477 **F.4.3.2.1 Assignment**

6478 In FDP_DAU.2.1, the PP/ST author **should** specify the list of objects or information types for
6479 which the TSF **shall** be capable of generating data authentication evidence.

6480 In FDP_DAU.2.2, the PP/ST author **should** specify the list of subjects that will have the ability to
6481 verify data authentication evidence for the objects identified in the previous element as well as
6482 the identity of the user that created the data authentication evidence.

6483 **F.5 Export from the TOE (FDP_ETC)**

6484 **F.5.1 User notes**

6485 This family defines functions for TSF-mediated exporting of user data from the TOE such that its
6486 security attributes either **can** be explicitly preserved or **can** be ignored once it has been
6487 exported. Consistency of these security attributes are addressed by Inter-TSF TSF data
6488 consistency (FPT_TDC).

6489 Export from the TOE (FDP_ETC) is concerned with limitations on export and association of
6490 security attributes with the exported user data.

6491 This family, and the corresponding Import family Import from outside of the TOE (FDP_ITC),
6492 address how the TOE deals with user data transferred into and outside its control. In principle,
6493 this family is concerned with the TSF-mediated exporting of user data and its related security
6494 attributes.

6495 A variety of activities might be involved here:

- 6496 a) exporting of user data without any security attributes;
- 6497 b) exporting user data including security attributes where the two are associated with
6498 one another and the security attributes unambiguously represent the exported
6499 user data.

6500 If there are multiple SFPs (access control and/or information flow control) then it **may** be
6501 appropriate to iterate these components once for each named SFP.

6502 **F.5.2 FDP_ETC.1 Export of user data without security attributes**

6503 **F.5.2.1 User application notes**

6504 This component is used to specify the TSF-mediated exporting of user data without the export
6505 of its security attributes.

6506 **F.5.2.2 Operations**

6507 **F.5.2.2.1 Assignment**

6508 In FDP_ETC.1.1, the PP/ST author **should** specify the access control SFP(s) and/or information
6509 flow control SFP(s) that will be enforced when exporting user data. The user data that this
6510 function exports is scoped by the assignment of these SFPs.

6511 **F.5.3 FDP_ETC.2 Export of user data with security attributes**

6512 **F.5.3.1 User application notes**

6513 The user data is exported together with its security attributes. The security attributes are
 6514 unambiguously associated with the user data. There are several ways of achieving this
 6515 association. One way that this **can** be achieved is by physically collocating the user data and the
 6516 security attributes.

EXAMPLE

On the same external media

6517 or by using cryptographic techniques such as secure signatures to associate the attributes and
 6518 the user data. Inter-TSF trusted channel (FTP_ITC) **could** be used to assure that the attributes
 6519 are correctly received at the other trusted IT product while Inter-TSF TSF data consistency
 6520 (FPT_TDC) **can** be used to make sure that those attributes are properly interpreted.
 6521 Furthermore, Trusted path (FTP_TRP) **could** be used to make sure that the export is being
 6522 initiated by the proper user.

6523 F.5.3.2 Operations

6524 F.5.3.2.1 Assignment

6525 In FDP_ETC.2.1, the PP/ST author **should** specify the access control SFP(s) and/or information
 6526 flow control SFP(s) that will be enforced when exporting user data. The user data that this
 6527 function exports is scoped by the assignment of these SFPs.

6528 In FDP_ETC.2.4, the PP/ST author **should** specify any additional exportation control rules or
 6529 “none” if there are no additional exportation control rules. These rules will be enforced by the
 6530 TSF in addition to the access control SFPs and/or information flow control SFPs selected in
 6531 FDP_ETC.2.1.

6532 F.6 Information flow control policy (FDP_IFC)

6533 F.6.1 User notes

6534 This family covers the identification of information flow control SFPs; and, for each, specifies
 6535 the scope of control of the SFP.

6536 The components in this family are capable of identifying the information flow control SFPs to be
 6537 enforced by the traditional Mandatory Access Control mechanisms that would be found in a
 6538 TOE. However, they go beyond just the traditional MAC mechanisms and **can** be used to identify
 6539 and describe non-interference policies and state-transitions. It further defines the subjects
 6540 under control of the policy, the information under control of the policy, and operations which
 6541 cause controlled information to flow to and from controlled subjects for each information flow
 6542 control SFP in the TOE. The information flow control SFP will be defined by other families such
 6543 as Information flow control functions (FDP_IFF) and Export from the TOE (FDP_ETC). The
 6544 information flow control SFPs named here in Information flow control policy (FDP_IFC) are
 6545 meant to be used throughout the remainder of the functional components that have an
 6546 operation that calls for an assignment or selection of an “information flow control SFP.”

6547 These components are quite flexible. They allow the domain of flow control to be specified and
 6548 there is no requirement that the mechanism be based upon labels. The different elements of the
 6549 information flow control components also permit different degrees of exception to the policy.

6550 Each SFP covers a set of triplets: subject, information, and operations that cause information to
 6551 flow to and from subjects. Some information flow control policies **may** be at a very low level of
 6552 detail and explicitly describe subjects in terms of processes within an operating system. Other
 6553 information flow control policies **may** be at a high level and describe subjects in the generic
 6554 sense of users or input/output channels. If the information flow control policy is at too high a
 6555 level of detail, it **may** not clearly define the desired IT security functions. In such cases, it is
 6556 more appropriate to include such descriptions of information flow control policies as objectives.
 6557 Then the desired IT security functions **can** be specified as supportive of those objectives.

6558 In the second component (FDP_IFC.2 Complete information flow control), each information flow
 6559 control SFP will cover all possible operations that cause information covered by that SFP to flow
 6560 to and from subjects covered by that SFP. Furthermore, all information flows will need to be
 6561 covered by a SFP. Therefore, for each action that causes information to flow, there will be a set
 6562 of rules that define whether the action is allowed. If there are multiple SFPs that are applicable
 6563 for a given information flow, all involved SFPs must allow this flow before it is permitted to take
 6564 place.

6565 An information flow control SFP covers a well-defined set of operations. The SFPs coverage **may**
 6566 be “complete” with respect to some information flows, or it **may** address only some of the
 6567 operations that affect the information flow.

6568 An access control SFP controls access to the objects that contain information. An information
 6569 flow control SFP controls access to the information, independent of its container. The attributes
 6570 of the information, which **may** be associated with the attributes of the container (or **may** not, as
 6571 in the case of a multi-level database) stay with the information as it flows. The accessor does
 6572 not have the ability, in the absence of an explicit authorization, to change the attributes of the
 6573 information.

6574 Information flows and operations **can** be expressed at multiple levels. In the case of a ST, the
 6575 information flows and operations might be specified at a system-specific level: TCP/IP packets
 6576 flowing through a firewall based upon known IP addresses. For a PP, the information flows and
 6577 operations might be expressed as types: email, data repositories, observe accesses, etc.

6578 The components in this family **can** be applied multiple times in a PP/ST to different subsets of
 6579 operations and objects. This will accommodate TOEs that contain multiple policies, each
 6580 addressing a particular set of objects, subjects, and operations.

6581 **F.6.2 FDP_IFC.1 Subset information flow control**

6582 **F.6.2.1 User application notes**

6583 This component requires that an information flow control policy apply to a subset of the
 6584 possible operations in the TOE.

6585 **F.6.2.2 Operations**

6586 **F.6.2.2.1 Assignment**

6587 In FDP_IFC.1.1, the PP/ST author **should** specify a uniquely named information flow control SFP
 6588 to be enforced by the TSF.

6589 In FDP_IFC.1.1, the PP/ST author **should** specify the list of subjects, information, and operations
 6590 which cause controlled information to flow to and from controlled subjects covered by the SFP.
 6591 As mentioned above, the list of subjects **could** be at various levels of detail depending on the
 6592 needs of the PP/ST author.

EXAMPLE

It **could** specify users, machines, or processes.

6593 Information **could** refer to data such as email or network protocols, or more specific objects
 6594 similar to those specified under an access control policy. If the information that is specified is
 6595 contained within an object that is subject to an access control policy, then both the access
 6596 control policy and information flow control policy must be enforced before the specified
 6597 information **could** flow to or from the object.

6598 **F.6.3 FDP_IFC.2 Complete information flow control**

6599 **F.6.3.1 User application notes**

6600 This component requires that all possible operations that cause information to flow to and from
 6601 subjects included in the SFP, are covered by an information flow control SFP.

6602 The PP/ST author must demonstrate that each combination of information flows and subjects is
6603 covered by an information flow control SFP.

6604 **F.6.3.2 Operations**

6605 **F.6.3.2.1 Assignment**

6606 In FDP_IFC.2.1, the PP/ST author **should** specify a uniquely named information flow control SFP
6607 to be enforced by the TSF.

6608 In FDP_IFC.2.1, the PP/ST author **should** specify the list of subjects and information that will be
6609 covered by the SFP. All operations that cause that information to flow to and from subjects will
6610 be covered by the SFP. As mentioned above, the list of subjects **could** be at various levels of
6611 detail depending on the needs of the PP/ST author.

EXAMPLE

It **could** specify users, machines, or processes.

6612 Information **could** refer to data such as email or network protocols, or more specific objects
6613 similar to those specified under an access control policy. If the information that is specified is
6614 contained within an object that is subject to an access control policy, then both the access
6615 control policy and information flow control policy must be enforced before the specified
6616 information **could** flow to or from the object.

6617 **F.7 Information flow control functions (FDP_IFF)**

6618 **F.7.1 User notes**

6619 This family describes the rules for the specific functions that **can** implement the information
6620 flow control SFPs named in Information flow control policy (FDP_IFC), which also specifies the
6621 scope of control of the policies. It consists of two “trees:” one addressing the common
6622 information flow control function issues, and a second addressing illicit information flows (i.e.
6623 covert channels) with respect to one or more information flow control SFPs. This division arises
6624 because the issues concerning illicit information flows are, in some sense, orthogonal to the rest
6625 of an SFP. Illicit information flows are flows in violation of policy; thus, they are not a policy
6626 issue.

6627 In order to implement strong protection against disclosure or modification in the face of
6628 untrusted software, controls on information flow are required. Access controls alone are not
6629 sufficient because they only control access to containers, allowing the information they contain
6630 to flow, without controls, throughout a system.

6631 In this family, the phrase “types of illicit information flows” is used. This phrase **may** be used to
6632 refer to the categorization of flows as “Storage Channels” or “Timing Channels”, or it **can** refer to
6633 improved categorizations reflective of the needs of a PP/ST author.

6634 The flexibility of these components allows the definition of a privilege policy within FDP_IFF.1
6635 Simple security attributes and FDP_IFF.2 Hierarchical security attributes to allow the controlled
6636 bypass of all or part of a particular SFP. If there is a need for a predefined approach to SFP
6637 bypass, the PP/ST author **should** consider incorporating a privilege policy.

6638 **F.7.2 FDP_IFF.1 Simple security attributes**

6639 **F.7.2.1 User application notes**

6640 This component requires security attributes on information, and on subjects that cause that
6641 information to flow and subjects that act as recipients of that information. The attributes of the
6642 containers of the information **should** also be considered if it is desired that they **should** play a
6643 part in information flow control decisions or if they are covered by an access control policy.
6644 This component specifies the key rules that are enforced and describes how security attributes
6645 are derived.

6646 This component does not specify the details of how a security attribute is assigned (i.e. user
6647 versus process). Flexibility in policy is provided by having assignments that allow specification
6648 of additional policy and function requirements, as necessary.

6649 This component also provides requirements for the information flow control functions to be
6650 able to explicitly authorize and deny an information flow based upon security attributes. This
6651 **could** be used to implement a privilege policy that covers exceptions to the basic policy defined
6652 in this component.

6653 **F.7.2.2 Operations**

6654 **F.7.2.2.1 Assignment**

6655 In FDP_IFF.1.1, the PP/ST author **should** specify the information flow control SFPs enforced by
6656 the TSF. The name of the information flow control SFP, and the scope of control for that policy
6657 are defined in components from Information flow control policy (FDP_IFC).

6658 In FDP_IFF.1.1, the PP/ST author **should** specify, for each type of controlled subject and
6659 information, the security attributes that are relevant to the specification of the SFP rules.

EXAMPLE

For example, such security attributes **may** be things such the subject identifier, subject sensitivity label, subject clearance label, information sensitivity label, etc.

6660 The types of security attributes **should** be sufficient to support the environmental needs.

6661 In FDP_IFF.1.2, the PP/ST author **should** specify for each operation, the security attribute-based
6662 relationship that must hold between subject and information security attributes that the TSF
6663 will enforce.

6664 In FDP_IFF.1.3, the PP/ST author **should** specify any additional information flow control SFP
6665 rules that the TSF is to enforce. This includes all rules of the SFP that are either not based on the
6666 security attributes of the information and the subject or rules that automatically modify the
6667 security attributes of information or subjects as a result of an access operation. An example for
6668 the first case is a rule of the SFP controlling a threshold value for specific types of information.
6669 This would for example be the case when the information flow SFP contains rules on access to
6670 statistical data where a subject is only allowed to access this type of information up to a specific
6671 number of accesses. An example for the second case would be a rule stating under which
6672 conditions and how the security attributes of a subject or object change as the result of an
6673 access operation. Some information flow policies for example **may** limit the number of access
6674 operations to information with specific security attributes. If there are no additional rules then
6675 the PP/ST author **should** specify “none”.

6676 In FDP_IFF.1.4, the PP/ST author **should** specify the rules, based on security attributes, that
6677 explicitly authorize information flows. These rules are in addition to those specified in the
6678 preceding elements. They are included in FDP_IFF.1.4 as they are intended to contain
6679 exceptions to the rules in the preceding elements.

EXAMPLE

An example of rules to explicitly authorize information flows is based on a privilege vector associated with a subject that always grants the subject the ability to cause an information flow for information that is covered by the SFP that has been specified.

6680 If such a capability is not desired, then the PP/ST author **should** specify “none”.

6681 In FDP_IFF.1.5, the PP/ST author **should** specify the rules, based on security attributes, that
6682 explicitly deny information flows. These rules are in addition to those specified in the preceding
6683 elements. They are included in FDP_IFF.1.5 as they are intended to contain exceptions to the
6684 rules in the preceding elements. An example of rules to explicitly deny information flows is
6685 based on a privilege vector associated with a subject that always denies the subject the ability

6686 to cause an information flow for information that is covered by the SFP that has been specified.
6687 If such a capability is not desired, then the PP/ST author **should** specify “none”.

6688 **F.7.3 FDP_IFF.2 Hierarchical security attributes**

6689 **F.7.3.1 User application notes**

6690 This component requires that the named information flow control SFP uses hierarchical
6691 security attributes that form a lattice.

6692 It is important to note that the hierarchical relationship requirements identified in FDP_IFF.2.4
6693 need only apply to the information flow control security attributes for the information flow
6694 control SFPs that have been identified in FDP_IFF.2.1. This component is not meant to apply to
6695 other SFPs such as access control SFPs.

6696 FDP_IFF.2.6 phrases the requirements for the set of security attributes to form a lattice. A
6697 number of information flow policies defined in the literature and implemented in IT products
6698 are based on a set of security attributes that form a lattice. FDP_IFF.2.6 is specifically included
6699 to address this type of information flow policies.

6700 If it is the case that multiple information flow control SFPs are to be specified, and that each of
6701 these SFPs will have their own security attributes that are not related to one another, then the
6702 PP/ST author **should** iterate this component once for each of those SFPs. Otherwise a conflict
6703 might arise with the sub-items of FDP_IFF.2.4 since the required relationships will not exist.

6704 **F.7.3.2 Operations**

6705 **F.7.3.2.1 Assignment**

6706 In FDP_IFF.2.1, the PP/ST author **should** specify the information flow control SFPs enforced by
6707 the TSF. The name of the information flow control SFP, and the scope of control for that policy
6708 are defined in components from Information flow control policy (FDP_IFC).

6709 In FDP_IFF.2.1, the PP/ST author **should** specify, for each type of controlled subject and
6710 information, the security attributes that are relevant to the specification of the SFP rules. For
6711 example, such security attributes **may** be things such the subject identifier, subject sensitivity
6712 label, subject clearance label, information sensitivity label, etc. The types of security attributes
6713 **should** be sufficient to support the environmental needs.

6714 In FDP_IFF.2.2, the PP/ST author **should** specify for each operation, the security attribute-based
6715 relationship that must hold between subject and information security attributes that the TSF
6716 will enforce. These relationships **should** be based upon the ordering relationships between the
6717 security attributes.

6718 In FDP_IFF.2.3, the PP/ST author **should** specify any additional information flow control SFP
6719 rules that the TSF is to enforce. This includes all rules of the SFP that are either not based on the
6720 security attributes of the information and the subject or rules that automatically modify the
6721 security attributes of information or subjects as a result of an access operation. An example for
6722 the first case is a rule of the SFP controlling a threshold value for specific types of information.

EXAMPLE 1

This would for example be the case when the information flow SFP contains rules on access to statistical data where a subject is only allowed to access this type of information up to a specific number of accesses. An example for the second case would be a rule stating under which conditions and how the security attributes of a subject or object change as the result of an access operation.

6723 Some information flow policies **may** limit the number of access operations to information with
6724 specific security attributes. If there are no additional rules then the PP/ST author **should** specify
6725 “none”.

6726 In FDP_IFF.2.4, the PP/ST author **should** specify the rules, based on security attributes, that
6727 explicitly authorize information flows. These rules are in addition to those specified in the

6728 preceding elements. They are included in FDP_IFF.2.4 as they are intended to contain
6729 exceptions to the rules in the preceding elements.

EXAMPLE 2

An example of rules to explicitly authorize information flows is based on a privilege vector associated with a subject that always grants the subject the ability to cause an information flow for information that is covered by the SFP that has been specified.

6730 If such a capability is not desired, then the PP/ST author **should** specify “none”.

6731 In FDP_IFF.2.5, the PP/ST author **should** specify the rules, based on security attributes, that
6732 explicitly deny information flows. These rules are in addition to those specified in the preceding
6733 elements. They are included in FDP_IFF.2.5 as they are intended to contain exceptions to the
6734 rules in the preceding elements. An example of rules to explicitly deny information flows is
6735 based on a privilege vector associated with a subject that always denies the subject the ability
6736 to cause an information flow for information that is covered by the SFP that has been specified.
6737 If such a capability is not desired, then the PP/ST author **should** specify “none”.

6738 **F.7.4 FDP_IFF.3 Limited illicit information flows**

6739 **F.7.4.1 User application notes**

6740 This component **should** be used when at least one of the SFPs that requires control of illicit
6741 information flows does not require elimination of flows.

6742 For the specified illicit information flows, certain maximum capacities **should** be provided. In
6743 addition, a PP/ST author has the ability to specify whether the illicit information flows must be
6744 audited.

6745 **F.7.4.2 Operations**

6746 **F.7.4.2.1 Assignment**

6747 In FDP_IFF.3.1, the PP/ST author **should** specify the information flow control SFPs enforced by
6748 the TSF. The name of the information flow control SFP, and the scope of control for that policy
6749 are defined in components from Information flow control policy (FDP_IFC).

6750 In FDP_IFF.3.1, the PP/ST author **should** specify the types of illicit information flows that are
6751 subject to a maximum capacity limitation.

6752 In FDP_IFF.3.1, the PP/ST author **should** specify the maximum capacity permitted for any
6753 identified illicit information flows.

6754 **F.7.5 FDP_IFF.4 Partial elimination of illicit information flows**

6755 **F.7.5.1 User application notes**

6756 This component **should** be used when all the SFPs that requires control of illicit information
6757 flows require elimination of some (but not necessarily all) illicit information flows.

6758 **F.7.5.2 Operations**

6759 **F.7.5.2.1 Assignment**

6760 In FDP_IFF.4.1, the PP/ST author **should** specify the information flow control SFPs enforced by
6761 the TSF. The name of the information flow control SFP, and the scope of control for that policy
6762 are defined in components from Information flow control policy (FDP_IFC).

6763 In FDP_IFF.4.1, the PP/ST author **should** specify the types of illicit information flows which are
6764 subject to a maximum capacity limitation.

6765 In FDP_IFF.4.1, the PP/ST author **should** specify the maximum capacity permitted for any
6766 identified illicit information flows.

6767 In FDP_IFF.4.2, the PP/ST author **should** specify the types of illicit information flows to be
6768 eliminated. This list **may** not be empty as this component requires that some illicit information
6769 flows are to be eliminated.

6770 **F.7.6 FDP_IFF.5 No illicit information flows**

6771 **F.7.6.1 User application notes**

6772 This component **should** be used when the SFPs that require control of illicit information flows
6773 require elimination of all illicit information flows. However, the PP/ST author **should** carefully
6774 consider the potential impact that eliminating all illicit information flows might have on the
6775 normal functional operation of the TOE. Many practical applications have shown that there is an
6776 indirect relationship between illicit information flows and normal functionality within a TOE
6777 and eliminating all illicit information flows **may** result in less than desired functionality.

6778 **F.7.6.2 Operations**

6779 **F.7.6.2.1 Assignment**

6780 In FDP_IFF.5.1, the PP/ST author **should** specify the information flow control SFP for which
6781 illicit information flows are to be eliminated. The name of the information flow control SFP, and
6782 the scope of control for that policy are defined in components from Information flow control
6783 policy (FDP_IFC).

6784 **F.7.7 FDP_IFF.6 Illicit information flow monitoring**

6785 **F.7.7.1 User application notes**

6786 This component **should** be used when it is desired that the TSF provide the ability to monitor
6787 the use of illicit information flows that exceed a specified capacity. If it is desired that such flows
6788 be audited, then this component **could** serve as the source of audit events to be used by
6789 components from the Security audit data generation (FAU_GEN) family.

6790 **F.7.7.2 Operations**

6791 **F.7.7.2.1 Assignment**

6792 In FDP_IFF.6.1, the PP/ST author **should** specify the information flow control SFPs enforced by
6793 the TSF. The name of the information flow control SFP, and the scope of control for that policy
6794 are defined in components from Information flow control policy (FDP_IFC).

6795 In FDP_IFF.6.1, the PP/ST author **should** specify the types of illicit information flows that will be
6796 monitored for exceeding a maximum capacity.

6797 In FDP_IFF.6.1, the PP/ST author **should** specify the maximum capacity above which illicit
6798 information flows will be monitored by the TSF.

6799 **F.8 Information retention control (FDP_IRC)**

6800 **F.8.1 User notes**

6801 While a great aspect of the elimination of the objects as required by FDP_IRC refers to the
6802 information stored within the object as a container, it also includes all attributes (also in the
6803 meaning of metadata) that may be associated with the object.

6804 In this aspect, the focus of FDP_IRC differs from other components related to access or
6805 information flow control policies, such as FDP_IFF and FDP_IFC. More important, objects here
6806 are always considered in the context of selected activities that are performed on these objects.
6807 In contrast to residual information protection (FDP_RIP), FDP_IRC excludes objects from any
6808 access or information flow and deletes them, irreversibly and untraceably when they are no
6809 longer needed by a set of activities.

6810 While it may not be completely clear, which objects to consider, it is essential that the list of
 6811 objects is assigned by the PP/ST author at the very latest in order to allow for concrete tests. In
 6812 any case the list of objects shall be derived from a structured analysis.

6813 **F.8.2 FDP_IRC.1 Information retention control**

6814 **F.8.2.1 User application notes**

6815 The Information erasure policy as defined in FDP_IRC.1 serves to protect all information that is
 6816 contained in the assigned objects from being misused, regardless of whether the information is
 6817 primary content or any kind of attribute. The policy covers combinations of objects and
 6818 activities. The policy's coverage **may** be "complete" with respect to all objects related to one or
 6819 more activities, or it **may** address only some of objects related to one or more activities.

6820 The term "promptly" in FDP_IRC.1 specifically refers to the fact that the objects **shall** be
 6821 terminated in a manner so that it can be secured they cannot be accessed before.

6822 **F.8.2.2 Operations**

6823 **F.8.2.2.1 Assignment**

6824 In FDP_IRC.1.1, the PP/ST author **should** specify a uniquely named information erasure policy
 6825 to be enforced by the TSF.

6826 In FDP_IRC.1.1, the PP/ST author **should** specify the list of objects that are required for the
 6827 respective list of activities, e.g. "all message objects".

6828 In FDP_IRC.1.1, the PP/ST author **should** specify the list of activities that the information
 6829 erasure policy is concerned with, e.g. "all activities related to passing a message on, such as
 6830 receiving a message, cryptographic handling of a message, sending a message".

6831 In FDP_IRC.1.2, the PP/ST author **should** specify the list of objects that are required for the
 6832 respective list of activities. This assignment **shall** be identical to the assigned objects in
 6833 FDP_IRC.1.1.

6834 **F.9 Import from outside of the TOE (FDP_ITC)**

6835 **F.9.1 User notes**

6836 This family defines mechanisms for TSF-mediated importing of user data from outside the TOE
 6837 into the TOE such that the user data security attributes **can** be preserved. Consistency of these
 6838 security attributes are addressed by Inter-TSF TSF data consistency (FPT_TDC).

6839 Import from outside of the TOE (FDP_ITC) is concerned with limitations on import, user
 6840 specification of security attributes, and association of security attributes with the user data.

6841 This family, and the corresponding export family Export from the TOE (FDP_ETC), address how
 6842 the TOE deals with user data outside its control. This family is concerned with assigning and
 6843 abstraction of the user data security attributes.

EXAMPLE

A variety of activities might be involved here:

- a) importing user data from an unformatted medium (such as, tape, scanner, video or audio signal), without including any security attributes, and physically marking the medium to indicate its contents;
- b) importing user data, including security attributes, from a medium and verifying that the object security attributes are appropriate;
- c) importing user data, including security attributes, from a medium using a cryptographic sealing technique to protect the association of user data and security attributes.

6844 This family is not concerned with the determination of whether the user data **may** be imported.
 6845 It is concerned with the values of the security attributes to associate with the imported user
 6846 data.

6847 There are two possibilities for the import of user data: either the user data is unambiguously
 6848 associated with reliable object security attributes (values and meaning of the security attributes
 6849 is not modified), or no reliable security attributes (or no security attributes at all) are available
 6850 from the import source. This family addresses both cases.

6851 If there are reliable security attributes available, they **may** have been associated with the user
 6852 data by physical means (the security attributes are on the same media), or by logical means (the
 6853 security attributes are distributed differently but include unique object identification).

EXAMPLE

cryptographic checksum

6854 This family is concerned with TSF-mediated importing of user data and maintaining the
 6855 association of security attributes as required by the SFP. Other families are concerned with
 6856 other import aspects such as consistency, trusted channels, and integrity that are beyond the
 6857 scope of this family. Furthermore, Import from outside of the TOE (FDP_ITC) is only concerned
 6858 with the interface to the import medium. Export from the TOE (FDP_ETC) is responsible for the
 6859 other end point of the medium (the source).

6860 Some of the well-known import requirements are:

- 6861 a) importing of user data without any security attributes;
- 6862 b) importing of user data including security attributes where the two are associated
 6863 with one another and the security attributes unambiguously represent the
 6864 information being imported.

6865 These import requirements **may** be handled by the TSF with or without human intervention,
 6866 depending on the IT limitations and the organizational security policy. For example, if user data
 6867 is received on a “confidential” channel, the security attributes of the objects will be set to
 6868 “confidential”.

6869 If there are multiple SFPs (access control and/or information flow control) then it **may** be
 6870 appropriate to iterate these components once for each named SFP.

6871 **F.9.2 FDP_ITC.1 Import of user data without security attributes**

6872 **F.9.2.1 User application notes**

6873 This component is used to specify the import of user data that does not have reliable (or any)
 6874 security attributes associated with it. This function requires that the security attributes for the
 6875 imported user data be initialized within the TSF. It **could** also be the case that the PP/ST author
 6876 specifies the rules for import. It **may** be appropriate, in some environments, to require that
 6877 these attributes be supplied via a trusted path or a trusted channel mechanism.

6878 **F.9.2.2 Operations**

6879 **F.9.2.2.1 Assignment**

6880 In FDP_ITC.1.1, the PP/ST author **should** specify the access control SFP(s) and/or information
 6881 flow control SFP(s) that will be enforced when importing user data from outside of the TOE.
 6882 The user data that this function imports is scoped by the assignment of these SFPs.

6883 In FDP_ITC.1.3, the PP/ST author **should** specify any additional importation control rules or
 6884 “none” if there are no additional importation control rules. These rules will be enforced by the
 6885 TSF in addition to the access control SFPs and/or information flow control SFPs selected in
 6886 FDP_ITC.1.1.

6887 **F.9.3 FDP_ITC.2 Import of user data with security attributes**

6888 **F.9.3.1 User application notes**

6889 This component is used to specify the import of user data that has reliable security attributes
6890 associated with it. This function relies upon the security attributes that are accurately and
6891 unambiguously associated with the objects on the import medium. Once imported, those
6892 objects will have those same attributes. This requires Inter-TSF TSF data consistency
6893 (FPT_TDC) to ensure the consistency of the data. It **could** also be the case that the PP/ST author
6894 specifies the rules for import.

6895 **F.9.3.2 Operations**

6896 **F.9.3.2.1 Assignment**

6897 In FDP_ITC.2.1, the PP/ST author **should** specify the access control SFP(s) and/or information
6898 flow control SFP(s) that will be enforced when importing user data from outside of the TOE.
6899 The user data that this function imports is scoped by the assignment of these SFPs.

6900 In FDP_ITC.2.5, the PP/ST author **should** specify any additional importation control rules or
6901 “none” if there are no additional importation control rules. These rules will be enforced by the
6902 TSF in addition to the access control SFPs and/or information flow control SFPs selected in
6903 FDP_ITC.2.1.

6904 **F.10 Internal TOE transfer (FDP_ITT)**

6905 **F.10.1 User notes**

6906 This family provides requirements that address protection of user data when it is transferred
6907 between parts of a TOE across an internal channel. This **may** be contrasted with the Inter-TSF
6908 user data confidentiality transfer protection (FDP_UCT) and Inter-TSF user data integrity
6909 transfer protection (FDP_UIT) family, which provide protection for user data when it is
6910 transferred between distinct TSFs across an external channel, and Export from the TOE
6911 (FDP_ETC) and Import from outside of the TOE (FDP_ITC), which address TSF-mediated
6912 transfer of data to or from outside the TOE.

6913 The requirements in this family allow a PP/ST author to specify the desired security for user
6914 data while in transit within the TOE. This security **could** be protection against disclosure,
6915 modification, or loss of availability.

6916 The determination of the degree of physical separation above which this family **should** apply
6917 depends on the intended environment of use. In a hostile environment, there **may** be risks
6918 arising from transfers between parts of the TOE separated by only a system bus. In more benign
6919 environments, the transfers **may** be across more traditional network media.

6920 If there are multiple SFPs (access control and/or information flow control) then it **may** be
6921 appropriate to iterate these components once for each named SFP.

6922 **F.10.2 FDP_ITT.1 Basic internal transfer protection**

6923 **F.10.2.1 Operations**

6924 **F.10.2.1.1 Assignment**

6925 In FDP_ITT.1.1, the PP/ST author **should** specify the access control SFP(s) and/or information
6926 flow control SFP(s) covering the information being transferred.

6927 **F.10.2.1.2 Selection**

6928 In FDP_ITT.1.1, the PP/ST author **should** specify the types of transmission errors that the TSF
6929 **should** prevent occurring for user data while in transport. The options are disclosure,
6930 modification, loss of use.

6931 **F.10.3 FDP_ITT.2 Transmission separation by attribute**

6932 **F.10.3.1 User application notes**

6933 This component **could**, for example, be used to provide different forms of protection to
6934 information with different clearance levels.

6935 One of the ways to achieve separation of data when it is transmitted is through the use of
6936 separate logical or physical channels.

6937 **F.10.3.2 Operations**

6938 **F.10.3.2.1 Assignment**

6939 In FDP_ITT.2.1, the PP/ST author **should** specify the access control SFP(s) and/or information
6940 flow control SFP(s) covering the information being transferred.

6941 **F.10.3.2.2 Selection**

6942 In FDP_ITT.2.1, the PP/ST author **should** specify the types of transmission errors that the TSF
6943 **should** prevent occurring for user data while in transport. The options are disclosure,
6944 modification, loss of use.

6945 **F.10.3.2.3 Assignment**

6946 In FDP_ITT.2.2, the PP/ST author **should** specify the security attributes, the values of which the
6947 TSF will use to determine when to separate data that is being transmitted between physically-
6948 separated parts of the TOE. An example is that user data associated with the identity of one
6949 owner is transmitted separately from the user data associated with the identify of a different
6950 owner. In this case, the value of the identity of the owner of the data is what is used to
6951 determine when to separate the data for transmission.

6952 **F.10.4 FDP_ITT.3 Integrity monitoring**

6953 **F.10.4.1 User application notes**

6954 This component is used in combination with either FDP_ITT.1 Basic internal transfer protection
6955 or FDP_ITT.2 Transmission separation by attribute. It ensures that the TSF checks received user
6956 data (and their attributes) for integrity. FDP_ITT.1 Basic internal transfer protection or
6957 FDP_ITT.2 Transmission separation by attribute will provide the data in a manner such that it is
6958 protected from modification (so that FDP_ITT.3 Integrity monitoring **can** detect any
6959 modifications).

6960 The PP/ST author has to specify the types of errors that must be detected. The PP/ST author
6961 **should** consider: modification of data, substitution of data, unrecoverable ordering change of
6962 data, replay of data, incomplete data, in addition to other integrity errors.

6963 The PP/ST author must specify the actions that the TSF **should** take on detection of a failure.

EXAMPLE

For example: ignore the user data, request the data again, inform the authorized administrator, reroute traffic for other lines.

6964 **F.10.4.2 Operations**

6965 **F.10.4.2.1 Assignment**

6966 In FDP_ITT.3.1, the PP/ST author **should** specify the access control SFP(s) and/or information
6967 flow control SFP(s) covering the information being transferred and monitored for integrity
6968 errors.

6969 In FDP_ITT.3.1, the PP/ST author **should** specify the type of possible integrity errors to be
6970 monitored during transmission of the user data.

6971 In FDP_ITT.3.2, the PP/ST author **should** specify the action to be taken by the TSF when an
6972 integrity error is encountered.

EXAMPLE

An example is that the TSF should request the resubmission of the user data. The SFP(s) specified in FDP_ITT.3.1 will be enforced as the actions are taken by the TSF.

6973 **F.10.5 FDP_ITT.4 Attribute-based integrity monitoring**6974 **F.10.5.1 User application notes**

6975 This component is used in combination with FDP_ITT.2 Transmission separation by attribute. It
6976 ensures that the TSF checks received user data, that has been transmitted by separate channels
6977 (based on values of specified security attributes), for integrity. It allows the PP/ST author to
6978 specify actions to be taken upon detection of an integrity error.

EXAMPLE

This component **could** be used to provide different integrity error detection and action for information at different integrity levels.

6979 The PP/ST author has to specify the types of errors that must be detected. The PP/ST author
6980 **should** consider: modification of data, substitution of data, unrecoverable ordering change of
6981 data, replay of data, incomplete data, in addition to other integrity errors.

6982 The PP/ST author **should** specify the attributes (and associated transmission channels) that
6983 necessitate integrity error monitoring.

6984 The PP/ST author must specify the actions that the TSF **should** take on detection of a failure.

EXAMPLE

For example: ignore the user data, request the data again, inform the authorized administrator, reroute traffic for other lines.

6985 **F.10.5.2 Operations**6986 **F.10.5.2.1 Assignment**

6987 In FDP_ITT.4.1, the PP/ST author **should** specify the access control SFP(s) and/or information
6988 flow control SFP(s) covering the information being transferred and monitored for integrity
6989 errors.

6990 In FDP_ITT.4.1, the PP/ST author **should** specify the type of possible integrity errors to be
6991 monitored during transmission of the user data.

6992 In FDP_ITT.4.1, the PP/ST author **should** specify a list of security attributes that require
6993 separate transmission channels. This list is used to determine which user data to monitor for
6994 integrity errors, based on its security attributes and its transmission channel. This element is
6995 directly related to FDP_ITT.2 Transmission separation by attribute.

6996 In FDP_ITT.4.2, the PP/ST author **should** specify the action to be taken by the TSF when an
6997 integrity error is encountered. An example might be that the TSF **should** request the
6998 resubmission of the user data. The SFP(s) specified in FDP_ITT.4.1 will be enforced as the
6999 actions are taken by the TSF.

7000 **F.11 Residual information protection (FDP_RIP)**7001 **F.11.1 User notes**

7002 Residual information protection ensures that TSF-controlled resources when de-allocated from
7003 an object and before they are reallocated to another object are treated by the TSF in a way that
7004 it is not possible to reconstruct all or part of the data contained in the resource before it was de-
7005 allocated.

7006 A TOE usually has a number of functions that potentially de-allocate resources from an object
7007 and potentially re-allocate those resources to objects. Some, but not all of those resources **may**

7008 have been used to store critical data from the previous use of the resource and for those
 7009 resources FDP_RIP requires that they are prepared for reuse. Object reuse applies to explicit
 7010 requests of a subject or user to release resources as well as implicit actions of the TSF that
 7011 result in the de-allocation and subsequent re-allocation of resources to different objects.

EXAMPLE

Examples of explicit requests are the deletion or truncation of a file or the release of an area of main memory. Examples of implicit actions of the TSF are the de-allocation and re-allocation of cache regions.

7012 The requirement for object reuse is related to the content of the resource belonging to an
 7013 object, not all information about the resource or object that **may** be stored elsewhere in the TSF.
 7014 As an example, to satisfy the FDP_RIP requirement for files as objects requires that all sectors
 7015 that make up the file need to be prepared for re-use.

7016 It also applies to resources that are serially reused by different subjects within the system. For
 7017 example, most operating systems typically rely upon hardware registers (resources) to support
 7018 processes within the system. As processes are swapped from a “run” state to a “sleep” state
 7019 (and vice versa), these registers are serially reused by different subjects. While this “swapping”
 7020 action **may** not be considered an allocation or deallocation of a resource, Residual information
 7021 protection (FDP_RIP) **could** apply to such events and resources.

7022 Residual information protection (FDP_RIP) typically controls access to information that is not
 7023 part of any currently defined or accessible object; however, in certain cases this **may** not be
 7024 true. For example, object “A” is a file and object “B” is the disk upon which that file resides. If
 7025 object “A” is deleted, the information from object “A” is under the control of Residual
 7026 information protection (FDP_RIP) even though it is still part of object “B”.

7027 It is important to note that Residual information protection (FDP_RIP) applies only to on-line
 7028 objects and not off-line objects such as those backed-up on tapes. For example, if a file is deleted
 7029 in the TOE, Residual information protection (FDP_RIP) **can** be instantiated to require that no
 7030 residual information exists upon deallocation; however, the TSF cannot extend this
 7031 enforcement to that same file that exists on the off-line back-up. Therefore, that same file is still
 7032 available. If this is a concern, then the PP/ST author **should** make sure that the proper
 7033 environmental objectives are in place to support operational user guidance to address off-line
 7034 objects.

7035 Residual information protection (FDP_RIP) and Rollback (FDP_ROL) **can** conflict when Residual
 7036 information protection (FDP_RIP) is instantiated to require that residual information be cleared
 7037 at the time the application releases the object to the TSF (i.e. upon deallocation). Therefore, the
 7038 Residual information protection (FDP_RIP) selection of “deallocation” **should** not be used with
 7039 Rollback (FDP_ROL) since there would be no information to roll back. The other selection,
 7040 “unavailability upon allocation”, **may** be used with Rollback (FDP_ROL), but there is the risk that
 7041 the resource which held the information has been allocated to a new object before the roll back
 7042 took place. If that were to occur, then the roll back would not be possible.

7043 There are no audit requirements in Residual information protection (FDP_RIP) because this is
 7044 not a user-invokable function. Auditing of allocated or deallocated resources would be auditable
 7045 as part of the access control SFP or the information flow control SFP operations.

7046 This family **should** apply to the objects specified in the access control SFP(s) or the information
 7047 flow control SFP(s) as specified by the PP/ST author.

7048 **F.11.2 FDP_RIP.1 Subset residual information protection**

7049 **F.11.2.1 User application notes**

7050 This component requires that, for a subset of the objects in the TOE, the TSF will ensure that
 7051 there is no available residual information contained in a resource allocated to those objects or
 7052 deallocated from those objects.

7053 **F.11.2.2 Operations**7054 **F.11.2.2.1 Selection**

7055 In FDP_RIP.1.1, the PP/ST author **should** specify the event, allocation of the resource to or
7056 deallocation of the resource from, that invokes the residual information protection function.

7057 **F.11.2.2.2 Assignment**

7058 In FDP_RIP.1.1, the PP/ST author **should** specify the list of objects subject to residual
7059 information protection.

7060 **F.11.3 FDP_RIP.2 Full residual information protection**7061 **F.11.3.1 User application notes**

7062 This component requires that for all objects in the TOE, the TSF will ensure that there is no
7063 available residual information contained in a resource allocated to those objects or deallocated
7064 from those objects.

7065 **F.11.3.2 Operations**7066 **F.11.3.2.1 Selection**

7067 In FDP_RIP.2.1, the PP/ST author **should** specify the event, allocation of the resource to or
7068 deallocation of the resource from, that invokes the residual information protection function.

7069 **F.12 Rollback (FDP_ROL)**7070 **F.12.1 User notes**

7071 This family addresses the need to return to a well-defined valid state, such as the need of a user
7072 to undo modifications to a file or to undo transactions in case of an incomplete series of
7073 transaction as in the case of databases.

7074 This family is intended to assist a user in returning to a well-defined valid state after the user
7075 undoes the last set of actions, or, in distributed databases, the return of all of the distributed
7076 copies of the databases to the state before an operation failed.

7077 Residual information protection (FDP_RIP) and Rollback (FDP_ROL) conflict when Residual
7078 information protection (FDP_RIP) enforces that the contents will be made unavailable at the
7079 time that a resource is deallocated from an object. Therefore, this use of Residual information
7080 protection (FDP_RIP) cannot be combined with Rollback (FDP_ROL) as there would be no
7081 information to roll back. Residual information protection (FDP_RIP) **can** be used only with
7082 Rollback (FDP_ROL) when it enforces that the contents will be unavailable at the time that a
7083 resource is allocated to an object. This is because the Rollback (FDP_ROL) mechanism will have
7084 an opportunity to access the previous information that **may** still be present in the TOE in order
7085 to successfully roll back the operation.

7086 The rollback requirement is bounded by certain limits.

EXAMPLE

For example, a text editor typically only allows you roll back up to a certain number of commands. Another example would be backups. If backup tapes are rotated, after a tape is reused, the information **can** no longer be retrieved. This also poses a bound on the rollback requirement.

7087 **F.12.2 FDP_ROL.1 Basic rollback**7088 **F.12.2.1 User application notes**

7089 This component allows a user or subject to undo a set of operations on a predefined set of
7090 objects. The undo is only possible within certain limits, for example up to a number of
7091 characters or up to a time limit.

7092 **F.12.2.2 Operations**

7093 **F.12.2.2.1 Assignment**

7094 In FDP_ROL.1.1, the PP/ST author **should** specify the access control SFP(s) and/or information
7095 flow control SFP(s) that will be enforced when performing rollback operations. This is
7096 necessary to make sure that roll back is not used to circumvent the specified SFPs.

7097 In FDP_ROL.1.1, the PP/ST author **should** specify the list of operations that **can** be rolled back.

7098 In FDP_ROL.1.1, the PP/ST author **should** specify the information and/or list of objects that are
7099 subjected to the rollback policy.

7100 In FDP_ROL.1.2, the PP/ST author **should** specify the boundary limit to which rollback
7101 operations **may** be performed. The boundary **may** be specified as a predefined period of time,

EXAMPLE

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

7102 **F.12.3 FDP_ROL.2 Advanced rollback**

7103 **F.12.3.1 User application notes**

7104 This component enforces that the TSF provide the capability to rollback all operations;
7105 however, the user **can** choose to rollback only a part of them.

7106 **F.12.3.2 Operations**

7107 **F.12.3.2.1 Assignment**

7108 In FDP_ROL.2.1, the PP/ST author **should** specify the access control SFP(s) and/or information
7109 flow control SFP(s) that will be enforced when performing rollback operations. This is
7110 necessary to make sure that roll back is not used to circumvent the specified SFPs.

7111 In FDP_ROL.2.1, the PP/ST author **should** specify the list of objects that are subjected to the
7112 rollback policy.

7113 In FDP_ROL.2.2, the PP/ST author **should** specify the boundary limit to which rollback
7114 operations **may** be performed. The boundary **may** be specified as a predefined period of time,

EXAMPLE

for example, operations **may** be undone which were performed within the past two minutes.

7115 Other possible boundaries **may** be defined as the maximum number of operations allowable or
7116 the size of a buffer.

7117 **F.13 Stored data confidentiality (FDP_SDC)**

7118 **Editors' Note:**

7119 Since no contribution was received in regard to the application notes for FDP_SDC the editors' have
7120 proposed the following text. Subject matter experts are requested to review this carefully.

7121 If no comments are received during the commenting period for this draft, the editors' proposal will be
7122 accepted in the next draft.

7123 **F.13.1 User notes**

7124 This family provides requirements that address protection of user data confidentiality while the
7125 data is stored within memory areas protected by the TSF. The TSF provides access to the data in
7126 the memory through the specified interfaces only and prevents compromise of their

7127 information bypassing these interfaces. It complements the family Stored data integrity
7128 (FDP_SDI) which protects the user data from integrity errors while being stored in the memory

7129 **F.13.2 FDP_SDC.1 Stored data confidentiality**

7130 **F.13.2.1 User application notes**

7131 In FDP_SDC.1 Stored data confidentiality, the PP/ST author specifies which user data is to be
7132 protected and in which type of memory the user data is requested to be protected. In the second
7133 selection the PP/ST author provides the memory type where the user data is to be protected.

7134 **F.13.2.2 Operations**

7135 **F.13.2.2.1 Selection**

7136 In FDP_SDC.1.1 the PP/ST author shall select either “all user data” or provide a list of user data
7137 using the assignment below. In the second selection, the PP/ST author can specify either
7138 temporary memory, persistent memory or any memory. “Any memory” includes both
7139 temporary (volatile) and persistent (non-volatile) memory.

7140 **F.13.2.2.2 Assignment**

7141 In FDP_SDC.1.1 the PP/ST author provides a list of the user data that is to be protected in
7142 memory.

7143 **F.13.3 FDP_SDC.2 Stored data confidentiality with dedicated method**

7144 **F.13.3.1 User application notes**

7145 FDP_SDC.2 Stored data confidentiality with dedicated method refines the FDP_SDC.1.1 element
7146 by allowing the PP/ST author to refine the list of user data using a variety of data
7147 characteristics.

7148 **F.13.3.2 Evaluator application notes**

7149 In practice, the dependency to FCS_COP.1 may be satisfied by a PP/ST author by providing a
7150 rationale explaining an alternative method to cryptography is used in some dedicated cases.

7151 **F.13.3.3 Operations**

7152 **F.13.3.3.1 Assignment**

7153 The first assignment is the same

7154 For the second assignment the PP/ST author provides the data characteristics. Data
7155 characteristics can include items such as data length (shorter or longer than a threshold), data
7156 type (binary, text, image, sound, video), and data representation (binary, vector, character,
7157 frame).

7158 **F.14 Stored data integrity (FDP_SDI)**

7159 **F.14.1 User notes**

7160 This family provides requirements that address protection of user data while it is stored within
7161 containers controlled by the TSF.

7162 Hardware glitches or errors may affect data stored in memory. This family provides
7163 requirements to detect these unintentional errors. The integrity of user data while stored on
7164 storage devices controlled by the TSF are also addressed by this family.

7165 To prevent a subject from modifying the data, the Information flow control functions (FDP_IFF)
7166 or Access control functions (FDP_ACF) families are required (rather than this family).

7167 This family differs from Internal TOE transfer (FDP_ITT) that protects the user data from
7168 integrity errors while being transferred within the TOE.

7169 **F.14.2 FDP_SDI.1 Stored data integrity monitoring**

7170 **F.14.2.1 User application notes**

7171 This component monitors data stored on media for integrity errors. The PP/ST author **can**
7172 specify different kinds of user data attributes that will be used as the basis for monitoring.

7173 **F.14.2.2 Operations**

7174 **F.14.2.2.1 Assignment**

7175 In FDP_SDI.1.1, the PP/ST author **should** specify the integrity errors that the TSF will detect.

7176 In FDP_SDI.1.1, the PP/ST author **should** specify the user data attributes that will be used as the
7177 basis for the monitoring.

7178 **F.14.3 FDP_SDI.2 Stored data integrity monitoring and action**

7179 **F.14.3.1 User application notes**

7180 This component monitors data stored on media for integrity errors. The PP/ST author **can**
7181 specify which action **should** be taken in case an integrity error is detected.

7182 **F.14.3.2 Operations**

7183 **F.14.3.2.1 Assignment**

7184 In FDP_SDI.2.1, the PP/ST author **should** specify the integrity errors that the TSF will detect.

7185 In FDP_SDI.2.1, the PP/ST author **should** specify the user data attributes that will be used as the
7186 basis for the monitoring.

7187 In FDP_SDI.2.2, the PP/ST author **should** specify the actions to be taken in case an integrity
7188 error is detected.

7189 **F.15 Inter-TSF user data confidentiality transfer protection (FDP_UCT)**

7190 **F.15.1 User notes**

7191 This family defines the requirements for ensuring the confidentiality of user data when it is
7192 transferred using an external channel between the TOE and another trusted IT product.
7193 Confidentiality is enforced by preventing unauthorized disclosure of user data in transit
7194 between the two end points. The end points **may** be a TSF or a user.

7195 This family provides a requirement for the protection of user data during transit. In contrast,
7196 Confidentiality of exported TSF data (FPT_ITC) handles TSF data.

7197 **F.15.2 FDP_UCT.1 Basic data exchange confidentiality**

7198 **F.15.2.1 User application notes**

7199 Depending on the access control or information flow policies the TSF is required to send or
7200 receive user data in a manner such that the confidentiality of the user data is protected.

7201 **F.15.2.2 Operations**

7202 **F.15.2.2.1 Assignment**

7203 In FDP_UCT.1.1, the PP/ST author **should** specify the access control SFP(s) and/or information
7204 flow control SFP(s) that will be enforced when exchanging user data. The specified policies will
7205 be enforced to make decisions about who **can** exchange data and which data **can** be exchanged.

7206 **F.15.2.2.2 Selection**

7207 In FDP_UCT.1.1, the PP/ST author **should** specify whether this element applies to a mechanism
7208 that transmits or receives user data.

7209 **F.16 Inter-TSF user data integrity transfer protection (FDP_UIT)**7210 **F.16.1 User notes**

7211 This family defines the requirements for providing integrity for user data in transit between the
7212 TSF and another trusted IT product and recovering from detectable errors. At a minimum, this
7213 family monitors the integrity of user data for modifications. Furthermore, this family supports
7214 different ways of correcting detected integrity errors.

7215 This family defines the requirements for providing integrity for user data in transit; while
7216 Integrity of exported TSF data (FPT_ITI) handles TSF data.

7217 Inter-TSF user data integrity transfer protection (FDP_UIT) and Inter-TSF user data
7218 confidentiality transfer protection (FDP_UCT) are duals of each other, as Inter-TSF user data
7219 confidentiality transfer protection (FDP_UCT) addresses user data confidentiality. Therefore,
7220 the same mechanism that implements Inter-TSF user data integrity transfer protection
7221 (FDP_UIT) **could** possibly be used to implement other families such as Inter-TSF user data
7222 confidentiality transfer protection (FDP_UCT) and Import from outside of the TOE (FDP_ITC).

7223 **F.16.2 FDP_UIT.1 Data exchange integrity**7224 **F.16.2.1 User application notes**

7225 Depending on the access control or information flow policies the TSF is required to send or
7226 receive user data in a manner such that modification of the user data is detected. There is no
7227 requirement for a TSF mechanism to attempt to recover from the modification.

7228 **F.16.2.2 Operations**7229 **F.16.2.2.1 Assignment**

7230 In FDP_UIT.1.1, the PP/ST author **should** specify the access control SFP(s) and/or information
7231 flow control SFP(s) that will be enforced on the transmitted data or on the received data. The
7232 specified policies will be enforced to make decisions about who **can** transmit or who **can** receive
7233 data, and which data **can** be transmitted or received.

7234 **F.16.2.2.2 Selection**

7235 In FDP_UIT.1.1, the PP/ST author **should** specify whether this element applies to a TSF that is
7236 transmitting or receiving objects.

7237 In FDP_UIT.1.1, the PP/ST author **should** specify whether the data **should** be protected from
7238 modification, deletion, insertion, or replay.

7239 In FDP_UIT.1.2, the PP/ST author **should** specify whether the errors of the type: modification,
7240 deletion, insertion, or replay are detected.

7241 **F.16.3 FDP_UIT.2 Source data exchange recovery**7242 **F.16.3.1 User application notes**

7243 This component provides the ability to recover from a set of identified transmission errors, if
7244 required, with the help of the other trusted IT product. As the other trusted IT product is
7245 outside the TOE, the TSF cannot control its behaviour. However, it **can** provide functions that
7246 have the ability to cooperate with the other trusted IT product for the purposes of recovery.

EXAMPLE

For example, the TSF **could** include functions that depend upon the source trusted IT product to re-send the data in the event that an error is detected.

7247 This component deals with the ability of the TSF to handle such an error recovery.

7248 **F.16.3.2 Operations**7249 **F.16.3.2.1 Assignment**

7250 In FDP_UIT.2.1, the PP/ST author **should** specify the access control SFP(s) and/or information
7251 flow control SFP(s) that will be enforced when recovering user data. The specified policies will
7252 be enforced to make decisions about which data **can** be recovered and how it **can** be recovered.

7253 In FDP_UIT.2.1, the PP/ST author **should** specify the list of integrity errors from which the TSF,
7254 with the help of the source trusted IT product, is be able to recover the original user data.

7255 **F.16.4 FDP_UIT.3 Destination data exchange recovery**

7256 **F.16.4.1 User application notes**

7257 This component provides the ability to recover from a set of identified transmission errors. It
7258 accomplishes this task without help from the source trusted IT product. For example, if certain
7259 errors are detected, the transmission protocol must be robust enough to allow the TSF to
7260 recover from the error based on checksums and other information available within that
7261 protocol.

7262 **F.16.4.2 Operations**

7263 **F.16.4.2.1 Assignment**

7264 In FDP_UIT.3.1, the PP/ST author **should** specify the access control SFP(s) and/or information
7265 flow control SFP(s) that will be enforced when recovering user data. The specified policies will
7266 be enforced to make decisions about which data **can** be recovered and how it **can** be recovered.

7267 In FDP_UIT.3.1, the PP/ST author **should** specify the list of integrity errors from which the
7268 receiving TSF, alone, is able to recover the original user data.

7269
7270
7271
7272

Annex G (normative)

Class FIA: Identification and authentication- application notes

7273 G.1 General information

7274 A common security requirement is to unambiguously identify the person and/or entity
7275 performing functions in a TOE. This involves not only establishing the claimed identity of each
7276 user, but also verifying that each user is indeed who he/she claims to be. This is achieved by
7277 requiring users to provide the TSF with some information that is known by the TSF to be
7278 associated with the user in question.

7279 Families in this class address the requirements for functions to establish and verify a claimed
7280 user identity. Identification and Authentication is required to ensure that users are associated
7281 with the proper security attributes

EXAMPLE

Security attributes include identity, groups, roles, security, or integrity levels.

7282 The unambiguous identification of authorized users and the correct association of security
7283 attributes with users and subjects is critical to the enforcement of the security policies.

7284 The Authentication failures (FIA_AFL) family addresses defining limits on repeated
7285 unsuccessful authentication attempts.

7286 The Authentication proof of identity (FIA_API) family addresses defining the functionality
7287 provided by the TOE to prove its identity and to be verified by an external entity in the TOE IT
7288 environment.

7289 The User attribute definition (FIA_ATD) family addresses the definition of user attributes that
7290 are used in the enforcement of the SFRs.

7291 The Specification of secrets (FIA_SOS) family addresses the generation and verification of
7292 secrets that satisfy a defined metric.

7293 The User authentication (FIA_UAU) family addresses verifying the identity of a user.

7294 The User identification (FIA_UID) family addresses determining the identity of a user.

7295 The User-subject binding (FIA_USB) family addresses the correct association of security
7296 attributes for each authorized user.

7297 G.2 Authentication failures (FIA_AFL)

7298 G.2.1 User notes

7299 This family addresses requirements for defining values for authentication attempts and TSF
7300 actions in cases of authentication attempt failure. Parameters include, but are not limited to, the
7301 number of attempts and time thresholds.

7302 The session establishment process is the interaction with the user to perform the session
7303 establishment independent of the actual implementation. If the number of unsuccessful
7304 authentication attempts exceeds the indicated threshold, either the user account or the terminal
7305 (or both) will be locked. If the user account is disabled, the user cannot log-on to the system. If
7306 the terminal is disabled, the terminal (or the address that the terminal has) cannot be used for
7307 any log-on. Both of these situations continue until the condition for re-establishment is
7308 satisfied.

7309 G.2.2 FIA_AFL.1 Authentication failure handling

7310 **G.2.2.1 User application notes**

7311 The PP/ST author **may** define the number of unsuccessful authentication attempts or **may**
 7312 choose to let the TOE developer or the authorized user to define this number. The unsuccessful
 7313 authentication attempts need not be consecutive, but rather related to an authentication event.
 7314 Such an authentication event **could** be the count from the last successful session establishment
 7315 at a given terminal.

7316 The PP/ST author **could** specify a list of actions that the TSF **shall** take in the case of
 7317 authentication failure. An authorized administrator **could** also be allowed to manage the events,
 7318 if deemed opportune by the PP/ST author. These actions **could** be, among other things, terminal
 7319 deactivation, user account deactivation, or administrator alarm. The conditions under which the
 7320 situation will be restored to normal must be specified on the action.

7321 In order to prevent denial of service, TOEs usually ensure that there is at least one user account
 7322 that cannot be disabled.

7323 Further actions for the TSF **can** be stated by the PP/ST author, including rules for re-enabling
 7324 the user session establishment process, or sending an alarm to the administrator.

EXAMPLE

Examples of these actions are: until a specified time has lapsed, until the authorized administrator re-enables the terminal/account, a time related to failed previous attempts (every time the attempt fails, the disabling time is doubled).

7325 **G.2.2.2 Operations**7326 **G.2.2.2.1 Selection**

7327 In FIA_AFL.1 Authentication failure handling, the PP/ST author **should** select either the
 7328 assignment of a positive integer, or the phrase “an administrator configurable positive integer”
 7329 specifying the range of acceptable values.

7330 **G.2.2.2.2 Assignment**

7331 In FIA_AFL.1 Authentication failure handling, the PP/ST author **should** specify the
 7332 authentication events. Examples of these authentication events are: the unsuccessful
 7333 authentication attempts since the last successful authentication for the indicated user identity,
 7334 the unsuccessful authentication attempts since the last successful authentication for the current
 7335 terminal, the number of unsuccessful authentication attempts in the last 10 minutes. At least
 7336 one authentication event must be specified.

7337 In FIA_AFL.1 Authentication failure handling, if the assignment of a positive integer is selected,
 7338 the PP/ST author **should** specify the default number (positive integer) of unsuccessful
 7339 authentication attempts that, when met or surpassed, will trigger the events.

7340 In FIA_AFL.1 Authentication failure handling, if an administrator configurable positive integer is
 7341 selected, the PP/ST author **should** specify the range of acceptable values from which the
 7342 administrator of the TOE **may** configure the number of unsuccessful authentication attempts.
 7343 The number of authentication attempts **should** be less than or equal to the upper bound and
 7344 greater or equal to the lower bound values.

7345 **G.2.2.2.3 Selection**

7346 In FIA_AFL.1.2, the PP/ST author **should** select whether the event of meeting or surpassing the
 7347 defined number of unsuccessful authentication attempts **shall** trigger an action by the TSF.

7348 **G.2.2.2.4 Assignment**

7349 In FIA_AFL.1.2, the PP/ST author **should** specify the actions to be taken in case the threshold is
 7350 met or surpassed, as selected. These actions **could** be disabling of an account for 5 minutes,
 7351 disabling the terminal for an increasing amount of time (2 to the power of the number of
 7352 unsuccessful attempts in seconds), or disabling of the account until unlocked by the

7353 administrator and simultaneously informing the administrator. The actions **should** specify the
 7354 measures and if applicable the duration of the measure (or the conditions under which the
 7355 measure will be ended).

7356 **G.3 Authentication proof of identity (FIA_API)**

7357 **G.3.1 User notes**

7358 The other families of the Class FIA describe only the authentication verification of users'
 7359 identity performed by the TOE and do not describe the functionality of the user to prove their
 7360 identity. The family FIA_API allows the specification the functionality allowing a TOE to prove
 7361 its own identity.

7362 **G.3.2 FIA_API.1 Authentication proof of identity**

7363 **Editor's Note:**

7364 Since no contributions were received for this text, the editors have proposed the text below. Please
 7365 review carefully.

7366 If no comments are received during the commenting period for this draft, the editors' proposal will be
 7367 accepted in the next draft.

7368 **G.3.2.1 User application notes**

7369 FIA_API.1 Authentication proof of identity allows the specification of the authentication
 7370 mechanism used to support proving the identity of the TOE to external entities.

7371 **G.3.2.2 Operations**

7372 **G.3.2.2.1 Assignment**

7373 The first assignment is where a PP/ST author specifies the authentication mechanism to be
 7374 used.

EXAMPLE

Examples of such an authentication method is "an Authentication Mechanism based on Triple-DES" and "Chip Authentication Protocol according to TR-03110"

7375

7376 The second assignment allows the PP/ST author to specify to what the proof of identity is
 7377 associated with. This can be an object, authorized user or a role.

7378 **Editors' Note:**

7379 Editors observe that in many STs using this component the second completion is for "TOE" which is
 7380 neither an object, authorized user or a role. Should FIA_API.1.1 be updated to allow for the specification
 7381 of TOE in the assignment?

7382 **G.4 User attribute definition (FIA_ATD)**

7383 **G.4.1 User notes**

7384 All authorized users **may** have a set of security attributes, other than the user's identity, that are
 7385 used to enforce the SFRs. This family defines the requirements for associating user security
 7386 attributes with users as needed to support the TSF in making security decisions.

7387 There are dependencies on the individual security policy (SFP) definitions. These individual
 7388 definitions **should** contain the listing of attributes that are necessary for policy enforcement.

7389 **G.4.2 FIA_ATD.1 User attribute definition**

7390 **G.4.2.1 User application notes**

7391 This component specifies the security attributes that **should** be maintained at the level of the
7392 user. This means that the security attributes listed are assigned to and **can** be changed at the
7393 level of the user. In other words, changing a security attribute in this list associated with a user
7394 **should** have no impact on the security attributes of any other user.

7395 In case security attributes belong to a group of users (such as Capability List for a group), the
7396 user will need to have a reference (as security attribute) to the relevant group.

7397 **G.4.2.2 Operations**

7398 **G.4.2.2.1 Assignment**

7399 In FIA_ATD.1.1, the PP/ST author **should** specify the security attributes that are associated to an
7400 individual user.

EXAMPLE

An example of such a list is {"clearance", "group identifier", "rights"}.

7401 **G.5 Specification of secrets (FIA_SOS)**

7402 **G.5.1 User notes**

7403 This family defines requirements for mechanisms that enforce defined quality metrics on
7404 provided secrets and generate secrets to satisfy the defined metric. Examples of such
7405 mechanisms **may** include automated checking of user supplied passwords, or automated
7406 password generation.

7407 A secret **can** be generated outside the TOE

EXAMPLE

An example of a secret generated outside of the TOE could be one that is selected by the user and introduced in the TOE.

7408 In such cases, the FIA_SOS.1 Verification of secrets component **can** be used to ensure that the
7409 external generated secret adheres to certain standards, for example a minimum size, not
7410 present in a dictionary, and/or not previously used.

7411 Secrets **can** also be generated by the TOE. In those cases, the FIA_SOS.2 TSF Generation of
7412 secrets component **can** be used to require the TOE to ensure that the secrets that will adhere to
7413 some specified metrics.

7414 Secrets contain the authentication data provided by the user for an authentication mechanism
7415 that is based on knowledge the user possesses. When cryptographic keys are employed, the
7416 class FCS: Cryptographic support **should** be used instead of this family.

7417 **G.5.2 FIA_SOS.1 Verification of secrets**

7418 **G.5.2.1 User application notes**

7419 Secrets **can** be generated by the user. This component ensures that those user generated secrets
7420 **can** be verified to meet a certain quality metric.

7421 **G.5.2.2 Operations**

7422 **G.5.2.2.1 Assignment**

7423 In FIA_SOS.1.1, the PP/ST author **should** provide a defined quality metric. The quality metric
7424 specification **can** be as simple as a description of the quality checks to be performed, or as
7425 formal as a reference to a government published standard that defines the quality metrics that
7426 secrets must meet.

7427

EXAMPLE

quality metrics **could** include a description of the alphanumeric structure of acceptable secrets and/or the space size that acceptable secrets must meet.

7428 **G.5.3 FIA_SOS.2 TSF Generation of secrets**7429 **G.5.3.1 User application notes**

7430 This component allows the TSF to generate secrets for specific functions such as authentication
7431 by means of passwords.

7432 When a pseudo-random number generator is used in a secret generation algorithm, it **should**
7433 accept as input random data that would provide output that has a high degree of
7434 unpredictability. This random data (seed) **can** be derived from a number of available
7435 parameters such as a system clock, system registers, date, time, etc. The parameters **should** be
7436 selected to ensure that the number of unique seeds that **can** be generated from these inputs
7437 **should** be at least equal to the minimum number of secrets that must be generated.

7438 **G.5.3.2 Operations**7439 **G.5.3.2.1 Assignment**

7440 In FIA_SOS.2.1, the PP/ST author **should** provide a defined quality metric. The quality metric
7441 specification **can** be as simple as a description of the quality checks to be performed or as
7442 formal as a reference to a government published standard that defines the quality metrics that
7443 secrets must meet.

EXAMPLE

quality metrics **could** include a description of the alphanumeric structure of acceptable secrets and/or the space size that acceptable secrets must meet.

7444 In FIA_SOS.2.2, the PP/ST author **should** provide a list of TSF functions for which the TSF
7445 generated secrets must be used. An example of such a function **could** include a password-based
7446 authentication mechanism.

7447 **G.6 User authentication (FIA_UAU)**7448 **G.6.1 User notes**

7449 This family defines the types of user authentication mechanisms supported by the TSF. This
7450 family defines the required attributes on which the user authentication mechanisms must be
7451 based.

7452 **G.6.2 FIA_UAU.1 Timing of authentication**7453 **G.6.2.1 User application notes**

7454 This component requires that the PP/ST author define the TSF-mediated actions that **can** be
7455 performed by the TSF on behalf of the user before the claimed identity of the user is
7456 authenticated. The TSF-mediated actions **should** have no security concerns with users
7457 incorrectly identifying themselves prior to being authenticated. For all other TSF-mediated
7458 actions not in the list, the user must be authenticated before the action **can** be performed by the
7459 TSF on behalf of the user.

7460 This component cannot control whether the actions **can** also be performed before the
7461 identification took place. This requires the use of either FIA_UID.1 Timing of identification or
7462 FIA_UID.2 User identification before any action with the appropriate assignments.

7463 **G.6.2.2 Operations**7464 **G.6.2.2.1 Assignment**

7465 In FIA_UAU.1.1, the PP/ST author **should** specify a list of TSF-mediated actions that **can** be
7466 performed by the TSF on behalf of a user before the claimed identity of the user is
7467 authenticated. This list cannot be empty. If no actions are appropriate, component FIA_UAU.2
7468 User authentication before any action **should** be used instead.

EXAMPLE

Such an action might include the request for help on the login procedure.

7469 **G.6.3 FIA_UAU.2 User authentication before any action**

7470 **G.6.3.1 User application notes**

7471 This component requires that a user is authenticated before any other TSF-mediated action **can**
7472 take place on behalf of that user.

7473 **G.6.4 FIA_UAU.3 Unforgeable authentication**

7474 **G.6.4.1 User application notes**

7475 This component addresses requirements for mechanisms that provide protection of
7476 authentication data. Authentication data that is copied from another user, or is in some way
7477 constructed **should** be detected and/or rejected. These mechanisms provide confidence that
7478 users authenticated by the TSF are actually who they claim to be.

7479 This component **may** be useful only with authentication mechanisms that are based on
7480 authentication data that cannot be shared. It is impossible for a TSF to detect or prevent the
7481 sharing of passwords outside the control of the TSF.

EXAMPLE

An example of authentication data that cannot be shared is biometrics

7482 **Editors' Note**

7483 **Is this a good example? Editors' suggest that replay attacks could be "sharing" biometrics.**

7484 **G.6.4.2 Operations**

7485 **G.6.4.2.1 Selection**

7486 In FIA_UAU.3.1, the PP/ST author **should** specify whether the TSF will detect, prevent, or detect
7487 and prevent forging of authentication data.

7488 In FIA_UAU.3.2, the PP/ST author **should** specify whether the TSF will detect, prevent, or detect
7489 and prevent copying of authentication data.

7490 **G.6.5 FIA_UAU.4 Single-use authentication mechanisms**

7491 **G.6.5.1 User application notes**

7492 This component addresses requirements for authentication mechanisms based on single-use
7493 authentication data. Single-use authentication data **can** be something the user has or knows, but
7494 not something the user is.

EXAMPLE

Single-use authentication data include single-use passwords, encrypted time-stamps,
and/or random numbers from a secret lookup table.

7495 The PP/ST author **can** specify to which authentication mechanism(s) this requirement applies.

7496 **G.6.5.2 Operations**

7497 **G.6.5.2.1 Assignment**

7498 In FIA_UAU.4.1, the PP/ST author **should** specify the list of authentication mechanisms to which
 7499 this requirement applies. This assignment **can** be “all authentication mechanisms”. An example
 7500 of this assignment **could** be “the authentication mechanism employed to authenticate people on
 7501 the external network”.

7502 **G.6.6 FIA_UAU.5 Multiple authentication mechanisms**

7503 **G.6.6.1 User application notes**

7504 The use of this component allows specification of requirements for more than one
 7505 authentication mechanism to be used within a TOE. For each distinct mechanism, applicable
 7506 requirements must be chosen from the FIA: Identification and authentication class to be applied
 7507 to each mechanism. It is possible that the same component **could** be selected multiple times in
 7508 order to reflect different requirements for the different use of the authentication mechanism.

7509 The management functions in the class FMT **may** provide maintenance capabilities for the set of
 7510 authentication mechanisms, as well as the rules that determine whether the authentication was
 7511 successful.

7512 To allow anonymous users to interact with the TOE, a “none” authentication mechanism **can** be
 7513 incorporated. The use of such access **should** be clearly explained in the rules of FIA_UAU.5.2.

7514 **G.6.6.2 Operations**

7515 **G.6.6.2.1 Assignment**

7516 In FIA_UAU.5.1, the PP/ST author **should** define the available authentication mechanisms.

EXAMPLE 1

Such a list **could** be: “none, password mechanism, biometric (retinal scan), S/key mechanism”.

7517 In FIA_UAU.5.2, the PP/ST author **should** specify the rules that describe how the authentication
 7518 mechanisms provide authentication and when each is to be used. This means that for each
 7519 situation the set of mechanisms that might be used for authenticating the user must be
 7520 described.

EXAMPLE 2

A list of such rules is: “if the user has special privileges a password mechanism and a biometric mechanism both **shall** be used, with success only if both succeed; for all other users a password mechanism **shall** be used.”

7521 The PP/ST author might give the boundaries within which the authorized administrator **may**
 7522 specify specific rules. An example of a rule is: “the user **shall** always be authenticated by means
 7523 of a token; the administrator might specify additional authentication mechanisms that also
 7524 must be used.” The PP/ST author also might choose not to specify any boundaries but leave the
 7525 authentication mechanisms and their rules completely up to the authorized administrator.

7526 **G.6.7 FIA_UAU.6 Re-authenticating**

7527 **G.6.7.1 User application notes**

7528 This component addresses potential needs to re-authenticate users at defined points in time.
 7529 These **may** include user requests for the TSF to perform security relevant actions, as well as
 7530 requests from non-TSF entities for re-authentication.

EXAMPLE

A server application requesting that the TSF re-authenticate the client it is serving.

7531 **G.6.7.2 Operations**

7532 **G.6.7.2.1 Assignment**

7533 In FIA_UAU.6.1, the PP/ST author **should** specify the list of conditions requiring re-
7534 authentication. This list **could** include a specified user inactivity period that has elapsed, the
7535 user requesting a change in active security attributes, or the user requesting the TSF to perform
7536 some security critical function.

7537 The PP/ST author might give the boundaries within which the re-authentication **should** occur
7538 and leave the specifics to the authorized administrator.

EXAMPLE

“the user **shall** always be re-authenticated at least once a day; the administrator might specify that the re-authentication **should** happen more often but not more often than once every 10 minutes.”

7539 **G.6.8 FIA_UAU.7 Protected authentication feedback**

7540 **G.6.8.1 User application notes**

7541 This component addresses the feedback on the authentication process that will be provided to
7542 the user. In some systems, the feedback consists of indicating how many characters have been
7543 typed but not showing the characters themselves, in other systems even this information might
7544 not be appropriate.

7545 This component requires that the authentication data is not provided as-is back to the user. In a
7546 workstation environment, it **could** display a “dummy” for each password character provided,
7547 and not the original character.

Example

A “dummy” **could** be a star “*” character.

7548

7549 **G.6.8.2 Operations**

7550 **G.6.8.2.1 Assignment**

7551 In FIA_UAU.7 Protected authentication feedback, the PP/ST author **should** specify the feedback
7552 related to the authentication process that will be provided to the user.

EXAMPLE

A feedback assignment **could** be “the number of characters typed”, another type of feedback is “the authentication mechanism that failed the authentication”.

7553 **G.7 User identification (FIA_UID)**

7554 **G.7.1 User notes**

7555 This family defines the conditions under which users are required to identify themselves before
7556 performing any other actions that are to be mediated by the TSF and that require user
7557 identification.

7558 **G.7.2 FIA_UID.1 Timing of identification**

7559 **G.7.2.1 User application notes**

7560 This component poses requirements for the user to be identified. The PP/ST author **can** indicate
7561 specific actions that **can** be performed before the identification takes place.

7562 If FIA_UID.1 Timing of identification is used, the TSF-mediated actions mentioned in FIA_UID.1
7563 Timing of identification **should** also appear in this FIA_UAU.1 Timing of authentication.

7564 **G.7.2.2 Operations**

7565 **G.7.2.2.1 Assignment**

7566 In FIA_UID.1.1, the PP/ST author **should** specify a list of TSF-mediated actions that **can** be
 7567 performed by the TSF on behalf of a user before the user has to identify itself. If no actions are
 7568 appropriate, component FIA_UID.2 User identification before any action **should** be used instead.
 7569 An example of such an action might include the request for help on the login procedure.

7570 **G.7.3 FIA_UID.2 User identification before any action**

7571 **G.7.3.1 User application notes**

7572 In this component users will be identified. A user is not allowed by the TSF to perform any
 7573 action before being identified.

7574 **G.8 User-subject binding (FIA_USB)**

7575 **G.8.1 User notes**

7576 An authenticated user, in order to use the TOE, typically activates a subject. The user's security
 7577 attributes are associated (totally or partially) with this subject. This family defines
 7578 requirements to create and maintain the association of the user's security attributes to a subject
 7579 acting on the user's behalf.

7580 **G.8.2 FIA_USB.1 User-subject binding**

7581 **G.8.2.1 User application notes**

7582 It is intended that a subject is acting on behalf of the user who caused the subject to come into
 7583 being or to be activated to perform a certain task.

7584 Therefore, when a subject is created, that subject is acting on behalf of the user who initiated
 7585 the creation. In cases where anonymity is used, the subject is still acting on behalf of a user, but
 7586 the identity of that user is unknown. A special category of subjects is those subjects that serve
 7587 multiple users. In such cases the user that created this subject is assumed to be the "owner".

EXAMPLE

An example of a user is a server process.

7588 **G.8.2.2 Operations**

7589 **G.8.2.2.1 Assignment**

7590 In FIA_USB.1.1, the PP/ST author **should** specify a list of the user security attributes that are to
 7591 be bound to subjects.

7592 In FIA_USB.1.2, the PP/ST author **should** specify any rules that are to apply upon initial
 7593 association of attributes with subjects, or "none".

7594 In FIA_USB.1.3, the PP/ST author **should** specify any rules that are to apply when changes are
 7595 made to the user security attributes associated with subjects acting on behalf of users, or
 7596 "none".

Annex H (normative)

Class FMT: Security management- application notes

7597
7598
7599
7600

7601 H.1 General information

7602 This class specifies the management of several aspects of the TSF: security attributes, TSF data
7603 and functions in the TSF. The different management roles and their interaction, such as
7604 separation of capability, **can** also be specified.

7605 In an environment where the TOE is made up of multiple physically separated parts, the timing
7606 issues with respect to propagation of security attributes, TSF data, and function modification
7607 become very complex, especially if the information is required to be replicated across the parts
7608 of the TOE. This **should** be considered when selecting components such as FMT_REV.1
7609 Revocation, or FMT_SAE.1 Time-limited authorization, where the behaviour might be impaired.
7610 In such situations, use of components from Internal TOE TSF data replication consistency
7611 (FPT_TRC) is advisable.

7612 Editors' Notes

7613 The newly added family of FMT_LIM as well as other families need to be discussed in the general
7614 information part.

7615 The text below has been suggested by the editors.

7616 The FMT_LIM family provides requirements that allow the specification of a policy that limits
7617 the capabilities and the availability of TSF functions. This is useful when a PP/ST author needs
7618 to enforce design principles such as least privilege and attack surface minimization.

7619 Note These, and other architectural and design principles along with appropriate evaluation considerations
7620 are discussed in ISO/IEC 19249, Information technology — Security techniques — Catalogue of architectural and
7621 design principles for secure products, systems, and applications.

7622 H.2 Limited capabilities and availability (FMT_LIM)

7623 H.2.1 User notes

7624 The functional requirements FMT_LIM.1 and FMT_LIM.2 assume that there are two types of
7625 mechanisms (limitation of capabilities and limitation of availability) which together **shall**
7626 provide protection in order to enforce the policy. This also allows that

- 7627 a) the TSF is provided without restrictions in the product in its user environment but its
7628 capabilities are so limited that the policy is enforced or conversely
- 7629 b) the TSF is designed with high functionality but is removed or disabled in the product in
7630 its user environment.

7631 The combination of both requirements **shall** enforce the policy.

7632 H.2.2 FMT_LIM.1 Limited capabilities

7633 H.2.2.1 User application notes

7634 An example of a limited capability JTAG interface enablement, which **could** be either enabled or
7635 disabled.

7636 H.2.2.2 Operations

7637 H.2.2.2.1 Assignment

7638 In FMT_LIM.1.1, the PP/ST author **should** specify the limited capability policy.

7639 **H.2.3 FMT_LIM.2 Limited availability**7640 **H.2.3.1 User application notes**

EXAMPLE

An example of a limited availability is JTAG interface enablement, which could be either enabled or disabled before operational use of the TOE.

7641 **H.2.3.2 Operations**7642 **H.2.3.2.1 Assignment**

7643 In FMT_LIM.2.1, the PP/ST author should specify the limited availability policy.

7644 **H.3 Management of functions in TSF (FMT_MOF)**7645 **H.3.1 User notes**

7646 The TSF management functions enable authorized users to set up and control the secure
7647 operation of the TOE. These administrative functions typically fall into a number of different
7648 categories:

- 7649 a) Management functions that relate to access control, accountability and
7650 authentication controls enforced by the TOE. For example, definition and update of
7651 user security characteristics or definition and update of auditing system controls,
7652 definition and update of per-user policy attributes, definition of known system
7653 access control labels, and control and management of user groups.

EXAMPLE

User security characteristics: unique identifiers associated with user names, user accounts, system entry parameters

Auditing system controls: selection of audit events, management of audit trails, audit trail analysis, and audit report generation

User policy attributes: user clearance

- 7654 b) Management functions that relate to controls over availability. For example,
7655 definition and update of availability parameters or resource quotas.
- 7656 c) Management functions that relate to general installation and configuration. For
7657 example, TOE configuration, manual recovery, installation of TOE security fixes (if
7658 any), repair and reinstallation of hardware.
- 7659 d) Management functions that relate to routine control and maintenance of TOE
7660 resources. For example, enabling and disabling peripheral devices, mounting of
7661 removable storage media, backup, and recovery.

7662 NOTE These functions need to be present in a TOE based on the families included in the PP or ST. It is the
7663 responsibility of the PP/ST author to ensure that adequate functions will be provided to manage the TOE in a secure
7664 fashion.

7665 The TSF might contain functions that **can** be controlled by an administrator. For example, the
7666 auditing functions **could** be switched off, the time synchronization **could** be switchable, and/or
7667 the authentication mechanism **could** be modifiable.

7668 **H.3.2 FMT_MOF.1 Management of security functions behaviour**7669 **H.3.2.1 User application notes**

7670 This component allows identified roles to manage the security functions of the TSF. This might
7671 entail obtaining the current status of a security function, disabling, or enabling the security
7672 function, or modifying the behaviour of the security function.

EXAMPLE

modifying the behaviour of the security functions is changing of authentication mechanisms.

7673 **H.3.2.2 Operations**

7674 **H.3.2.2.1 Selection**

7675 In FMT_MOF.1.1, the PP/ST author **should** select whether the role **can** determine the behaviour
7676 of, disable, enable, and/or modify the behaviour of the security functions.

7677 **H.3.2.2.2 Assignment**

7678 In FMT_MOF.1.1, the PP/ST author **should** specify the functions that **can** be modified by the
7679 identified roles. Examples include auditing and time determination.

7680 In FMT_MOF.1.1, the PP/ST author **should** specify the roles that are allowed to modify the
7681 functions in the TSF. The possible roles are specified in FMT_SMR.1 Security roles.

7682 **H.4 Management of security attributes (FMT_MSA)**

7683 **H.4.1 User notes**

7684 This family defines the requirements on the management of security attributes.

7685 Security attributes affect the behaviour of the TSF.

EXAMPLE

Examples of security attributes are the groups to which a user belongs, the roles he/she might assume, the priority of a process (subject), and the rights belonging to a role or a user.

7686 These security attributes might need to be managed by the user, a subject, a specific authorized
7687 user (a user with explicitly given rights for this management) or inherit values according to a
7688 given policy/set of rules.

7689 It is noted that the right to assign rights to users is itself a security attribute and/or potentially
7690 subject to management by FMT_MSA.1 Management of security attributes.

7691 FMT_MSA.2 Secure security attributes **can** be used to ensure that any accepted combination of
7692 security attributes is within a secure state. The definition of what “secure” means is left to the
7693 TOE guidance.

7694 In some instances, subjects, objects, or user accounts are created. If no explicit values for the
7695 related security attributes are given, default values need to be used. FMT_MSA.1 Management of
7696 security attributes **can** be used to specify that these default values **can** be managed.

7697 **H.4.2 FMT_MSA.1 Management of security attributes**

7698 **H.4.2.1 User application notes**

7699 This component allows users acting in certain roles to manage identified security attributes.
7700 The users are assigned to a role within the component FMT_SMR.1 Security roles.

7701 The default value of a parameter is the value the parameter takes when it is instantiated
7702 without specifically assigned values. An initial value is provided during the instantiation
7703 (creation) of a parameter and overrides the default value.

7704 **H.4.2.2 Operations**

7705 **H.4.2.2.1 Assignment**

7706 In FMT_MSA.1.1, the PP/ST author **should** list the access control SFP(s) or the information flow
7707 control SFP(s) for which the security attributes are applicable.

7708 **H.4.2.2.2 Selection**

7709 In FMT_MSA.1.1, the PP/ST author **should** specify the operations that **can** be applied to the
 7710 identified security attributes. The PP/ST author **can** specify that the role **can** modify the default
 7711 value (change_default), query, modify the security attribute, delete the security attributes
 7712 entirely or define their own operation.

7713 H.4.2.2.3 Assignment

7714 In FMT_MSA.1.1, the PP/ST author **should** specify the security attributes that **can** be operated
 7715 on by the identified roles. It is possible for the PP/ST author to specify that the default value
 7716 such as default access-rights **can** be managed.

EXAMPLE

Examples of these security attributes are user-clearance, priority of service level, access control list, default access rights.

7717 In FMT_MSA.1.1, the PP/ST author **should** specify the roles that are allowed to operate on the
 7718 security attributes. The possible roles are specified in FMT_SMR.1 Security roles.

7719 In FMT_MSA.1.1, if selected, the PP/ST author **should** specify which other operations the role
 7720 **could** perform.

EXAMPLE

An example of such an operation **could** be “create”.

7721 H.4.3 FMT_MSA.2 Secure security attributes

7722 H.4.3.1 User application notes

7723 This component contains requirements on the values that **can** be assigned to security attributes.
 7724 The assigned values **should** be such that the TOE will remain in a secure state.

7725 The definition of what “secure” means is not answered in this component but is left to the
 7726 development of the TOE and the resulting information in the guidance. An example **could** be
 7727 that if a user account is created, it **should** have a non-trivial password.

7728 H.4.3.2 Operations

7729 H.4.3.2.1 Assignment

7730 In FMT_MSA.2.1, the PP/ST author **should** specify the list of security attributes that require only
 7731 secure values to be provided.

7732 H.4.4 FMT_MSA.3 Static attribute initialization

7733 H.4.4.1 User application notes

7734 This component requires that the TSF provide default values for relevant object security
 7735 attributes, which **can** be overridden by an initial value. It **may** still be possible for a new object
 7736 to have different security attributes at creation if a mechanism exists to specify the permissions
 7737 at time of creation.

7738 H.4.4.2 Operations

7739 H.4.4.2.1 Assignment

7740 In FMT_MSA.3.1, the PP/ST author **should** list the access control SFP or the information flow
 7741 control SFP for which the security attributes are applicable.

7742 H.4.4.2.2 Selection

7743 In FMT_MSA.3.1, the PP/ST author **should** select whether the default property of the access
 7744 control attribute will be restrictive, permissive, or another property. Only one of these options
 7745 **may** be chosen.

7746 H.4.4.2.3 Assignment

7747 In FMT_MSA.3.1, if the PP/ST author selects another property, the PP/ST author **should** specify
7748 the desired characteristics of the default values.

7749 In FMT_MSA.3.2, the PP/ST author **should** specify the roles that are allowed to modify the
7750 values of the security attributes. The possible roles are specified in FMT_SMR.1 Security roles.

7751 **H.4.5 FMT_MSA.4 Security attribute value inheritance**

7752 **H.4.5.1 User application notes**

7753 This component requires specification of the set of rules through which the security attribute
7754 inherits values and the conditions to be met for these rules to be applied.

7755 **H.4.5.2 Operations**

7756 **H.4.5.2.1 Assignment**

7757 In FMT_MSA.4.1, the PP/ST author specifies the rules governing the value that will be inherited
7758 by the specified security attribute, including the conditions that are to be met for the rules to be
7759 applied.

EXAMPLE

For example, if a new file or directory is created (in a multilevel filesystem), its label is the label at which the user is logged in at the time it is created.

7760 **H.5 Management of TSF data (FMT_MTD)**

7761 **H.5.1 User notes**

7762 This component imposes requirements on the management of TSF data. Examples of TSF data
7763 are the current time and the audit trail.

EXAMPLE

this family allows the specification of whom **can** read, delete, or create the audit trail.

7764 **H.5.2 FMT_MTD.1 Management of TSF data**

7765 **H.5.2.1 User application notes**

7766 This component allows users with a certain role to manage values of TSF data. The users are
7767 assigned to a role within the component FMT_SMR.1 Security roles.

7768 The default value of a parameter is the values the parameter takes when it is instantiated
7769 without specifically assigned values. An initial value is provided during the instantiation
7770 (creation) of a parameter and overrides the default value.

7771 **H.5.2.2 Operations**

7772 **H.5.2.2.1 Selection**

7773 In FMT_MTD.1.1, the PP/ST author **should** specify the operations that **can** be applied to the
7774 identified TSF data. The PP/ST author **can** specify that the role **can** modify the default value
7775 (change_default), clear, query or modify the TSF data, or delete the TSF data entirely. If so
7776 desired the PP/ST author **could** specify any type of operation. To clarify “clear TSF data” means
7777 that the content of the TSF data is removed, but that the entity that stores the TSF data remains
7778 in the TOE.

7779 **H.5.2.2.2 Assignment**

7780 In FMT_MTD.1.1, the PP/ST author **should** specify the TSF data that **can** be operated on by the
7781 identified roles. It is possible for the PP/ST author to specify that the default value **can** be
7782 managed.

7783 In FMT_MTD.1.1, the PP/ST author **should** specify the roles that are allowed to operate on the
7784 TSF data. The possible roles are specified in FMT_SMR.1 Security roles.

7785 In FMT_MTD.1.1, if selected, the PP/ST author **should** specify which other operations the role
7786 **could** perform.

EXAMPLE

An example of an operation is “create”.

7787 H.5.3 FMT_MTD.2 Management of limits on TSF data

7788 H.5.3.1 User application notes

7789 This component specifies limits on TSF data, and actions to be taken if these limits are
7790 exceeded. This component will allow limits on the size of the audit trail to be defined, and
7791 specification of the actions to be taken when these limits are exceeded.

7792 H.5.3.2 Operations

7793 H.5.3.2.1 Assignment

7794 In FMT_MTD.2.1, the PP/ST author **should** specify the TSF data that **can** have limits, and the
7795 value of those limits. An example of such TSF data is the number of users logged-in.

7796 In FMT_MTD.2.1, the PP/ST author **should** specify the roles that are allowed to modify the limits
7797 on the TSF data and the actions to be taken. The possible roles are specified in FMT_SMR.1
7798 Security roles.

7799 In FMT_MTD.2.2, the PP/ST author **should** specify the actions to be taken if the specified limit
7800 on the specified TSF data is exceeded.

EXAMPLE

An example of such a TSF action is that the authorized user is informed and an audit record is generated.

7801 H.5.4 FMT_MTD.3 Secure TSF data

7802 H.5.4.1 User application notes

7803 This component covers requirements on the values that **can** be assigned to TSF data. The
7804 assigned values **should** be such that the TOE will remain in a secure state.

7805 The definition of what “secure” means is not answered in this component but is left to the
7806 development of the TOE and the resulting information in the guidance.

7807 H.5.4.2 Operations

7808 H.5.4.2.1 Assignment

7809 In FMT_MTD.3.1, the PP/ST author **should** specify what TSF data require only secure values to
7810 be accepted.

7811 H.6 Revocation (FMT_REV)

7812 H.6.1 User notes

7813 This family addresses revocation of security attributes for a variety of entities within a TOE.

7814 H.6.2 FMT_REV.1 Revocation

7815 H.6.2.1 User application notes

7816 This component specifies requirements on the revocation of rights. It requires the specification
7817 of the revocation rules. Examples are:

- 7818 a) Revocation will take place on the next login of the user;
- 7819 b) Revocation will take place on the next attempt to open the file;
- 7820 c) Revocation will take place within a fixed time. This might mean that all open
- 7821 connections are re-evaluated every x minutes.

7822 **H.6.2.2 Operations**

7823 **H.6.2.2.1 Assignment**

7824 In FMT_REV.1.1, the PP/ST author **should** specify which security attributes are to be revoked
7825 when a change is made to the associated object/subject/user/other resource.

7826 **H.6.2.2.2 Selection**

7827 In FMT_REV.1.1, the PP/ST author **should** specify whether the ability to revoke security
7828 attributes from users, subjects, objects, or any additional resources **shall** be provided by the
7829 TSF.

7830 **H.6.2.2.3 Assignment**

7831 In FMT_REV.1.1, the PP/ST author **should** specify the roles that are allowed to modify the
7832 functions in the TSF. The possible roles are specified in FMT_SMR.1 Security roles.

7833 In FMT_REV.1.1, the PP/ST author **should**, if additional resources is selected, specify whether
7834 the ability to revoke their security attributes **shall** be provided by the TSF.

7835 In FMT_REV.1.2, the PP/ST author **should** specify the revocation rules. Examples of these rules
7836 **could** include: “prior to the next operation on the associated resource”, or “for all new subject
7837 creations”.

7838 **H.7 Security attribute expiration (FMT_SAE)**

7839 **H.7.1 User notes**

7840 This family addresses the capability to enforce time limits for the validity of security attributes.
7841 This family **can** be applied to specify expiration requirements for access control attributes,
7842 identification and authentication attributes, certificates, audit attributes, etc.

EXAMPLE

An example of a certificate is key certificates such as ANSI X509.

7843 **H.7.2 FMT_SAE.1 Time-limited authorization**

7844 **H.7.2.1 Operations**

7845 **H.7.2.1.1 Assignment**

7846 In FMT_SAE.1.1, the PP/ST author **should** provide the list of security attributes for which
7847 expiration is to be supported.

EXAMPLE

An example of such an attribute might be a user's security clearance.

7848 In FMT_SAE.1.1, the PP/ST author **should** specify the roles that are allowed to modify the
7849 security attributes in the TSF. The possible roles are specified in FMT_SMR.1 Security roles.

7850 In FMT_SAE.1.2, the PP/ST author **should** provide a list of actions to be taken for each security
7851 attribute when it expires. An example might be that the user's security clearance, when it
7852 expires, is set to the lowest allowable clearance on the TOE. If immediate revocation is desired
7853 by the PP/ST, the action “immediate revocation” **should** be specified.

7854 **H.8 Specification of Management Functions (FMT_SMF)**

7855 H.8.1 User notes

7856 This family allows the specification of the management functions to be provided by the TOE.
7857 Each security management function that is listed in fulfilling the assignment is either security
7858 attribute management, TSF data management, or security function management.

7859 H.8.2 FMT_SMF.1 Specification of Management Functions

7860 H.8.2.1 User application notes

7861 This component specifies the management functions to be provided.

7862 PP/ST authors **should** consult the “Management” subclauses for components included in their
7863 PP/ST to provide a basis for the management functions to be listed via this component.

7864 H.8.2.2 Operations

7865 H.8.2.2.1 Assignment

7866 In FMT_SMF.1.1, the PP/ST author **should** specify the management functions to be provided by
7867 the TSF, either security attribute management, TSF data management, or security function
7868 management.

7869 H.9 Security management roles (FMT_SMR)

7870 H.9.1 User notes

7871 This family reduces the likelihood of damage resulting from users abusing their authority by
7872 taking actions outside their assigned functional responsibilities. It also addresses the threat that
7873 inadequate mechanisms have been provided to securely administer the TSF.

7874 This family requires that information be maintained to identify whether a user is authorized to
7875 use a particular security-relevant administrative function.

7876 Some management actions **can** be performed by users, others only by designated people within
7877 the organization. This family allows the definition of different roles, such as owner, auditor,
7878 administrator, daily-management.

7879 The roles as used in this family are security related roles. Each role **can** encompass an extensive
7880 set of capabilities or **can** be a single right. This family defines the roles. The capabilities of the
7881 role are defined in Limited capabilities and availability (FMT_LIM), Management of security
7882 attributes (FMT_MSA) and Management of TSF data (FMT_MTD).

EXAMPLE 1

Set of capabilities: root in UNIX

Single right: right to read a single object such as the helpfile.

7883 Some type of roles might be mutually exclusive.

EXAMPLE 2

The daily-management might be able to define and activate users but might not be able to
remove users (which is reserved for the administrator (role)).

7884 This class will allow policies such as two-person control to be specified.

7885 H.9.2 FMT_SMR.1 Security roles

7886 H.9.2.1 User application notes

7887 This component specifies the different roles that the TSF **should** recognize. Often the system
7888 distinguishes between the owner of an entity, an administrator, and other users.

7889 H.9.2.2 Operations

7890 H.9.2.2.1 Assignment

7891 In FMT_SMR.1.1, the PP/ST author **should** specify the roles that are recognized by the system.
7892 These are the roles that users **could** occupy with respect to security.

EXAMPLE
Examples of roles are: owner, auditor, and administrator.

7893 H.9.3 FMT_SMR.2 Restrictions on security roles

7894 H.9.3.1 User application notes

7895 This component specifies the different roles that the TSF **should** recognize, and conditions on
7896 how those roles **could** be managed. Often the system distinguishes between the owner of an
7897 entity, an administrator, and other users.

7898 The conditions on those roles specify the interrelationship between the different roles, as well
7899 as restrictions on when the role **can** be assumed by a user.

7900 H.9.3.2 Operations

7901 H.9.3.2.1 Assignment

7902 In FMT_SMR.2.1, the PP/ST author **should** specify the roles that are recognized by the system.
7903 These are the roles that users **could** occupy with respect to security.

EXAMPLE 1
Examples of roles are: owner, auditor, and administrator.

7904 In FMT_SMR.2.3, the PP/ST author **should** specify the conditions that govern role assignment.
7905

EXAMPLE2
Examples of these conditions are: "an account cannot have both the auditor and administrator role" or "a user with the assistant role must also have the owner role".

7906 H.9.4 FMT_SMR.3 Assuming roles

7907 H.9.4.1 User application notes

7908 This component specifies that an explicit request must be given to assume the specific role.

7909 H.9.4.2 Operations

7910 H.9.4.2.1 Assignment

7911 In FMT_SMR.3.1, the PP/ST author **should** specify the roles that require an explicit request to be
7912 assumed.

7913

EXAMPLE
Examples of roles are: owner, auditor, and administrator.

7914
7915
7916
7917

Annex I (normative)

Class FPR: Privacy- application notes

7918 I.1 General information

7919 This class describes the requirements that **could** be levied to satisfy the users' privacy needs,
7920 while still allowing the system flexibility as far as possible to maintain sufficient control over
7921 the operation of the system.

7922 In the components of this class there is flexibility as to whether or not authorized users are
7923 covered by the required security functionality.

EXAMPLE 1

a PP/ST author might consider it appropriate not to require protection of the privacy of users against a suitably authorized user.

7924 This class, together with other classes (such as those concerned with audit, access control,
7925 trusted path, and non-repudiation) provides the flexibility to specify the desired privacy
7926 behaviour. On the other hand, the requirements in this class might impose limitations on the
7927 use of the components of other classes, such as FIA: Identification and authentication or FAU:
7928 Security audit.

EXAMPLE 2

If authorized users are not allowed to see the user identity (perhaps because of Anonymity or Pseudonymity), it will obviously not be possible to hold individual users accountable for any security relevant actions they perform that are covered by the privacy requirements. However, it may still be possible to include audit requirements in a PP/ST, where the fact that a particular security relevant event has occurred is more important than knowing who was responsible for it.

7929 Additional information is provided in the application notes for class FAU: Security audit, where
7930 it is explained that the definition of "identity" in the context of auditing **can** also be an alias or
7931 other information that **could** identify a user.

7932 This class describes four families: Anonymity, Pseudonymity, Unlinkability and Unobservability.
7933 Anonymity, Pseudonymity and Unlinkability have a complex interrelationship. When choosing a
7934 family, the choice **should** depend on the threats identified. For some types of privacy threats,
7935 pseudonymity will be more appropriate than anonymity.

EXAMPLE 3

If there is a requirement for auditing.

7936 In addition, some types of privacy threats are best countered by a combination of components
7937 from several families.

7938 All families assume that a user does not explicitly perform an action that discloses the user's
7939 own identity.

EXAMPLE 4

The TSF is not expected to screen the user name in electronic messages or databases.

7940 All families in this class have components that are scoped through operations. These operations
7941 allow the PP/ST author to state the cooperating users/subjects to which the TSF must be
7942 resistant.

EXAMPLE 5

An instantiation of anonymity **could** be: "The TSF shall ensure that the users and/or subjects are unable to determine the user identity bound to the teleconsulting application".

It is noted that the TSF should not only provide this protection against individual users, but also against users cooperating to obtain the information.

7943 **I.2 Anonymity (FPR_ANO)**

7944 **I.2.1 User notes**

7945 Anonymity ensures that a subject **may** use a resource or service without disclosing its user
7946 identity.

7947 The intention of this family is to specify that a user or subject might take action without
7948 releasing its user identity to others such as users, subjects, or objects. The family provides the
7949 PP/ST author with a means to identify the set of users that cannot see the identity of someone
7950 performing certain actions.

7951 Therefore, if a subject, using anonymity, performs an action, another subject will not be able to
7952 determine either the identity or even a reference to the identity of the user employing the
7953 subject. The focus of the anonymity is on the protection of the user's identity, not on the
7954 protection of the subject identity; hence, the identity of the subject is not protected from
7955 disclosure.

7956 Although the identity of the subject is not released to other subjects or users, the TSF is not
7957 explicitly prohibited from obtaining the users identity. In case the TSF is not allowed to know
7958 the identity of the user, FPR_ANO.2 Anonymity without soliciting information **could** be invoked.
7959 In that case, the TSF **should** not request the user information.

7960 The interpretation of "determine" **should** be taken in the broadest sense of the word.

7961 The Components leveling and description distinguishes between the users and an authorized
7962 user. An authorized user is often excluded from the component, and therefore allowed to
7963 retrieve a user's identity. However, there is no specific requirement that an authorized user
7964 must be able to have the capability to determine the user's identity. For ultimate privacy, the
7965 components would be used to say that no user or authorized user **can** see the identity of anyone
7966 performing any action.

7967 Although some systems will provide anonymity for all services that are provided, other systems
7968 provide anonymity for certain subjects/operations. To provide this flexibility, an operation is
7969 included where the scope of the requirement is defined. If the PP/ST author wants to address
7970 all subjects/operations, the words "all subjects and all operations" **could** be provided.

7971 Possible applications include the ability to make enquiries of a confidential nature to public
7972 databases, respond to electronic polls, or make anonymous payments or donations.

EXAMPLE

Potential hostile users or subjects include providers, system operators, communication partners and users, who smuggle malicious parts (including malware) into systems. All of these users can investigate usage patterns (such as which users used which services) and misuse this information.

7973 **I.2.2 FPR_ANO.1 Anonymity**

7974 **I.2.2.1 User application notes**

7975 This component ensures that the identity of a user is protected from disclosure. There **may** be
7976 instances, however, that a given authorized user **can** determine who performed certain actions.
7977 This component gives the flexibility to capture either a limited or total privacy policy.

7978 **I.2.2.2 Operations**

7979 **I.2.2.2.1 Assignment**

7980 In FPR_ANO.1.1, the PP/ST author **should** specify the set of users and/or subjects against which
 7981 the TSF must provide protection. For example, even if the PP/ST author specifies a single user
 7982 or subject role, the TSF must not only provide protection against each individual user or subject
 7983 but must protect with respect to cooperating users and/or subjects.

EXAMPLE 1

A set of users **could** be a group of users which can operate under the same role or can all use the same process(es).

7984 In FPR_ANO.1.1, the PP/ST author **should** identify the list of subjects and/or operations and/or
 7985 objects where the real user name of the subject **should** be protected.

EXAMPLE 2

An example of an object is “the voting application”.

7986 **I.2.3 FPR_ANO.2 Anonymity without soliciting information**7987 **I.2.3.1 User application notes**

7988 This component is used to ensure that the TSF is not allowed to know the identity of the user.

7989 **I.2.3.2 Operations**7990 **I.2.3.2.1 Assignment**

7991 In FPR_ANO.2.1, the PP/ST author **should** specify the set of users and/or subjects against which
 7992 the TSF must provide protection. For example, even if the PP/ST author specifies a single user
 7993 or subject role, the TSF must not only provide protection against each individual user or subject
 7994 but must protect with respect to cooperating users and/or subjects.

EXAMPLE 1

A set of users **could** be a group of users which can operate under the same role or can all use the same process(es).

7995 In FPR_ANO.2.1, the PP/ST author **should** identify the list of subjects and/or operations and/or
 7996 objects where the real user name of the subject **should** be protected.

EXAMPLE 2

“the voting application”.

7997 In FPR_ANO.2.2, the PP/ST author **should** identify the list of services which are subject to the
 7998 anonymity requirement, for example, “the accessing of job descriptions”.

7999 In FPR_ANO.2.2, the PP/ST author **should** identify the list of subjects from which the real user
 8000 name of the subject **should** be protected when the specified services are provided.

8001 **I.3 Pseudonymity (FPR_PSE)**8002 **I.3.1 User notes**

8003 Pseudonymity ensures that a user **may** use a resource or service without disclosing its identity
 8004 but **can** still be accountable for that use. The user **can** be accountable by directly being related to
 8005 a reference (alias) held by the TSF, or by providing an alias that will be used for processing
 8006 purposes, such as an account number.

8007 In several respects, pseudonymity resembles anonymity. Both pseudonymity and anonymity
 8008 protect the identity of the user, but in pseudonymity a reference to the user's identity is
 8009 maintained for accountability or other purposes.

8010 The component FPR_PSE.1 Pseudonymity does not specify the requirements on the reference to
 8011 the user's identity. For the purpose of specifying requirements on this reference two sets of

8012 requirements are presented: FPR_PSE.2 Reversible pseudonymity and FPR_PSE.3 Alias
8013 pseudonymity.

8014 A way to use the reference is by being able to obtain the original user identity.

EXAMPLE 1

In a digital cash environment, it would be advantageous to be able to trace the user's identity when a check has been issued multiple times (i.e. fraud).

8015 In general, the user's identity needs to be retrieved under specific conditions. The PP/ST author
8016 might want to incorporate FPR_PSE.2 Reversible pseudonymity to describe those services.

8017 Another usage of the reference is as an alias for a user.

EXAMPLE 2

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

8018 Using these constructs above, digital money **can** be created using FPR_PSE.2 Reversible
8019 pseudonymity specifying that the user identity will be protected and, if so specified in the
8020 condition, that there be a requirement to trace the user identity if the digital money is spent
8021 twice. When the user is honest, the user identity is protected; if the user tries to cheat, the user
8022 identity **can** be traced.

8023 A different kind of system **could** be a digital credit card, where the user will provide a
8024 pseudonym that indicates an account from which the cash **can** be subtracted. In such cases, for
8025 example, FPR_PSE.3 Alias pseudonymity **could** be used. This component would specify that the
8026 user identity will be protected and, furthermore, that the same user will only get assigned
8027 values for which he/she has provided money (if so specified in the conditions).

8028 It **should** be realized that the more stringent components potentially cannot be combined with
8029 other requirements, such as identification and authentication or audit. The interpretation of
8030 "determine the identity" **should** be taken in the broadest sense of the word. The information is
8031 not provided by the TSF during the operation, nor **can** the entity determine the subject or the
8032 owner of the subject that invoked the operation, nor will the TSF record information, available
8033 to the users or subjects, which might release the user identity in the future.

8034 The intent is that the TSF not reveal any information that would compromise the identity of the
8035 user.

EXAMPLE 3

The identity of subjects acting on the user's behalf.

8036 The information that is considered to be sensitive depends on the effort an attacker is capable
8037 of spending.

8038 Possible applications include the ability to charge a caller for premium rate telephone services
8039 without disclosing his or her identity, or to be charged for the anonymous use of an electronic
8040 payment system.

EXAMPLE 4

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

8041 **I.3.2 FPR_PSE.1 Pseudonymity**8042 **I.3.2.1 User application notes**

8043 This component provides the user protection against disclosure of identity to other users. The
8044 user will remain accountable for its actions.

8045 **I.3.2.2 Operations**8046 **I.3.2.2.1 Assignment**

8047 In FPR_PSE.1.1, the PP/ST author **should** specify the set of users and/or subjects against which
8048 the TSF must provide protection. For example, even if the PP/ST author specifies a single user
8049 or subject role, the TSF must not only provide protection against each individual user or subject
8050 but must protect with respect to cooperating users and/or subjects.

EXAMPLE 1

A set of users **could** be a group of users which **can** operate under the same role or **can** all use the same process(es).

8051 In FPR_PSE.1.1, the PP/ST author **should** identify the list of subjects and/or operations and/or
8052 objects where the real user name of the subject **should** be protected.

EXAMPLE 2

“the accessing of job offers”.

8053 Note “objects” includes any other attributes that might enable another user or subject to derive the actual
8054 identity of the user.

8055 In FPR_PSE.1.2, the PP/ST author **should** identify the (one or more) number of aliases the TSF is
8056 able to provide.

8057 In FPR_PSE.1.2, the PP/ST author **should** identify the list of subjects to whom the TSF is able to
8058 provide an alias.

8059 **I.3.2.2.2 Selection**

8060 In FPR_PSE.1.3, the PP/ST author **should** specify whether the user alias is generated by the TSF
8061 or supplied by the user. Only one of these options **may** be chosen.

8062 **I.3.2.2.3 Assignment**

8063 In FPR_PSE.1.3, the PP/ST author **should** identify the metric to which the TSF-generated or
8064 user-generated alias **should** conform.

8065 **I.3.3 FPR_PSE.2 Reversible pseudonymity**8066 **I.3.3.1 User application notes**

8067 In this component, the TSF **shall** ensure that under specified conditions the user identity related
8068 to a provided reference **can** be determined.

8069 In FPR_PSE.1 Pseudonymity the TSF **shall** provide an alias instead of the user identity. When the
8070 specified conditions are satisfied, the user identity to which the alias belong **can** be determined.

EXAMPLE

Such a condition in an electronic cash environment is: “The TSF shall provide the notary a capability to determine the user identity based on the provided alias only under the conditions that a check has been issued twice.”

8071 **I.3.3.2 Operations**8072 **I.3.3.2.1 Assignment**

8073 In FPR_PSE.2.1, the PP/ST author **should** specify the set of users and/or subjects against which
8074 the TSF must provide protection.

EXAMPLE 1

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

8075 In FPR_PSE.2.1, the PP/ST author **should** identify the list of subjects and/or operations and/or
8076 objects where the real user name of the subject **should** be protected.

EXAMPLE 2

“the accessing of job offers”.

8077

8078 NOTE “objects” includes any other attributes that might enable another user or subject to derive the actual
8079 identity of the user.

8080 In FPR_PSE.2.2, the PP/ST author **should** identify the (one or more) number of aliases the TSF,
8081 is able to provide.

8082 In FPR_PSE.2.2, the PP/ST author **should** identify the list of subjects to whom the TSF is able to
8083 provide an alias.

8084 **I.3.3.2.2 Selection**

8085 In FPR_PSE.2.3, the PP/ST author **should** specify whether the user alias is generated by the TSF
8086 or supplied by the user. Only one of these options **may** be chosen.

8087 **I.3.3.2.3 Assignment**

8088 In FPR_PSE.2.3, the PP/ST author **should** identify the metric to which the TSF-generated or
8089 user-generated alias **should** conform.

8090 **I.3.3.2.4 Selection**

8091 In FPR_PSE.2.4, the PP/ST author **should** select whether the authorized user and/or trusted
8092 subjects **can** determine the real user name.

8093 **I.3.3.2.5 Assignment**

8094 In FPR_PSE.2.4, the PP/ST author **should** identify the list of conditions under which the trusted
8095 subjects and authorized user **can** determine the real user name based on the provided
8096 reference. These conditions **can** be conditions such as time of day, or they **can** be administrative
8097 such as on a court order.

8098 In FPR_PSE.2.4, the PP/ST author **should** identify the list of trusted subjects that **can** obtain the
8099 real user name under a specified condition.

EXAMPLE

a notary or special authorized user.

8100 **I.3.4 FPR_PSE.3 Alias pseudonymity**

8101 **I.3.4.1 User application notes**

8102 In this component, the TSF **shall** ensure that the provided reference meets certain construction
8103 rules, and thereby **can** be used in a secure way by potentially insecure subjects.

8104 If a user wants to use disk resources without disclosing its identity, pseudonymity **can** be used.
8105 However, every time the user accesses the system, the same alias must be used. Such conditions
8106 **can** be specified in this component.

8107 **I.3.4.2 Operations**

8108 **I.3.4.2.1 Assignment**

8109 In FPR_PSE.3.1, the PP/ST author **should** specify the set of users and/or subjects against which
 8110 the TSF must provide protection. For example, even if the PP/ST author specifies a single user
 8111 or subject role, the TSF must not only provide protection against each individual user or subject
 8112 but must protect with respect to cooperating users and/or subjects.

EXAMPLE

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).

8113 In FPR_PSE.3.1, the PP/ST author **should** identify the list of subjects and/or operations and/or
 8114 objects where the real user name of the subject **should** be protected.

EXAMPLE

“the accessing of job offers”.

8115

8116 NOTE “objects” includes any other attributes which might enable another user or subject to derive the actual
 8117 identity of the user.

8118 In FPR_PSE.3.2, the PP/ST author **should** identify the (one or more) number of aliases the TSF is
 8119 able to provide.

8120 In FPR_PSE.3.2, the PP/ST author **should** identify the list of subjects to whom the TSF is able to
 8121 provide an alias.

8122 I.3.4.2.2 Selection

8123 In FPR_PSE.3.3, the PP/ST author **should** specify whether the user alias is generated by the TSF,
 8124 or supplied by the user. Only one of these options **may** be chosen.

8125 I.3.4.2.3 Assignment

8126 In FPR_PSE.3.3, the PP/ST author **should** identify the metric to which the TSF-generated or
 8127 user-generated alias **should** conform.

8128 In FPR_PSE.3.4, the PP/ST author **should** identify the list of conditions that indicate when the
 8129 used reference for the real user name **shall** be identical and when it **shall** be different, for
 8130 example, “when the user logs on to the same host” it will use a unique alias.

8131 I.4 Unlinkability (FPR_UNL)

8132 I.4.1 User notes

8133 Unlinkability ensures that a user **may** make multiple uses of resources or services without
 8134 others being able to link these uses together. Unlinkability differs from pseudonymity that,
 8135 although in pseudonymity the user is also not known, relations between different actions **can** be
 8136 provided.

8137 The requirements for unlinkability are intended to protect the user identity against the use of
 8138 profiling of the operations.

EXAMPLE 1

For example, when a telephone smart card is employed with a unique number, the telephone company can determine the behaviour of the user of this telephone card. When a telephone profile of the users is known, the card can be linked to a specific user.

8139 Hiding the relationship between different invocations of a service or access of a resource will
 8140 prevent this kind of information gathering.

8141 As a result, a requirement for unlinkability **could** imply that the subject and user identity of an
 8142 operation must be protected. Otherwise this information might be used to link operations
 8143 together.

8144 Unlinkability requires that different operations cannot be related. This relationship **can** take
8145 several forms.

EXAMPLE 2

The user associated with the operation, or the terminal which initiated the action, or the time the action was executed.

8146 The PP/ST author **can** specify what kind of relationships are present that must be countered.

8147 Possible applications include the ability to make multiple use of a pseudonym without creating
8148 a usage pattern that might disclose the user's identity.

EXAMPLE 3

Potential hostile subjects and users include providers, system operators, communication partners and users, who smuggle malicious parts, (including malware) into systems, they do not operate but want to get information about. All of these attackers **can** investigate (such as which users used which services) and misuse this information.

8149 Unlinkability protects users from linkages, which **could** be drawn between several actions of a
8150 customer.

EXAMPLE 4

a series of phone calls made by an anonymous customer to different partners, where the combination of the partner's identities might disclose the identity of the customer.

8151 I.4.2 FPR_UNL.1 Unlinkability

8152 I.4.2.1 User application notes

8153 This component ensures that users cannot link different operations in the system and thereby
8154 obtain information.

8155 I.4.2.2 Operations

8156 I.4.2.2.1 Assignment

8157 In FPR_UNL.1.1, the PP/ST author **should** specify the set of users and/or subjects against which
8158 the TSF must provide protection.

EXAMPLE 1

Even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject but must protect with respect to cooperating users and/or subjects.

8159

EXAMPLE 2

A set of users **could** be a group of users which can operate under the same role or can all use the same process(es).

8160 In FPR_UNL.1.1, the PP/ST author **should** identify the list of operations which **should** be
8161 subjected to the unlinkability requirement.

EXAMPLE 3

"sending email".

8162 I.4.2.2.2 Selection

8163 In FPR_UNL.1.1, the PP/ST author **should** select the relationships that **should** be obscured. The
8164 selection allows either the user identity or an assignment of relations to be specified.

8165 I.4.2.2.3 Assignment

8166 In FPR_UNL.1.1, the PP/ST author **should** identify the list of relations which **should** be protected
8167 against.

EXAMPLE

“originate from the same IP address”.

8168 **I.5 Unobservability (FPR_UNO)**8169 **I.5.1 User notes**

8170 Unobservability ensures that a user **may** use a resource or service without others, especially
8171 third parties, being able to observe that the resource or service is being used.

8172 Unobservability approaches the user identity from a different direction than the previous
8173 families Anonymity, Pseudonymity and Unlinkability. In this case, the intent is to hide the use of
8174 a resource or service, rather than to hide the user's identity.

8175 A number of techniques **can** be applied to implement unobservability.

EXAMPLE

Examples of techniques to provide unobservability are:

- a) Allocation of information impacting unobservability: Unobservability relevant information (such as information that describes that an operation occurred) can be allocated in several locations within the TOE. The information might be allocated to a single randomly chosen part of the TOE such that an attacker does not know which part of the TOE should be attacked. An alternative system might distribute the information such that no single part of the TOE has sufficient information that, if circumvented, the privacy of the user would be compromised. This technique is explicitly addressed in FPR_UNO.2 Allocation of information impacting unobservability.
- b) Broadcast: When information is broadcast (such as Internet and Radio frequencies, including Ethernet, Bluetooth, WiFi and Near-field communication bands), users cannot determine who actually received and used that information. This technique is especially useful when information should reach receivers which have to fear a stigma for being interested in that information (such as sensitive medical information).
- c) Cryptographic protection and message padding: People observing a message stream might obtain information from the fact that a message is transferred and from attributes on that message. By traffic padding, message padding and encrypting the message stream, the transmission of a message and its attributes can be protected.

8176 Sometimes, users **should** not see the use of a resource, but an authorized user must be allowed
8177 to see the use of the resource in order to perform his duties. In such cases, the FPR_UNO.4
8178 Authorized user observability **could** be used, which provides the capability for one or more
8179 authorized users to see the usage.

8180 This family makes use of the concept “parts of the TOE”. This is considered any part of the TOE
8181 that is either physically or logically separated from other parts of the TOE.

8182 Unobservability of communications **may** be an important factor in many areas, such as the
8183 enforcement of constitutional rights, organizational policies, or in defense related applications.

8184 **I.5.2 FPR_UNO.1 Unobservability**8185 **I.5.2.1 User application notes**

8186 This component requires that the use of a function or resource cannot be observed by
8187 unauthorized users.

8188 **I.5.2.2 Operations**8189 **I.5.2.2.1 Assignment**

8190 In FPR_UNO.1.1, the PP/ST author **should** specify the list of users and/or subjects against which
8191 the TSF must provide protection.

EXAMPLE 1

Even if the PP/ST author specifies a single user or subject role, the TSF must not only provide protection against each individual user or subject but must protect with respect to cooperating users and/or subjects.

8192

EXAMPLE 2

A set of users **could** be a group of users which **can** operate under the same role or **can** all use the same process(es).

8193 In FPR_UNO.1.1, the PP/ST author **should** identify the list of operations that are subjected to the
8194 unobservability requirement. Other users/subjects will then not be able to observe the
8195 operations on a covered object in the specified list.

EXAMPLE 3

reading and writing to the object.

8196 In FPR_UNO.1.1, the PP/ST author **should** identify the list of objects which are covered by the
8197 unobservability requirement.

EXAMPLE 4

a specific mail server or ftp site.

8198 In FPR_UNO.1.1, the PP/ST author **should** specify the set of protected users and/or subjects
8199 whose unobservability information will be protected.

EXAMPLE 5

“Users accessing the system through the internet”.

8200 **I.5.3 FPR_UNO.2 Allocation of information impacting unobservability**

8201 **I.5.3.1 User application notes**

8202 This component requires that the use of a function or resource cannot be observed by specified
8203 users or subjects. Furthermore, this component specifies that information related to the privacy
8204 of the user is distributed within the TOE such that attackers might not know which part of the
8205 TOE to target, or they need to attack multiple parts of the TOE.

8206 An example of the use of this component is the use of a randomly allocated node to provide a
8207 function. In such a case the component might require that the privacy related information **shall**
8208 only be available to one identified part of the TOE and will not be communicated outside this
8209 part of the TOE.

EXAMPLE

A more complex example can be found in some “voting algorithms”. Several parts of the TOE will be involved in the service, but no individual part of the TOE will be able to violate the policy. So, a person may cast a vote (or not) without the TOE being able to determine whether a vote has been cast and what the vote happened to be (unless the vote was unanimous).

8210 **I.5.3.2 Operations**

8211 **I.5.3.2.1 Assignment**

8212 In FPR_UNO.2.1, the PP/ST author **should** specify the list of users and/or subjects against which
8213 the TSF must provide protection. For example, even if the PP/ST author specifies a single user
8214 or subject role, the TSF must not only provide protection against each individual user or subject
8215 but must protect with respect to cooperating users and/or subjects.

EXAMPLE 1

A set of users **could** be a group of users which **can** operate under the same role or **can** all use the same process(es).

8216 In FPR_UNO.2.1, the PP/ST author **should** identify the list of operations that are subjected to the
8217 unobservability requirement. Other users/subjects will then not be able to observe the
8218 operations on a covered object in the specified list

EXAMPLE 2

Reading and writing to the object.

8219 In FPR_UNO.2.1, the PP/ST author **should** identify the list of objects which are covered by the
8220 unobservability requirement. An example **could** be a specific mail server or ftp site.

8221 In FPR_UNO.2.1, the PP/ST author **should** specify the set of protected users and/or subjects
8222 whose unobservability information will be protected.

EXAMPLE 3

“users accessing the system through the internet”.

8223 In FPR_UNO.2.2, the PP/ST author **should** identify which privacy related information **should** be
8224 distributed in a controlled manner.

EXAMPLE 4

This information **could** include: IP address of subject, IP address of object, time, used encryption keys.

8225 In FPR_UNO.2.2, the PP/ST author **should** specify the conditions to which the dissemination of
8226 the information **should** adhere. These conditions **should** be maintained throughout the lifetime
8227 of the privacy related information of each instance.

EXAMPLE 5

Examples of these conditions **could** be:

- “the information shall only be present at a single separated part of the TOE and shall not be communicated outside this part of the TOE.”,
- “the information shall only reside in a single separated part of the TOE, but shall be moved to another part of the TOE periodically”;
- “the information shall be distributed between the different parts of the TOE such that compromise of any 5 separated parts of the TOE will not compromise the security policy”.

8228 **I.5.4 FPR_UNO.3 Unobservability without soliciting information**

8229 **I.5.4.1 User application notes**

8230 This component is used to require that the TSF does not try to obtain information that might
8231 compromise unobservability when provided specific services. Therefore, the TSF will not solicit
8232 (i.e. try to obtain from other entities) any information that might be used to compromise
8233 unobservability.

8234 **I.5.4.2 Operations**

8235 **I.5.4.2.1 Assignment**

8236 In FPR_UNO.3.1, the PP/ST author **should** identify the list of services which are subject to the
8237 unobservability requirement.

EXAMPLE 1

“the accessing of job descriptions”.

8238 In FPR_UNO.3.1, the PP/ST author **should** identify the list of subjects from which privacy related
8239 information **should** be protected when the specified services are provided.

8240 In FPR_UNO.3.1, the PP/ST author **should** specify the privacy related information that will be
8241 protected from the specified subjects.

EXAMPLE 2

Examples include the identity of the subject that used a service and the quantity of a service that has been used such as memory resource utilization.

8242 **I.5.5 FPR_UNO.4 Authorized user observability**

8243 **I.5.5.1 User application notes**

8244 This component is used to require that there will be one or more authorized users with the
8245 rights to view the resource utilization. Without this component, this review is allowed, but not
8246 mandated.

8247 **I.5.5.2 Operations**

8248 **I.5.5.2.1 Assignment**

8249 In FPR_UNO.4.1, the PP/ST author **should** specify the set of authorized users for which the TSF
8250 must provide the capability to observe the resource utilization. A set of authorized users, for
8251 example, **could** be a group of authorized users which **can** operate under the same role or **can** all
8252 use the same process(es).

8253 In FPR_UNO.4.1, the PP/ST author **should** specify the set of resources and/or services that the
8254 authorized user must be able to observe.

Annex J (normative)

8255
8256
8257
8258

Class FPT: Protection of the TSF- application notes

8259 J.1 General information

8260 This class contains families of functional requirements that relate to the integrity and
8261 management of the mechanisms that constitute the TSF and to the integrity of TSF data. In some
8262 sense, families in this class **may** appear to duplicate components in the FDP: User data
8263 protection class; they **may** even be implemented using the same mechanisms. However, FDP:
8264 User data protection focuses on user data protection, while FPT: Protection of the TSF focuses
8265 on TSF data protection. In fact, components from the FPT: Protection of the TSF class are
8266 necessary to provide requirements that the SFPs in the TOE cannot be tampered with or
8267 bypassed.

8268 From the point of view of this class, regarding to the TSF there are three significant elements:

- 8269 a) The TSF's implementation, which executes and implements the mechanisms that
8270 enforce the SFRs.
- 8271 b) The TSF's data, which are the administrative databases that guide the enforcement
8272 of the SFRs.
- 8273 c) The external entities that the TSF **may** interact with in order to enforce the SFRs.

8274 All of the families in the FPT: Protection of the TSF class **can** be related to these areas, and fall
8275 into the following groupings:

- 8276 a) TOE emanation (FPT_EMS), which addresses potential leakage of information from
8277 the TOE via emanations.
- 8278 b) Trusted recovery (FPT_RCV), Fail secure (FPT_FLS), and Internal TOE TSF data
8279 replication consistency (FPT_TRC), which address the behaviour of the TSF when
8280 failure occurs and immediately after.
- 8281 c) TSF initialization (FPT_INI), which addresses the initialization of the TOE into a
8282 correct and secure operational state.
- 8283 d) Internal TOE TSF data transfer (FPT_ITT), which addresses protection of TSF data
8284 when it is transmitted between physically-separated parts of the TOE.
- 8285 e) TSF physical protection (FPT_PHP), which provides an authorized user with the
8286 ability to detect external attacks on the parts of the TOE that comprise the TSF.
- 8287 f) Availability of exported TSF data (FPT_ITA), Confidentiality of exported TSF data
8288 (FPT_ITC), Integrity of exported TSF data (FPT_ITI), which address the protection
8289 and availability of TSF data between the TSF and another trusted IT product.
- 8290 g) Replay detection (FPT_RPL), which addresses the replay of various types of
8291 information and/or operations.
- 8292 h) State synchrony protocol (FPT_SSP), which addresses the synchronization of states,
8293 based upon TSF data, between different parts of a distributed TSF.
- 8294 i) Time stamps (FPT_STM), which addresses reliable timing.
- 8295 j) Inter-TSF TSF data consistency (FPT_TDC), which addresses the consistency of TSF
8296 data shared between the TSF and another trusted IT product.
- 8297 k) Testing of external entities (FPT_TEE) and TSF self-test (FPT_TST), which provide
8298 an authorized user with the ability to verify the correct operation of the external

8299 entities interacting with the TSF to enforce the SFRs, and the integrity of the TSF
 8300 data and TSF itself.

8301 **J.2 FPT_EMS TOE emanation**

8302 **J.2.1 User notes**

8303 This family defines the requirements for the TOE to be able to prevent or mitigate attacks
 8304 against data stored in and used by the TOE where the attack is based on external observable
 8305 physical phenomena of the TOE.

EXAMPLE

Examples of such attacks are analysis of TOE's electromagnetic radiation, simple power analysis (SPA), differential power analysis (DPA), timing attacks, etc..

8306 FPT_EMS.1.1 Limit of Emissions requires the TOE to not emit intelligible emissions enabling
 8307 access to TSF data or user data.

8308 **J.2.2 FPT_EMS.1 TOE emanation**

8309 **J.2.3 User application notes**

8310 Table 2 - FPT_EMS.1.1 Table found as part of the FPT_EMS.1.1 Limit of Emissions element shall
 8311 be completed by the PP/ST author. Each row, which can be identified using the "Identifier",
 8312 provides a set of assignments for completing the SFR, allowing the PP/ST author to specify the
 8313 requirements for TOE emanation protection for various different combinations of emissions,
 8314 interfaces, TSF data and user data.

8315 **J.2.3.1 Operations**

8316 **J.2.3.1.1 Assignment**

EXAMPLE

Types of emission can include audio frequencies and radio frequencies.

Types of interfaces can include physical ports, I.C. boundaries, and electronic components.

8317 **J.3 Fail secure (FPT_FLS)**

8318 **J.3.1 User notes**

8319 The requirements of this family ensure that the TOE will always enforce its SFRs in the event of
 8320 certain types of failures in the TSF.

8321 **J.3.2 FPT_FLS.1 Failure with preservation of secure state**

8322 **J.3.2.1 User application notes**

8323 The term "secure state" refers to a state in which the TSF data are consistent and the TSF
 8324 continues correct enforcement of the SFRs.

8325 Although it is desirable to audit situations in which failure with preservation of secure state
 8326 occurs, it is not possible in all situations. The PP/ST author **should** specify those situations in
 8327 which audit is desired and feasible.

8328 Failures in the TSF **may** include "hard" failures, which indicate an equipment malfunction and
 8329 which **may** require maintenance, service, or repair of the TSF. Failures in the TSF **may** also
 8330 include recoverable "soft" failures, which **may** only require initialization or resetting of the TSF.

8331 **J.3.2.2 Operations**

8332 **J.3.2.2.1 Assignment**

8333 In FPT_FLS.1.1, the PP/ST author **should** list the types of failures in the TSF for which the TSF
 8334 **should** “fail secure,” that is, **should** preserve a secure state and continue to correctly enforce the
 8335 SFRs.

8336 **J.4 TSF Initialization (FPT_INI)**

8337 **Editors' note.**

8338 **The CCUF have been asked to contribute the application notes for FPT_INI during the commenting period**
 8339 **on this draft.**

8340 **J.4.1 User notes**

8341 **J.4.2 FPT_INI.1 XXX**

8342 **J.4.3 User application notes**

8343 **J.4.3.1 Operations**

8344 **J.4.3.1.1 Assignment**

8345 **J.5 Availability of exported TSF data (FPT_ITA)**

8346 **J.5.1 User notes**

8347 This family defines the rules for the prevention of loss of availability of TSF data moving
 8348 between the TSF and another trusted IT product. This data **could** be TSF critical data such as
 8349 passwords, keys, audit data, or TSF executable code.

8350 This family is used in a distributed context where the TSF is providing TSF data to another
 8351 trusted IT product. The TSF **can** only take the measures at its site and cannot be held
 8352 responsible for the TSF at the other trusted IT product.

8353 If there are different availability metrics for different types of TSF data, then this component
 8354 **should** be iterated for each unique pairing of metrics and types of TSF data.

8355 **J.5.2 FPT_ITA.1 Inter-TSF availability within a defined availability metric**

8356 **J.5.2.1 Operations**

8357 **J.5.2.1.1 Assignment**

8358 In FPT_ITA.1.1, the PP/ST author **should** specify the types of TSF data that are subject to the
 8359 availability metric.

8360 In FPT_ITA.1.1, the PP/ST **should** specify the availability metric for the applicable TSF data.

8361 In FPT_ITA.1.1, the PP/ST author **should** specify the conditions under which availability must be
 8362 ensured.

EXAMPLE

There must be a connection between the TOE and another trusted IT product.

8363 **J.6 Confidentiality of exported TSF data (FPT_ITC)**

8364 **J.6.1 User notes**

8365 This family defines the rules for the protection from unauthorized disclosure of TSF data
 8366 moving between the TSF and another trusted IT product.

EXAMPLE

Examples of this data are TSF critical data such as passwords, keys, audit data, or TSF executable code.

8367 This family is used in a distributed context where the TSF is providing TSF data to another
 8368 trusted IT product. The TSF **can** only take the measures at its site and cannot be held
 8369 responsible for the behaviour of the other trusted IT product.

8370 **J.6.2 FPT_ITC.1 Inter-TSF confidentiality during transmission**

8371 **J.6.2.1 Evaluator notes**

8372 Confidentiality of TSF Data during transmission is necessary to protect such information from
 8373 disclosure.

EXAMPLE

Some possible implementations that **could** provide confidentiality include the use of cryptographic algorithms as well as spread spectrum techniques.

8374 **J.7 Integrity of exported TSF data (FPT_ITI)**

8375 **J.7.1 User notes**

8376 This family defines the rules for the protection, from unauthorized modification, of TSF data
 8377 during transmission between the TSF and another trusted IT product.

EXAMPLE

Examples of this data are TSF critical data such as passwords, keys, audit data, or TSF executable code.

8378 This family is used in a distributed context where the TSF is exchanging TSF data with another
 8379 trusted IT product. Note that a requirement that addresses modification, detection, or recovery
 8380 at another trusted IT product cannot be specified, as the mechanisms that another trusted IT
 8381 product will use to protect its data cannot be determined in advance. For this reason, these
 8382 requirements are expressed in terms of the “TSF providing a capability” which another trusted
 8383 IT product **can** use.

8384 **J.7.2 FPT_ITI.1 Inter-TSF detection of modification**

8385 **J.7.2.1 User application notes**

8386 This component **should** be used in situations where it is sufficient to detect when data have
 8387 been modified. An example of such a situation is one in which another trusted IT product **can**
 8388 request the TOE's TSF to retransmit data when modification has been detected or respond to
 8389 such types of request.

8390 The desired strength of modification detection is based upon a specified modification metric
 8391 that is a function of the algorithm used, which **may** range from a weak checksum and parity
 8392 mechanisms that **may** fail to detect multiple bit changes, to more complicated cryptographic
 8393 checksum approaches.

8394 **J.7.2.2 Operations**

8395 **J.7.2.2.1 Assignment**

8396 In FPT_ITI.1.1, the PP/ST **should** specify the modification metric that the detection mechanism
 8397 must satisfy. This modification metric **shall** specify the desired strength of the modification
 8398 detection.

8399 In FPT_ITI.1.2, the PP/ST **should** specify the actions to be taken if a modification of TSF data has
 8400 been detected. An example of an action is: “ignore the TSF data and request the originating
 8401 trusted product to send the TSF data again”.

8402 **J.7.3 FPT_ITI.2 Inter-TSF detection and correction of modification**

8403 **J.7.3.1 User application notes**

8404 This component **should** be used in situations where it is necessary to detect or correct
8405 modifications of TSF critical data.

8406 The desired strength of modification detection is based upon a specified modification metric
8407 that is a function of the algorithm used, which **may** range from a checksum and parity
8408 mechanisms that **may** fail to detect multiple bit changes, to more complicated cryptographic
8409 checksum approaches. The metric that needs to be defined **can** either refer to the attacks it will
8410 resist or to mechanisms that are well known in the public literature.

EXAMPLE

Attack reference: "only 1 in a 1000 random messages will be accepted".

Well known mechanism: "the strength must be conformant to the strength offered by Secure Hash Algorithm".

8411

8412 The approach taken to correct modification might be done through some form of error
8413 correcting checksum.

8414 **J.7.3.2 Evaluator notes**

8415 Some possible means of satisfying this requirement involves the use of cryptographic functions
8416 or some form of checksum.

8417 **J.7.3.3 Operations**

8418 **J.7.3.3.1 Assignment**

8419 In FPT_ITI.2.1, the PP/ST **should** specify the modification metric that the detection mechanism
8420 must satisfy. This modification metric **shall** specify the desired strength of the modification
8421 detection.

8422 In FPT_ITI.2.2, the PP/ST **should** specify the actions to be taken if a modification of TSF data has
8423 been detected.

EXAMPLE

An example of an action is: "ignore the TSF data and request the originating trusted product to send the TSF data again".

8424

8425 In FPT_ITI.2.3, the PP/ST author **should** define the types of modification from which the TSF
8426 **should** be capable of recovering.

8427 **J.8 Internal TOE TSF data transfer (FPT_ITT)**

8428 **J.8.1 User notes**

8429 This family provides requirements that address protection of TSF data when it is transferred
8430 between separate parts of a TOE across an internal channel.

8431 The determination of the degree of separation (i.e., physical, or logical) that would make
8432 application of this family useful depends on the intended environment of use. In a hostile
8433 environment, there **may** be risks arising from transfers between parts of the TOE separated by
8434 only a system bus or an inter-process communications channel. In more benign environments,
8435 the transfers **may** be across more traditional network media.

8436 **J.8.2 Evaluator notes**

8437 One practical mechanism available to a TSF to provide this protection is cryptographically-
8438 based.

8439 **J.8.3 FPT_ITT.1 Basic internal TSF data transfer protection**

8440 **J.8.3.1 Operations**

8441 **J.8.3.1.1 Selection**

8442 In FPT_ITT.1.1, the PP/ST author **should** specify the desired type of protection to be provided
8443 from the choices: disclosure, modification.

8444 **J.8.4 FPT_ITT.2 TSF data transfer separation**

8445 **J.8.4.1 User application notes**

8446 One of the ways to achieve separation of TSF data based on SFP-relevant attributes is through
8447 the use of separate logical or physical channels.

8448 **J.8.4.2 Operations**

8449 **J.8.4.2.1 Selection**

8450 In FPT_ITT.2.1, the PP/ST author **should** specify the desired type of protection to be provided
8451 from the choices: disclosure, modification.

8452 **J.8.5 FPT_ITT.3 TSF data integrity monitoring**

8453 **J.8.5.1 Operations**

8454 **J.8.5.1.1 Selection**

8455 In FPT_ITT.3.1, the PP/ST author **should** specify the desired type of modification that the TSF
8456 **shall** be able to detect. The PP/ST author **should** select from: modification of data, substitution
8457 of data, re-ordering of data, deletion of data, or any other integrity errors.

8458 **J.8.5.1.2 Assignment**

8459 In FPT_ITT.3.1, if the PP/ST author chooses the latter selection noted in the preceding
8460 paragraph, then the author **should** also specify what those other integrity errors are that the
8461 TSF **should** be capable of detecting.

8462 In FPT_ITT.3.2, the PP/ST author **should** specify the action to be taken when an integrity error
8463 is identified.

8464 **J.9 TSF physical protection (FPT_PHP)**

8465 **J.9.1 User notes**

8466 TSF physical protection components refer to restrictions on unauthorized physical access to the
8467 TSF, and to the deterrence of, and resistance to, unauthorized physical modification, or
8468 substitution of the TSF.

8469 The requirements in this family ensure that the TSF is protected from physical tampering and
8470 interference. Satisfying the requirements of these components results in the TSF being
8471 packaged and used in such a manner that physical tampering is detectable, or resistance to
8472 physical tampering is measurable based on defined work factors. Without these components,
8473 the protection functions of a TSF lose their effectiveness in environments where physical
8474 damage cannot be prevented. This component also provides requirements regarding how the
8475 TSF must respond to physical tampering attempts.

EXAMPLE 1

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

8476 It is acceptable for the functions that are available to an authorized user for detecting physical
8477 tampering to be available only in an off-line or maintenance mode. Controls **should** be in place
8478 to limit access during such modes to authorized users. As the TSF **may** not be “operational”

8479 during those modes, it **may** not be able to provide normal enforcement for authorized user
 8480 access. The physical implementation of a TOE might consist of several structures. This set of
 8481 “elements” as a whole must protect (protect, notify and resist) the TSF from physical tampering.
 8482 This does not mean that all devices must provide these features, but the complete physical
 8483 construct as a whole **should**.

EXAMPLE 2

Examples of structures include an outer shielding, cards, and chips.

8484 Although there is only minimal auditing associating with these components, this is solely
 8485 because there is the potential that the detection and alarm mechanisms **may** be implemented
 8486 completely in hardware, below the level of interaction with an audit subsystem. Nevertheless, a
 8487 PP/ST author **may** determine that for a particular anticipated threat environment, there is a
 8488 need to audit physical tampering. If this is the case, the PP/ST author **should** include
 8489 appropriate requirements in the list of audit events.

8490 NOTE inclusion of these requirements **may** have implications on the hardware design and its interface to the
 8491 software.

EXAMPLE 3

Examples of a hardware-based detection system is one based on breaking a circuit and lighting a light emitting diode (LED) if the circuit is broken when a button is pressed by the authorized user.

8492 **J.9.2 FPT_PHP.1 Passive detection of physical attack**

8493 **J.9.2.1 User application notes**

8494 FPT_PHP.1 Passive detection of physical attack **should** be used when threats from unauthorized
 8495 physical tampering with parts of the TOE are not countered by procedural methods. It
 8496 addresses the threat of undetected physical tampering with the TSF. Typically, an authorized
 8497 user would be given the function to verify whether tampering took place. As written, this
 8498 component simply provides a TSF capability to detect tampering. Specification of management
 8499 functions in FMT_LIM.1 **should** be considered to specify who **can** make use of that capability,
 8500 and how they **can** make use of that capability. If this is done by non-IT mechanisms such as
 8501 physical inspection. management functions are not required.

8502 **J.9.3 FPT_PHP.2 Notification of physical attack**

8503 **J.9.3.1 User application notes**

8504 FPT_PHP.2 Notification of physical attack **should** be used when threats from unauthorized
 8505 physical tampering with parts of the TOE are not countered by procedural methods, and it is
 8506 required that designated individuals be notified of physical tampering. It addresses the threat
 8507 that physical tampering with TSF elements, although detected, **may** not be noticed. Specification
 8508 of management functions in FMT_MOF.1 Management of security functions behaviour **should** be
 8509 considered to specify who **can** make use of that capability, and how they **can** make use of that
 8510 capability.

8511 **J.9.3.2 Operations**

8512 **J.9.3.2.1 Assignment**

8513 In FPT_PHP.2.3, the PP/ST author **should** provide a list of TSF devices/elements for which
 8514 active detection of physical tampering is required.

8515 In FPT_PHP.2.3, the PP/ST author **should** designate a user or role that is to be notified when
 8516 tampering is detected. The type of user or role **may** vary depending on the particular security
 8517 administration component (from the FMT_LIM.1 family) included in the PP/ST.

8518 **J.9.4 FPT_PHP.3 Resistance to physical attack**

8519 **J.9.4.1 User application notes**

8520 For some forms of tampering, it is necessary that the TSF not only detects the tampering, but
8521 actually resists it or delays the attacker.

8522 This component **should** be used when TSF devices and TSF elements are expected to operate in
8523 an environment where a physical tampering of the internals of a TSF device or TSF element
8524 itself is a threat.

EXAMPLE

Physical tampering includes observation, analysis, or modification.

8525 **J.9.4.2 Operations**

8526 **J.9.4.2.1 Assignment**

8527 In FPT_PHP.3.1, the PP/ST author **should** specify tampering scenarios to a list of TSF
8528 devices/elements for which the TSF **should** resist physical tampering. This list **may** be applied
8529 to a defined subset of the TSF physical devices and elements based on considerations such as
8530 technology limitations and relative physical exposure of the device. Such sub setting **should** be
8531 clearly defined and justified. Furthermore, the TSF **should** automatically respond to physical
8532 tampering. The automatic response **should** be such that the policy of the device is preserved.

EXAMPLE

An example of policy protection:

with a confidentiality policy, it would be acceptable to physically disable the device so that
the protected information **may** not be retrieved.

8533

8534 In FPT_PHP.3.1, the PP/ST author **should** specify the list of TSF devices/elements for which the
8535 TSF **should** resist physical tampering in the scenarios that have been identified.

8536 **J.10 Trusted recovery (FPT_RCV)**

8537 **J.10.1 User notes**

8538 The requirements of this family ensure that the TSF **can** determine that the TOE is started-up
8539 without protection compromise and **can** recover without protection compromise after
8540 discontinuity of operations. This family is important because the start-up state of the TSF
8541 determines the protection of subsequent states.

8542 Recovery components reconstruct the TSF secure states, or prevent transitions to insecure
8543 states, as a direct response to occurrences of expected failures, discontinuity of operation or
8544 start-up.

EXAMPLE

Failures that must be generally anticipated include the following:

- a) Unmaskable action failures that always result in a system crash (such as persistent inconsistency of critical system tables, uncontrolled transfers within the TSF code caused by transient failures of hardware or firmware, power failures, processor failures, communication failures).
- b) Media failures causing part or all of the media representing the TSF objects to become inaccessible or corrupt (such as parity errors, disk head crash, persistent read/write failure caused by misaligned disk heads, worn-out magnetic coating, dust on the disk surface, loss of Internet connection).
- c) Discontinuity of operation caused by erroneous administrative action or lack of timely administrative action (such as unexpected shutdowns by turning off power, ignoring the exhaustion of critical resources, inadequate installed configuration).

8545

8546 NOTE Recovery may be from either a complete or partial failure scenario. Although a complete failure might
 8547 occur in a monolithic operating system, it is less likely to occur in a distributed environment. In such environments,
 8548 subsystems may fail, but other portions remain operational. Further, critical components may be redundant (disk
 8549 mirroring, alternative routes), and checkpoints may be available. Thus, recovery is expressed in terms of recovery to
 8550 a secure state.

8551 There are different interactions between Trusted recovery (FPT_RCV) and TSF self-test
 8552 (FPT_TST) components to be considered when selecting Trusted recovery (FPT_RCV):

- 8553 a) The need for trusted recovery **may** be indicated through the results of TSF self-
 8554 testing, where the results of the self-tests indicate that the TSF is in an insecure
 8555 state and return to a secure state or entrance in maintenance mode is required.
- 8556 b) A failure, as discussed above, **may** be identified by an administrator. Either the
 8557 administrator **may** perform the actions to return the TOE to a secure state and then
 8558 invoke TSF self-tests to confirm that the secure state has been achieved. Or, the TSF
 8559 self-tests **may** be invoked to complete the recovery process.
- 8560 c) A combination of a. and b. above, where the need for trusted recovery is indicated
 8561 through the results of TSF self-testing, the administrator performs the actions to
 8562 return the TOE to a secure state and then invokes TSF self-tests to confirm that the
 8563 secure state has been achieved.
- 8564 d) Self-tests detect a failure/service discontinuity, then either automated recovery or
 8565 entrance to a maintenance mode.

8566 This family identifies a maintenance mode. In this maintenance mode, normal operation might
 8567 be impossible or severely restricted, as otherwise insecure situations might occur. Typically,
 8568 only authorized users **should** be allowed access to this mode but the real details of who **can**
 8569 access this mode is a function of FMT: Security management. If FMT: Security management does
 8570 not put any controls on who **can** access this mode, then it **may** be acceptable to allow any user
 8571 to restore the system if the TOE enters such a state. However, in practice, this is probably not
 8572 desirable as the user restoring the system has an opportunity to configure the TOE in such a
 8573 way as to violate the SFRs.

8574 Mechanisms designed to detect exceptional conditions during operation fall under TSF self-test
 8575 (FPT_TST), Fail secure (FPT_FLS), and other areas that address the concept of “Software Safety.”
 8576 It is likely that the use of one of these families will be required to support the adoption of
 8577 Trusted recovery (FPT_RCV). This is to ensure that the TOE will be able to detect when recovery
 8578 is required.

8579 Throughout this family, the phrase “secure state” is used. This refers to some state in which the
 8580 TOE has consistent TSF data and a TSF that **can** correctly enforce the policy. This state **may** be
 8581 the initial “boot” of a clean system, or it might be some checkpointed state.

8582 Following recovery, it **may** be necessary to confirm that the secure state has been achieved
 8583 through self-testing of the TSF. However, if the recovery is performed in a manner such that
 8584 only a secure state **can** be achieved, else recovery fails, then the dependency to the FPT_TST.1
 8585 TSF self-testing component **may** be argued away.

8586 **J.10.2 FPT_RCV.1 Manual recovery**

8587 **J.10.2.1 User application notes**

8588 In the hierarchy of the trusted recovery family, recovery that requires only manual intervention
 8589 is the least desirable, for it precludes the use of the system in an unattended fashion.

8590 This component is intended for use in TOEs that do not require unattended recovery to a secure
 8591 state. The requirements of this component reduce the threat of protection compromise
 8592 resulting from an attended TOE returning to an insecure state after recovery from a failure or
 8593 other discontinuity.

8594 **J.10.2.2 Evaluator notes**

8595 It is acceptable for the functions that are available to an authorized user for trusted recovery to
8596 be available only in a maintenance mode. Controls **should** be in place to limit access during
8597 maintenance to authorized users.

8598 **J.10.2.3 Operations**

8599 **J.10.2.3.1 Assignment**

8600 In FPT_RCV.1.1, the PP/ST author **should** specify the list of failures or service discontinuities
8601 following which the TOE will enter a maintenance mode.

EXAMPLE

power failure, audit storage exhaustion, any failure or discontinuity.

8602 **J.10.3 FPT_RCV.2 Automated recovery**

8603 **J.10.3.1 User application notes**

8604 Automated recovery is considered to be more useful than manual recovery, as it allows the
8605 machine to operate in an unattended fashion.

8606 The component FPT_RCV.2 Automated recovery extends the feature coverage of FPT_RCV.1
8607 Manual recovery by requiring that there be at least one automated method of recovery from
8608 failure or service discontinuity. It addresses the threat of protection compromise resulting from
8609 an unattended TOE returning to an insecure state after recovery from a failure or other
8610 discontinuity.

8611 **J.10.3.2 Evaluator notes**

8612 It is acceptable for the functions that are available to an authorized user for trusted recovery to
8613 be available only in a maintenance mode. Controls **should** be in place to limit access during
8614 maintenance to authorized users.

8615 For FPT_RCV.2.1, it is the responsibility of the developer of the TSF to determine the set of
8616 recoverable failures and service discontinuities.

8617 It is assumed that the robustness of the automated recovery mechanisms will be verified.

8618 **J.10.3.3 Operations**

8619 **J.10.3.3.1 Assignment**

8620 In FPT_RCV.2.1, the PP/ST author **should** specify the list of failures or service discontinuities
8621 following which the TOE will need to enter a maintenance mode.

EXAMPLE

power failure, audit storage exhaustion.

8622 In FPT_RCV.2.2, the PP/ST author **should** specify the list of failures or other discontinuities for
8623 which automated recovery must be possible.

8624 **J.10.4 FPT_RCV.3 Automated recovery without undue loss**

8625 **J.10.4.1 User application notes**

8626 Automated recovery is considered to be more useful than manual recovery, but it runs the risk
8627 of losing a substantial number of objects. Preventing undue loss of objects provides additional
8628 utility to the recovery effort.

8629 The component FPT_RCV.3 Automated recovery without undue loss extends the feature
8630 coverage of FPT_RCV.2 Automated recovery by requiring that there not be undue loss of TSF
8631 data or objects under the control of the TSF. At FPT_RCV.2 Automated recovery, the automated
8632 recovery mechanisms **could** conceivably recover by deleting all objects and returning the TSF to
8633 a known secure state. This type of drastic automated recovery is precluded in FPT_RCV.3
8634 Automated recovery without undue loss.

8635 This component addresses the threat of protection compromise resulting from an unattended
8636 TOE returning to an insecure state after recovery from a failure or other discontinuity with a
8637 large loss of TSF data or objects under the control of the TSF.

8638 **J.10.4.2 Evaluator notes**

8639 It is acceptable for the functions that are available to an authorized user for trusted recovery to
8640 be available only in a maintenance mode. Controls **should** be in place to limit access during
8641 maintenance to authorized users.

8642 It is assumed that the evaluators will verify the robustness of the automated recovery
8643 mechanisms.

8644 **J.10.4.3 Operations**

8645 **J.10.4.3.1 Assignment**

8646 In FPT_RCV.3.1, the PP/ST author **should** specify the list of failures or service discontinuities
8647 following which the TOE will need to enter a maintenance mode.

EXAMPLE

power failure, audit storage exhaustion.

8648 In FPT_RCV.3.2, the PP/ST author **should** specify the list of failures or other discontinuities for
8649 which automated recovery must be possible.

8650 In FPT_RCV.3.3, the PP/ST author **should** provide a quantification for the amount of loss of TSF
8651 data or objects that is acceptable.

8652 **J.10.5 FPT_RCV.4 Function recovery**

8653 **J.10.5.1 User application notes**

8654 Function recovery requires that if there **should** be some failure in the TSF, that certain functions
8655 in the TSF **should** either complete successfully or recover to a secure state.

8656 **J.10.5.2 Operations**

8657 **J.10.5.2.1 Assignment**

8658 In FPT_RCV.4.1, the PP/ST author **should** specify a list the functions and failure scenarios. In the
8659 event that any of the identified failure scenarios happen, the functions that have been specified
8660 must either complete successfully or recover to a consistent and secure state.

8661 **J.11 Replay detection (FPT_RPL)**

8662 **J.11.1 User notes**

8663 This family addresses detection of replay for various types of entities and subsequent actions to
8664 correct.

8665 **J.11.2 FPT_RPL.1 Replay detection**

8666 **J.11.2.1 User application notes**

8667 The entities included here are those that can be involved in replay detection.

EXAMPLE

Messages, service requests, service responses, or sessions.

8668 **J.11.2.2 Operations**

8669 **J.11.2.2.1 Assignment**

8670 In FPT_RPL.1.1, the PP/ST author **should** provide a list of identified entities for which detection
8671 of replay **should** be possible.

EXAMPLE

Messages, service requests, service responses, and user sessions.

8672 In FPT_RPL.1.2, the PP/ST author **should** specify the list of actions to be taken by the TSF when
 8673 replay is detected. The potential set of actions that **can** be taken includes: ignoring the replayed
 8674 entity, requesting confirmation of the entity from the identified source, and terminating the
 8675 subject from which the re-played entity originated.

8676 **J.12 State synchrony protocol (FPT_SSP)**

8677 **J.12.1 User notes**

8678 Distributed TOEs **may** give rise to greater complexity than monolithic TOEs through the
 8679 potential for differences in state between parts of the TOE, and through delays in
 8680 communication. In most cases, synchronization of state between distributed functions involves
 8681 an exchange protocol, not a simple action. When malice exists in the distributed environment of
 8682 these protocols, more complex defensive protocols are required.

8683 State synchrony protocol (FPT_SSP) establishes the requirement for certain critical functions of
 8684 the TSF to use a trusted protocol. State synchrony protocol (FPT_SSP) ensures that two
 8685 distributed parts of the TOE, such as hosts, have synchronized their states after a security-
 8686 relevant action.

8687 Some states **may** never be synchronized, or the transaction cost **may** be too high for practical
 8688 use.

EXAMPLE 1

encryption key revocation is an example, where knowing the state after the revocation
 action is initiated **can** never be known. Either the action was taken and acknowledgment
 cannot be sent, or the message was ignored by hostile communication partners and the
 revocation never occurred.

8689
 8690 Indeterminacy is unique to distributed TOEs. Indeterminacy and state synchrony are related,
 8691 and the same solution **may** apply. It is futile to design for indeterminate states; the PP/ST
 8692 author **should** express other requirements in such cases.

EXAMPLE 2

raise an alarm, audit the event.

8693 **J.12.2 FPT_SSP.1 Simple trusted acknowledgement**

8694 **J.12.2.1 User application notes**

8695 In this component, the TSF must supply an acknowledgement to another part of the TSF when
 8696 requested. This acknowledgement **should** indicate that one part of a distributed TOE
 8697 successfully received an unmodified transmission from a different part of the distributed TOE.

8698 **J.12.3 FPT_SSP.2 Mutual trusted acknowledgement**

8699 **J.12.3.1 User application notes**

8700 In this component, in addition to the TSF being able to provide an acknowledgement for the
 8701 receipt of a data transmission, the TSF must comply with a request from another part of the TSF
 8702 for an acknowledgement to the acknowledgement.

EXAMPLE

The local TSF transmits some data to a remote part of the TSF. The remote part of the TSF
 acknowledges the successful receipt of the data and requests that the sending TSF confirm
 that it receives the acknowledgement. This mechanism provides additional confidence that

both parts of the TSF involved in the data transmission know that the transmission completed successfully.

8703 **J.13 Time stamps (FPT_STM)**

8704 **J.13.1 User notes**

8705 This family addresses requirements for a reliable time stamp function within a TOE.

8706 It is the responsibility of the PP/ST author to clarify the meaning of the phrase “reliable time
8707 stamp”, and to indicate where the responsibility lies in determining the acceptance of trust.

8708 **J.13.2 FPT_STM.1 Reliable time stamps**

8709 **J.13.2.1 User application notes**

8710 Some possible uses of this component include providing reliable time stamps for the purposes
8711 of audit as well as for security attribute expiration.

8712 **J.14 Inter-TSF TSF data consistency (FPT_TDC)**

8713 **J.14.1 User notes**

8714 In a distributed or composite environment, a TOE **may** need to exchange TSF data with another
8715 trusted IT Product.

EXAMPLE

the SFP-attributes associated with data, audit information, identification information.

8716 This family defines the requirements for sharing and consistent interpretation of these
8717 attributes between the TSF of the TOE and that of a different trusted IT Product.

8718 The components in this family are intended to provide requirements for automated support for
8719 TSF data consistency when such data is transmitted between the TSF of the TOE and another
8720 trusted IT Product. It is also possible that wholly procedural means **could** be used to produce
8721 security attribute consistency, but they are not provided for here.

8722 This family is different from FDP_ETC and FDP_ITC, as those two families are concerned only
8723 with resolving the security attributes between the TSF and its import/export medium.

8724 If the integrity of the TSF data is of concern, requirements **should** be chosen from the Integrity
8725 of exported TSF data (FPT_ITI) family. These components specify requirements for the TSF to
8726 be able to detect or detect and correct modifications to TSF data in transit.

8727 **J.14.2 FPT_TDC.1 Inter-TSF basic TSF data consistency**

8728 **J.14.2.1 User application notes**

8729 The TSF is responsible for maintaining the consistency of TSF data used by or associated with
8730 the specified function and that are common between two or more trusted systems.

EXAMPLE

The TSF data of two different systems **may** have different conventions internally. For the
TSF data to be used properly (such as to afford the user data the same protection as within
the TOE) by the receiving trusted IT product, the TOE and the other trusted IT product
must use a pre-established protocol to exchange TSF data.

8731 **J.14.2.2 Operations**

8732 **J.14.2.2.1 Assignment**

8733 In FPT_TDC.1.1, the PP/ST author **should** define the list of TSF data types, for which the TSF
8734 **shall** provide the capability to consistently interpret, when shared between the TSF and another
8735 trusted IT product.

8736 In FPT_TDC.1.2, the PP/ST **should** assign the list of interpretation rules to be applied by the TSF.

8737 **J.15 Testing of external entities (FPT_TEE)**

8738 **J.15.1 User notes**

8739 This family defines requirements for the testing of one or more external entities by the TSF.
8740 These external entities are not human users, and they **can** include combinations of software
8741 and/or hardware interacting with the TOE.

EXAMPLE

Examples of the types of tests that **may** be run are:

- a) tests for the presence of a firewall, and possibly whether it is correctly configured;
- b) tests of some of the properties of the operating system that an application TOE runs on;
- c) tests of some of the properties of the IC that a smart card OS TOE runs on (such as the random number generator).

8742

8743 Note The external entity **may** “lie” about the test results, either on purpose or because it is not working
8744 correctly.

8745 These tests **can** be carried out either in some maintenance state, at start-up, on-line, or
8746 continuously. The actions to be taken by the TOE as the result of testing are defined also in this
8747 family.

8748 **J.15.2 Evaluator notes**

8749 The tests of external entities **should** be sufficient to test all of the characteristics of them upon
8750 which the TSF relies.

8751 **J.15.3 FPT_TEE.1 Testing of external entities**

8752 **J.15.3.1 User application notes**

8753 This component is not intended to be applied to human users.

8754 This component provides support for the periodic testing of properties related to external
8755 entities upon which the TSF's operation depends, by requiring the ability to periodically invoke
8756 testing functions.

8757 The PP/ST author **may** refine the requirement to state whether the function **should** be available
8758 in off-line, on-line or maintenance mode.

8759 **J.15.3.2 Evaluator notes**

8760 It is acceptable for the functions for periodic testing to be available only in an off-line or
8761 maintenance mode. Controls **should** be in place to limit access, during maintenance, to
8762 authorized users.

8763 **J.15.3.3 Operations**

8764 **J.15.3.3.1 Selection**

8765 In FPT_TEE.1.1, the PP/ST author **should** specify when the TSF will run the testing of external
8766 entities, during initial start-up, periodically during normal operation, at the request of an
8767 authorized user, or under other conditions. If the tests are run often, then the end users **should**
8768 have more confidence that the TOE is operating correctly than if the tests are run less
8769 frequently. However, this need for confidence that the TOE is operating correctly must be
8770 balanced with the potential impact on the availability of the TOE, as often times, the testing of
8771 external entities **may** delay the normal operation of a TOE.

8772 **J.15.3.3.2 Assignment**

8773 In FPT_TEE.1.1, the PP/ST author **should** specify the properties of the external entities to be
8774 checked by the tests.

EXAMPLE 1

Examples of these properties **may** include configuration or availability properties of a directory server supporting some access control part of the TSF.

8775 In FPT_TEE.1.1, the PP/ST author **should**, if other conditions are selected, specify the frequency
8776 with which the testing of external entities will be run.

EXAMPLE 2

An example of this other frequency or condition **may** be to run the tests each time a user requests to initiate a session with the TOE. For instance, this **could** be the case of testing a directory server before its interaction with the TSF during the user authentication process.

8777

8778 In FPT_TEE.1.2, the PP/ST author **should** specify what are the action(s) that the TSF **shall**
8779 perform when the testing fails.

EXAMPLE 3

Examples of these action(s), illustrated by a directory server instance, **may** include to connect to an alternative available server or otherwise to look for a backup server.

8780 **J.16 Internal TOE TSF data replication consistency (FPT_TRC)**8781 **J.16.1 User notes**

8782 The requirements of this family are needed to ensure the consistency of TSF data when such
8783 data is replicated internal to the TOE. Such data **may** become inconsistent if an internal channel
8784 between parts of the TOE becomes inoperative. If the TOE is internally structured as a network
8785 of parts of the TOE, this **can** occur when parts become disabled, network connections are
8786 broken, and so on.

8787 The method of ensuring consistency is not specified in this component. It **could** be attained
8788 through a form of transaction logging (where appropriate transactions are “rolled back” to a
8789 site upon reconnection); it **could** be updating the replicated data through a synchronization
8790 protocol. If a particular protocol is necessary for a PP/ST, it **can** be specified through
8791 refinement.

8792 It **can** be impossible to synchronize some states, or the cost of such synchronization **can** be too
8793 high.

EXAMPLE

Examples of this situation are communication channel and encryption key revocations.

8794 Indeterminate states **can** also occur; if a specific behaviour is desired, it **should** be specified via
8795 refinement.

8796 **J.16.2 FPT_TRC.1 Internal TSF consistency**8797 **J.16.2.1 Operations**8798 **J.16.2.1.1 Assignment**

8799 In FPT_TRC.1.2, the PP/ST author **should** specify the list of functions dependent on TSF data
8800 replication consistency.

8801 **J.17 TSF self-test (FPT_TST)**8802 **J.17.1 User notes**

8803 The family defines the requirements for the self-testing of the TSF with respect to some
8804 expected correct operation.

EXAMPLE

Examples are interfaces to enforcement functions, and sample arithmetical operations on critical parts of the TOE.

8805

8806 These tests **can** be carried out at start-up, periodically, at the request of an authorized user, or
8807 when other conditions are met. The actions to be taken by the TOE as the result of self-testing
8808 are defined in other families.

8809 The requirements of this family are also needed to detect the corruption of TSF data and TSF
8810 itself (i.e. TSF executable code or TSF hardware component) by various failures that do not
8811 necessarily stop the TOE's operation (which would be handled by other families). These checks
8812 must be performed because these failures **may** not necessarily be prevented. Such failures **can**
8813 occur either because of unforeseen failure modes or associated oversights in the design of
8814 hardware, firmware, or software, or because of malicious corruption of the TSF due to
8815 inadequate logical and/or physical protection.

8816 In addition, use of this component may, with appropriate conditions, help to prevent
8817 inappropriate or damaging TSF changes being applied to an operational TOE as the result of
8818 maintenance activities.

8819 The term “correct operation of the TSF” refers primarily to the operation of the TSF and the
8820 integrity of the TSF data.

8821 **J.17.2 FPT_TST.1 TSF testing**

8822 **J.17.2.1 User application notes**

8823 This component provides support for the testing of the critical functions of the TSF's operation
8824 by requiring the ability to invoke testing functions and check the integrity of TSF data and
8825 executable code.

8826 **J.17.2.2 Evaluator notes**

8827 It is acceptable for the functions that are available to the authorized user for periodic testing to
8828 be available only in an off-line or maintenance mode. Controls **should** be in place to limit access
8829 during these modes to authorized users.

8830 **J.17.2.3 Operations**

8831 **J.17.2.3.1 Selection**

8832 In FPT_TST.1.1, the PP/ST author **should** specify when the TSF will execute the TSF test; during
8833 initial start-up, periodically during normal operation, at the request of an authorized user, at
8834 other conditions. In the case of the latter option, the PP/ST author **should** also assign what
8835 those conditions are via the following assignment.

8836 In FPT_TST.1.1, the PP/ST author **should** specify whether the self-tests are to be carried out to
8837 demonstrate the correct operation of the entire TSF, or of only specified parts of TSF.

8838 **J.17.2.3.2 Assignment**

8839 In FPT_TST.1.1, the PP/ST author **should**, if selected, specify the conditions under which the
8840 self-test **should** take place.

8841 In FPT_TST.1.1, the PP/ST author **should**, if selected, specify the list of parts of the TSF that will
8842 be subject to TSF self-testing.

8843 **J.17.2.3.3 Selection**

8844 In FPT_TST.1., the PP/ST author **should** specify whether data integrity is to be verified for all
8845 TSF data, or only for selected data.

8846 **J.17.2.3.4 Assignment**

8847 In FPT_TST.1., the PP/ST author **should**, if selected, specify the list of TSF data that will be
8848 verified for integrity.

8849 **J.17.2.3.5 Selection**

8850 In FPT_TST.1., the PP/ST author **should** specify whether TSF integrity is to be verified for all
8851 TSF, or only for selected TSF.

8852 **J.17.2.3.6 Assignment**

8853 In FPT_TST.1., the PP/ST author **should**, if selected, specify the list of TSF that will be verified
8854 for integrity.

Annex K (normative)

Class FRU: Resource utilization- application notes

8859 **K.1 General information**

8860 This class provides three families that support the availability of required resources such as
8861 processing capability and/or storage capacity. The family Fault Tolerance provides protection
8862 against unavailability of capabilities caused by failure of the TOE. The family Priority of Service
8863 ensures that the resources will be allocated to the more important or time-critical tasks and
8864 cannot be monopolized by lower priority tasks. The family Resource Allocation provides limits
8865 on the use of available resources, therefore preventing users from monopolizing the resources.

8866 **K.2 Fault tolerance (FRU_FLT)**

8867 **K.2.1 User notes**

8868 This family provides requirements for the availability of capabilities even in the case of failures.

EXAMPLE 1

Examples of such failures are power failure, hardware failure, or software error.

8869
8870 In case of these errors, if so specified, the TOE will maintain the specified capabilities.

EXAMPLE 2

The PP/ST author **could** specify that a TOE used in a nuclear plant will continue the operation of the shut-down procedure in the case of power-failure or communication-failure

8871
8872 Because the TOE **can** only continue its correct operation if the SFRs are enforced, there is a
8873 requirement that the system must remain in a secure state after a failure. This capability is
8874 provided by FPT_FLS.1 Failure with preservation of secure state.

8875 The mechanisms to provide fault tolerance **could** be active or passive. In case of an active
8876 mechanism, specific functions are in place that are activated in case the error occurs. For
8877 example, a fire alarm is an active mechanism: the TSF will detect the fire and **can** take action
8878 such as switching operation to a backup. In a passive scheme, the architecture of the TOE is
8879 capable of handling the error. For example, the use of a majority voting scheme with multiple
8880 processors is a passive solution; failure of one processor will not disrupt the operation of the
8881 TOE (although it needs to be detected to allow correction).

8882 For this family, it does not matter whether the failure has been initiated accidentally (such as
8883 flooding or unplugging the wrong device) or intentionally (such as monopolizing).

8884 **K.2.2 FRU_FLT.1 Degraded fault tolerance**

8885 **K.2.2.1 User application notes**

8886 This component is intended to specify which capabilities the TOE will still provide after a failure
8887 of the system. Since it would be difficult to describe all specific failures, categories of failures
8888 **may** be specified.

EXAMPLE

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

8889 **K.2.2.2 Operations**8890 **K.2.2.2.1 Assignment**

8891 In FRU_FLT.1.1, the PP/ST author **should** specify the list of TOE capabilities the TOE will
8892 maintain during and after a specified failure.

8893 In FRU_FLT.1.1, the PP/ST author **should** specify the list of types of failures against which the
8894 TOE has to be explicitly protected. If a failure in this list occurs, the TOE will be able to continue
8895 its operation.

8896 **K.2.3 FRU_FLT.2 Limited fault tolerance**8897 **K.2.3.1 User application notes**

8898 This component is intended to specify against what type of failures the TOE must be resistant.
8899 Since it would be difficult to describe all specific failures, categories of failures **may** be specified.

EXAMPLE

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

8900

8901 **K.2.3.2 Operations**8902 **K.2.3.2.1 Assignment**

8903 In FRU_FLT.2.1, the PP/ST author **should** specify the list of types of failures against which the
8904 TOE has to be explicitly protected. If a failure in this list occurs, the TOE will be able to continue
8905 its operation.

8906 **K.3 Priority of service (FRU_PRS)**8907 **K.3.1 User notes**

8908 The requirements of this family allow the TSF to control the use of resources under the control
8909 of the TSF by users and subjects such that high priority activities under the control of the TSF
8910 will always be accomplished without interference or delay due to low priority activities. In
8911 other words, time critical tasks will not be delayed by tasks that are less time critical.

8912 This family **could** be applicable to several types of resources.

EXAMPLE

processing capacity, and communication channel capacity.

8913 The Priority of Service mechanism might be passive or active. In a passive Priority of Service
8914 system, the system will select the task with the highest priority when given a choice between
8915 two waiting applications. While using passive Priority of Service mechanisms, when a low
8916 priority task is running, it cannot be interrupted by a high priority task. While using an active
8917 Priority of Service mechanisms, lower priority tasks might be interrupted by new high priority
8918 tasks.

8919 The audit requirement states that all reasons for rejection **should** be audited. It is left to the
8920 developer to argue that an operation is not rejected but delayed.

8921 **K.3.2 FRU_PRS.1 Limited priority of service**8922 **K.3.2.1 User application notes**

8923 This component defines priorities for a subject, and the resources for which this priority will be
8924 used. If some subject attempts to act on a resource controlled by the Priority of Service
8925 requirements, the access and/or time of access will be dependent on the subject's priority, the
8926 priority of the currently acting subject, and the priority of the subjects still in the queue.

8927 **K.3.2.2 Operations**

8928 **K.3.2.2.1 Assignment**

8929 In FRU_PRS.1.2, the PP/ST author **should** specify the list of controlled resources for which the
8930 TSF enforces priority of service

EXAMPLE

resources such as processes, disk space, memory, bandwidth.

8931 **K.3.3 FRU_PRS.2 Full priority of service**

8932 **K.3.3.1 User application notes**

8933 This component defines priorities for a subject. All shareable resources under the control of the
8934 TSF will be subjected to the Priority of Service mechanism. If some subject attempts to take
8935 action on a shareable TSF resource, the access and/or time of access will be dependent on the
8936 subject's priority, the priority of the currently acting subject, and the priority of the subjects still
8937 in the queue.

8938 **K.4 Resource allocation (FRU_RSA)**

8939 **K.4.1 User notes**

8940 The requirements of this family allow the TSF to control the use of resources under the control
8941 of the TSF by users and subjects such that unauthorized denial of service will not take place by
8942 means of monopolization of resources by other users or subjects.

8943 Resource allocation rules allow the creation of quotas or other means of defining limits on the
8944 amount of resource space or time that **may** be allocated on behalf of a specific user or subjects.

EXAMPLE 1

These rules may, for example:

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

8945 In general, these functions will be implemented through the use of attributes assigned to users
8946 and resources.

8947 The objective of these components is to ensure a certain amount of fairness among the users
8948 and subjects.

EXAMPLE 2

A single user **should** not allocate all the available space

8949 Since resource allocation often goes beyond the lifespan of a subject (i.e. files often exist longer
8950 than the applications that generated them), and multiple instantiations of subjects by the same
8951 user **should** not negatively affect other users too much, the components allow that the
8952 allocation limits are related to the users. In some situations, the resources are allocated by a
8953 subject.

EXAMPLE 3

Main memory or CPU cycles.

8954 In those instances, the components allow that the resource allocation be on the level of subjects.

8955 This family imposes requirements on resource allocation, not on the use of the resource itself.
8956 The audit requirements therefore, as stated, also apply to the allocation of the resource, not to
8957 the use of the resource.

8958 **K.4.2 FRU_RSA.1 Maximum quotas**8959 **K.4.2.1 User application notes**

8960 This component provides requirements for quota mechanisms that apply to only a specified set
8961 of the shareable resources in the TOE. The requirements allow the quotas to be associated with
8962 a user, possibly assigned to groups of users or subjects as applicable to the TOE.

8963 **K.4.2.2 Operations**8964 **K.4.2.2.1 Assignment**

8965 In FRU_RSA.1.1, the PP/ST author **should** specify the list of controlled resources for which
8966 maximum resource allocation limits are required.

EXAMPLE

Example of controlled resources include processes, disk space, memory, and bandwidth.

8967 If all resources under the control of the TSF need to be included, the words “all TSF resources”
8968 **may** be specified.

8969 **K.4.2.2.2 Selection**

8970 In FRU_RSA.1.1, the PP/ST author **should** select whether the maximum quotas apply to
8971 individual users, to a defined group of users, or subjects or any combination of these.

8972 In FRU_RSA.1.1, the PP/ST author **should** select whether the maximum quotas are applicable to
8973 any given time (simultaneously), or over a specific time interval.

8974 **K.4.3 FRU_RSA.2 Minimum and maximum quotas**8975 **K.4.3.1 User application notes**

8976 This component provides requirements for quota mechanisms that apply to a specified set of
8977 the shareable resources in the TOE. The requirements allow the quotas to be associated with a
8978 user, or possibly assigned to groups of users as applicable to the TOE.

8979 **K.4.3.2 Operations**8980 **K.4.3.2.1 Assignment**

8981 In FRU_RSA.2.1, the PP/ST author **should** specify the controlled resources for which maximum
8982 and minimum resource allocation limits are required.

EXAMPLE

Example of controlled resources include processes, disk space, memory, and bandwidth.

8983 If all resources under the control of the TSF need to be included, the words “all TSF resources”
8984 **can** be specified.

8985 **K.4.3.2.2 Selection**

8986 In FRU_RSA.2.1, the PP/ST author **should** select whether the maximum quotas apply to
8987 individual users, to a defined group of users, or subjects or any combination of these.

8988 In FRU_RSA.2.1, the PP/ST author **should** select whether the maximum quotas are applicable to
8989 any given time (simultaneously), or over a specific time interval.

8990 **K.4.3.2.3 Assignment**

8991 In FRU_RSA.2.2, the PP/ST author specifies the controlled resources for which a minimum
8992 allocation limit needs to be set.

EXAMPLE

Example of controlled resources include processes, disk space, memory, and bandwidth.

8993 If all resources under the control of the TSF need to be included the words “all TSF resources”
8994 **can** be specified.

8995 **K.4.3.2.4 Selection**

8996 In FRU_RSA.2.2, the PP/ST author selects whether the minimum quotas apply to individual
8997 users, to a defined group of users, or subjects or any combination of these.

8998 In FRU_RSA.2.2, the PP/ST author selects whether the minimum quotas are applicable to any
8999 given time (simultaneously), or over a specific time interval.

9000
9001
9002
9003

Annex L (normative)

Class FTA: TOE access- application notes

9004 L.1 General information

9005 The establishment of a user's session typically consists of the creation of one or more subjects
9006 that perform operations in the TOE on behalf of the user. At the end of the session
9007 establishment procedure, provided the TOE access requirements are satisfied, the created
9008 subjects bear the attributes determined by the identification and authentication functions. This
9009 family specifies functional requirements for controlling the establishment of a user's session.

9010 A user session is defined as the period starting at the time of the identification/authentication,
9011 or if more appropriate, the start of an interaction between the user and the system, up to the
9012 moment that all subjects (resources and attributes) related to that session have been
9013 deallocated.

9014 L.2 Limitation on scope of selectable attributes (FTA_LSA)

9015 L.2.1 User notes

9016 This family defines requirements that will limit the session security attributes a user **may** select,
9017 and the subjects to which a user **may** be bound, based on: the method of access; the location or
9018 port of access; and/or the time.

EXAMPLE 1
Time-of-day, day-of-week.

9019 This family provides the capability for a PP/ST author to specify requirements for the TSF to
9020 place limits on the domain of an authorized user's security attributes based on an
9021 environmental condition.

EXAMPLE 2
A user **could** be allowed to establish a "secret session" during normal business hours but
outside those hours the same user **could** be constrained to only establishing "unclassified
sessions".

9022 The identification of relevant constraints on the domain of selectable attributes **may** be
9023 achieved through the use of the selection operation. These constraints **may** be applied on an
9024 attribute-by-attribute basis. When there exists a need to specify constraints on multiple
9025 attributes this component will have to be replicated for each attribute.

EXAMPLE 3
Examples of attributes that **could** be used to limit the session security attributes are:

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

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

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

9026 L.2.2 FTA_LSA.1 Limitation on scope of selectable attributes

9027 L.2.2.1 Operations

9028 **L.2.2.1.1 Assignment**

9029 In FTA_LSA.1.1, the PP/ST author specifies the set of session security attributes that are to be
9030 constrained.

EXAMPLE 1

Examples of these session security attributes are user clearance level, integrity level and roles.

9031 In FTA_LSA.1.1, the PP/ST author specifies the set of attributes that **can** be used to determine
9032 the scope of the session security attributes.

EXAMPLE 2

Examples of such attributes are user identity, originating location, time of access, and method of access.

9033 **L.3 Limitation on multiple concurrent sessions (FTA_MCS)**

9034 **L.3.1 User notes**

9035 This family defines how many sessions a user **may** have at the same time (concurrent sessions).
9036 This number of concurrent sessions **may** either be set for a group of users or for each individual
9037 user.

9038 **L.3.2 FTA_MCS.1 Basic limitation on multiple concurrent sessions**

9039 **L.3.2.1 User application notes**

9040 This component allows the system to limit the number of sessions in order to effectively use the
9041 resources of the TOE.

9042 **L.3.2.2 Operations**

9043 **L.3.2.2.1 Assignment**

9044 In FTA_MCS.1.2, the PP/ST author specifies the default number of maximum concurrent
9045 sessions to be used.

9046 **L.3.3 FTA_MCS.2 Per user attribute limitation on multiple concurrent sessions**

9047 **L.3.3.1 User application notes**

9048 This component provides additional capabilities over those of FTA_MCS.1 Basic limitation on
9049 multiple concurrent sessions, by allowing further constraints to be placed on the number of
9050 concurrent sessions that users are able to invoke. These constraints are in terms of a user's
9051 security attributes, such as a user's identity, or membership of a role.

9052 **L.3.3.2 Operations**

9053 **L.3.3.2.1 Assignment**

9054 In FTA_MCS.2.1, the PP/ST author specifies the rules that determine the maximum number of
9055 concurrent sessions.

EXAMPLE

An example of a rule is "maximum number of concurrent sessions is one if the user has a classification level of "secret" and five otherwise".

9056 In FTA_MCS.2.2, the PP/ST author specifies the default number of maximum concurrent
9057 sessions to be used.

9058 **L.4 Session locking and termination (FTA_SSL)**

9059 **L.4.1 User notes**

9060 This family defines requirements for the TSF to provide the capability for TSF-initiated and
9061 user-initiated locking, unlocking, and termination of interactive sessions.

9062 When a user is directly interacting with subjects in the TOE (interactive session), the user's
9063 terminal is vulnerable if left unattended. This family provides requirements for the TSF to
9064 disable (lock) the terminal or terminate the session after a specified period of inactivity, and for
9065 the user to initiate the disabling (locking) of the terminal or terminate the session. To reactivate
9066 the terminal, an event specified by the PP/ST author, such as the user re-authentication must
9067 occur.

9068 A user is considered inactive, if he/she has not provided any stimulus to the TOE for a specified
9069 period of time.

9070 PP/ST authors consider whether FTP_TRP.1 Trusted path **should** be included. In that case, the
9071 function "session locking" **must** be included in the operation in FTP_TRP.1 Trusted path.

9072 **L.4.2 FTA_SSL.1 TSF-initiated session locking**

9073 **L.4.2.1 User application notes**

9074 FTA_SSL.1 TSF-initiated session locking, provides the capability for the TSF to lock an active
9075 user session after a specified period of time. Locking a terminal would prevent any further
9076 interaction with an existing active session through the use of the locked terminal.

9077 If display devices are overwritten, the replacement contents need not be static (i.e. "screen
9078 savers" are permitted).

9079 This component allows the PP/ST author to specify what events will unlock the session. These
9080 events **may** be related to the terminal, the user, or time.

EXAMPLE

Examples of events include

- Terminal related: a fixed set of keystrokes to unlock the session.
- User related: reauthentication.
- Time related: after 15 minutes.

9081

9082 **L.4.2.2 Operations**

9083 **L.4.2.2.1 Assignment**

9084 In FTA_SSL.1.1, the PP/ST author specifies the interval of user inactivity that will trigger the
9085 locking of an interactive session. If so desired the PP/ST author **could**, through the assignment,
9086 specify that the time interval is left to the authorized administrator or the user. The
9087 management functions in the FMT class **can** specify the capability to modify this time interval,
9088 making it the default value.

9089 In FTA_SSL.1.2, the PP/ST author specifies the event(s) that **should** occur before the session is
9090 unlocked.

EXAMPLE

Examples of such an event are: "user re-authentication" or "user enters unlock key-sequence".

9091

9092 **L.4.3 FTA_SSL.2 User-initiated locking**

9093 **L.4.3.1 User application notes**

9094 FTA_SSL.2 User-initiated locking, provides the capability for an authorized user to lock and
9095 unlock his/her own interactive session. This would provide authorized users with the ability to

9096 effectively block further use of their active sessions without having to terminate the active
9097 session.

9098 If devices are overwritten, the replacement contents need not be static (i.e. “screen savers” are
9099 permitted).

9100 **L.4.3.2 Operations**

9101 **L.4.3.2.1 Assignment**

9102 In FTA_SSL.2.2, the PP/ST author specifies the event(s) that **must** occur before the session is
9103 unlocked.

EXAMPLE

Examples of such an event are: “user re-authentication”, or “user enters unlock key-sequence”.

9104 **L.4.4 FTA_SSL.3 TSF-initiated termination**

9105 **L.4.4.1 User application notes**

9106 FTA_SSL.3 TSF-initiated termination, requires that the TSF terminate an interactive user
9107 session after a period of inactivity.

9108 The PP/ST author **should** be aware that a session **may** continue after the user terminated
9109 his/her activity. This requirement would terminate this background subject after a period of
9110 inactivity of the user without regard to the status of the subject.

EXAMPLE

An example of a continuing session after a user terminated activity is background processing.

9111 **L.4.4.2 Operations**

9112 **L.4.4.2.1 Assignment**

9113 In FTA_SSL.3.1, the PP/ST author specifies the interval of user inactivity that will trigger the
9114 termination of an interactive session. If so desired, the PP/ST author **could**, through the
9115 assignment, specify that the interval is left to the authorized administrator or the user. The
9116 management functions in the FMT class **can** specify the capability to modify this time interval,
9117 making it the default value.

9118 **L.4.5 FTA_SSL.4 User-initiated termination**

9119 User application notes

9120 FTA_SSL.4 User-initiated termination, provides the capability for an authorized user to
9121 terminate his/her interactive session.

9122 The PP/ST author **should** be aware that a session **could** continue after the user terminated
9123 his/her activity.

EXAMPLE

An example of a continuing session after a user terminated activity is background processing.

9124 This requirement would allow the user to terminate this background subject without regard to
9125 the status of the subject.

9126 **L.5 TOE access banners (FTA_TAB)**

9127 **L.5.1 User notes**

9128 Prior to identification and authentication, TOE access requirements provide the ability for the
 9129 TOE to display an advisory warning message to potential users pertaining to appropriate use of
 9130 the TOE.

9131 **L.5.2 FTA_TAB.1 Default TOE access banners**

9132 **L.5.2.1 User application notes**

9133 This component requires that there is an advisory warning regarding the unauthorized use of
 9134 the TOE. A PP/ST author **could** refine the requirement to include a default banner.

9135 **L.6 TOE access history (FTA_TAH)**

9136 **L.6.1 User notes**

9137 This family defines requirements for the TSF to display to users, upon successful session
 9138 establishment to the TOE, a history of unsuccessful attempts to access the account. This history
 9139 **could** include the date, time, means of access, and port of the last successful access to the TOE,
 9140 as well as the number of unsuccessful attempts to access the TOE since the last successful
 9141 access by the identified user.

9142 **L.6.2 FTA_TAH.1 TOE access history**

9143 **L.6.2.1 User application notes**

9144 This family **can** provide authorized users with information that **could** indicate the possible
 9145 misuse of their user account.

9146 This component requests that the user is presented with the information. The user **should** be
 9147 able to review the information but is not forced to do so.

EXAMPLE

A user might create scripts that ignore this information and start other processes.

9148 **L.6.2.2 Operations**

9149 **L.6.2.2.1 Selection**

9150 In FTA_TAH.1.1, the PP/ST author selects the security attributes of the last successful session
 9151 establishment that will be shown at the user interface. The items are: date, time, method of
 9152 access, and/or location.

9153 In FTA_TAH.1.2, the PP/ST author selects the security attributes of the last unsuccessful session
 9154 establishment that will be shown at the user interface. The items are: date, time, method of
 9155 access, and/or location.

EXAMPLE

Method of access: ftp.

Location: terminal 50.

9156 **L.7 TOE session establishment (FTA_TSE)**

9157 **L.7.1 User notes**

9158 This family defines requirements to deny a user permission to establish a session with the TOE
 9159 based on attributes such as the location or port of access, the user's security attribute, ranges of
 9160 time or combinations of parameters.

EXAMPLE 1

Security attribute: identity, clearance level, integrity level, membership in a role.

Ranges of time: time-of-day, day-of-week, calendar dates.

9161 This family provides the capability for the PP/ST author to specify requirements for the TOE to
9162 place constraints on the ability of an authorized user to establish a session with the TOE. The
9163 identification of relevant constraints **can** be achieved through the use of the selection operation.

EXAMPLE 2

Examples of attributes that **could** be used to specify the session establishment constraints are:

- a) The location of access can be used to constrain the ability of a user to establish an active session with the TOE, based on the user's location or port of access. This capability is of particular use in environments where dial-up facilities or network facilities are available.
- b) The user's security attributes can be used to place constraints on the ability of a user to establish an active session with the TOE. For example, these attributes would provide the capability to deny session establishment based on any of the following:
 - a user's identity;
 - a user's clearance level;
 - a user's integrity level; and
 - a user's membership in a role.

This capability is particularly relevant in situations where authorization or login may take place at a different location from where TOE access checks are performed.

- c) The time of access can be used to constrain the ability of a user to establish an active session with the TOE based on ranges of time. For example, ranges may be based upon time-of-day, day-of-week, or calendar dates. This constraint provides some operational protection against actions that **could** occur at a time where proper monitoring or where proper procedural measures may not be in place.

9164 **L.7.2 FTA_TSE.1 TOE session establishment**

9165 **L.7.2.1 Operations**

9166 **L.7.2.1.1 Assignment**

9167 In FTA_TSE.1.1, the PP/ST author specifies the attributes that **can** be used to restrict the session
9168 establishment.

EXAMPLE

Examples of possible attributes are user identity, originating location (such as no remote terminals), time of access (such as outside hours), or method of access (such as telnet).

9169

9170
9171
9172
9173

Annex M (normative)

Class FTP: Trusted path/channels- application notes

9174 **M.1 General information**

9175 Users often need to perform functions through direct interaction with the TSF. A trusted path
9176 provides confidence that a user is communicating directly with the TSF whenever it is invoked.
9177 A user's response via the trusted path guarantees that untrusted applications cannot intercept
9178 or modify the user's response. Similarly, trusted channels are one approach for secure
9179 communication between the TSF and another trusted IT product.

9180 Absence of a trusted path **may** allow breaches of accountability or access control in
9181 environments where untrusted applications are used. These applications **can** intercept user-
9182 private information, such as passwords, and use it to impersonate other users. As a
9183 consequence, responsibility for any system actions cannot be reliably assigned to an
9184 accountable entity. Also, these applications **could** output erroneous information on an
9185 unsuspecting user's display, resulting in subsequent user actions that **could** be erroneous and
9186 **could** lead to a security breach.

9187 **M.2 Inter-TSF trusted channel (FTP_ITC)**

9188 **M.2.1 User notes**

9189 This family defines the rules for the creation of a trusted channel connection that goes between
9190 the TSF and another trusted IT product for the performance of security critical operations
9191 between the products.

EXAMPLE

An example of such a security critical operation is the updating of the TSF authentication database by the transfer of data from a trusted product whose function is the collection of audit data.

9192 **M.2.2 FTP_ITC.1 Inter-TSF trusted channel**

9193 **M.2.2.1 User application notes**

9194 This component is used when a trusted communication channel between the TSF and another
9195 trusted IT product is required.

9196 **M.2.2.2 Operations**

9197 **M.2.2.2.1 Selection**

9198 In FTP_ITC.1.2, the PP/ST author must specify whether the local TSF, another trusted IT
9199 product, or both **shall** have the capability to initiate the trusted channel.

9200 **M.2.2.2.2 Assignment**

9201 In FTP_ITC.1.3, the PP/ST author specifies the functions for which a trusted channel is required.

EXAMPLE

Examples of these functions **may** include transfer of user, subject, and/or object security attributes and ensuring consistency of TSF data.

9202 **M.3 Secure channel (FTP_PRO)**

9203 **M.3.1 User notes**

9204 This family defines the rules for the creation of a secure channel connection that goes between
9205 the TSF and another trusted IT product for the protection of data transfers.

9206 Separate iterations of the relevant FTP_PRO SFRs **may** be used for different roles where the
9207 completion of the SFR needs to be different for each role.

9208 **M.3.2 FTP_PRO.1**

9209 **M.3.2.1 User application notes**

9210 Where values used in the completion of FTP_PRO operations have dependencies between
9211 different SFR elements, these need to be made clear in the instantiation of the SFR.

EXAMPLE

A table could be given in which the columns represent the relevant selections and assignments, and the rows define the valid combination of completion values.

9212 **M.3.2.2 Operations**

9213 **M.3.2.2.1 Assignment**

9214 In FTP_PRO.1.1, if selected, the PP/ST author **should** specify a trusted channel protocol and the
9215 defined protocol roles.

EXAMPLE 1

Examples of “defined protocol roles” would be ‘client’ or ‘server’ (TLS), ‘initiator’ or ‘responder’ (IKEv2/IPsec), ‘Trust Center’ (ZigBee) or ‘Key Distribution Centre’ (Kerberos).

9216 In FTP_PRO.1.2 the first assignment is intended to state rules for when the secure channel is
9217 required to be used by the TOE, such as mandating its use for communications with an audit
9218 server. This assignment may take the value ‘None specified’ (also with ‘None specified’ in the
9219 second assignment) if no specific uses of the channel are mandated for the TOE.

9220 In FTP_PRO.1.5 the assignment is intended to state rules related to implementation of the
9221 protocol(s). It may take the value ‘None specified’ if no rules are required, or if the standards
9222 referenced in other elements of the SFR include the relevant rules and no specific evaluator
9223 check is required for the context in which the SFR is being used.

EXAMPLE 2

Rules include those for maximum packet sizes or rekey intervals

9224 In FTP_PRO.1.6 the assignment is intended to state rules related to negotiable aspects of the
9225 protocol, when intending to narrow the options provided by the TOE compared to the standard
9226 that defines the protocol.

EXAMPLE 3

The selection of ciphersuites or acceptance of older protocol versions.

9227 The assignment **may** take the value ‘None specified’ if no rules are required. Where the
9228 assignment is completed with a list then that list specifies the only configurations permitted –
9229 any other configuration would be a violation of the SFR. This element may be used to specify
9230 mandatory supported configurations without limiting the TOE to using these configurations by,
9231 for example, listing the required configurations with “(support required)” after each entry in
9232 the list and then including a final element which states that any other configuration permitted
9233 by the standard is allowed.

9234 **M.3.3 FTP_PRO.2**

9235 **M.3.3.1 User application notes**

9236 In FTP_PRO.2, the ‘list of rules for carrying out the authentication’ may be used to limit available
9237 parameters for the authentication mechanisms.

EXAMPLE

Rules might be stated for the format (e.g. FQDN or IP address, use of wildcards) or prioritization of identifiers when alternative sources of an identifier are available in the authentication data exchanged.

9238 **M.3.3.2 Operations**

9239 **M.3.3.2.1 Selection**

9240 In FTP_PRO.2.2 the selection indicating the direction of the authentication should be chosen.

9241 **M.3.3.2.2 Assignment**

9242 In FTP_PRO.2.1 The PP/ST author provides a list of key establishment mechanisms.

9243 In FTP_PRO.2.2 the assignments include providing a list of authentication mechanisms used
9244 during the authentication and a list of rules used during the authentication.

9245 **M.3.4 FTP_PRO.3**

9246 **M.3.4.1 User application notes**

9247 The FTP_PRO.3 component addresses protection (confidentiality and integrity) of data in
9248 transit through a trusted channel.

9249 **M.3.4.2 Operations**

9250 **M.3.4.2.1 Selection**

9251 The PP/ST author selects the attacks that are mitigated by the TSF.

9252 **M.3.4.2.2 Assignment**

9253 The PP/ST author completes the assignment by specifying a list of integrity protection
9254 mechanisms.

EXAMPLE

Examples of integrity protection mechanism include protection of contents and file-system permissions of system files and directories; protection of processes against code injection, and protection against unsigned kernel extensions.

9255 **M.4 Trusted path (FTP_TRP)**

9256 **M.4.1 User notes**

9257 This family defines the requirements to establish and maintain trusted communication to or
9258 from users and the TSF. A trusted path **may** be required for any security-relevant interaction.
9259 Trusted path exchanges **may** be initiated by a user during an interaction with the TSF, or the
9260 TSF **may** establish communication with the user via a trusted path.

9261 **M.4.2 FTP_TRP.1 Trusted path**

9262 **M.4.2.1 User application notes**

9263 This component is used when trusted communication between a user and the TSF is required,
9264 either for initial authentication purposes only or for additional specified user operations.

9265 **M.4.2.2 Operations**

9266 **M.4.2.2.1 Selection**

9267 In FTP_TRP.1.1, the PP/ST author specifies whether the trusted path must be extended to
9268 remote and/or local users.

9269 In FTP_TRP.1.1, the PP/ST author specifies whether the trusted path **shall** protect the data from
9270 modification, disclosure, and/or other types of integrity or confidentiality violation.

9271 **M.4.2.2.2 Assignment**

- 9272 In FTP_TRP.1.1, if selected, the PP/ST author identifies any additional types of integrity or
9273 confidentiality violation against which the trusted path shall protect the data.
- 9274 **M.4.2.2.3 Selection**
- 9275 In FTP_TRP.1.2, the PP/ST author specifies whether the TSF, local users, and/or remote users
9276 are able to initiate the trusted path.
- 9277 In FTP_TRP.1.3, the PP/ST author specifies whether the trusted path is to be used for initial
9278 user authentication and/or for other specified services.
- 9279 **M.4.2.2.4 Assignment**
- 9280 In FTP_TRP.1.3, if selected, the PP/ST author identifies other services for which trusted path is
9281 required, if any.