

# collaborative Protection Profile for Network Devices

Version 1.14.0

21-Jul-2016~~27-Feb-2015~~

## Acknowledgements

This collaborative Protection Profile (cPP) was developed by the Network international Technical Community with representatives from industry, Government agencies, Common Criteria Test Laboratories, and members of academia.

## 0. Preface

### 0.1 Objectives of Document

This document presents the Common Criteria (CC) collaborative Protection Profile (cPP) to express the security functional requirements (SFRs) and security assurance requirements (SARs) for a network device. The Evaluation Activities that specify the actions the evaluator performs to determine if a product satisfies the SFRs captured within this cPP are described in [SD].

### 0.2 Scope of Document

The scope of the cPP within the development and evaluation process is described in the Common Criteria for Information Technology Security Evaluation [CC]. In particular, a cPP defines the IT security requirements of a generic type of TOE and specifies the functional and assurance security measures to be offered by that TOE to meet stated requirements [CC1, Section C.1].

### 0.3 Intended Readership

The target audiences of this cPP are developers, CC consumers, system integrators, evaluators and schemes.

Although the cPPs and SDs may contain minor editorial errors, cPPs are recognized as living documents and the iTCs are dedicated to ongoing updates and revisions. Please report any issues to the NDFW iTC.

### 0.4 Related Documents

#### Common Criteria<sup>1</sup>

- |       |   |
|-------|---|
| [CC1] | Common Criteria for Information Technology Security Evaluation,<br>Part 1: Introduction and General Model,<br>CCMB-2012-09-001, Version 3.1 Revision 4, September 2012. |
| [CC2] | Common Criteria for Information Technology Security Evaluation,<br>Part 2: Security Functional Components,<br>CCMB-2012-09-002, Version 3.1 Revision 4, September 2012. |
| [CC3] | Common Criteria for Information Technology Security Evaluation,<br>Part 3: Security Assurance Components,<br>CCMB-2012-09-003, Version 3.1 Revision 4, September 2012.  |

---

<sup>1</sup> For details see <http://www.commoncriteriaportal.org/>

[CEM] Common Methodology for Information Technology Security Evaluation,  
Evaluation Methodology,  
CCMB-2012-09-004, Version 3.1, Revision 4, September 2012.

### Other Documents

[SD] Evaluation Activities for Network Device cPP, Version 1.0, ~~27 February 2015~~  
1, 21 July 2016

## 0.5 Revision History

Version	Date	Description
<u>1.1</u>	<u>21-Jul-2016</u>	<u>Updated draft published for public review</u>
<del>1.01.0</del>	<del>27-Feb-2015</del> <u>27-Feb-2015</u>	Released for use
0.4	26-Jan-2015	Incorporated comments received from the CCDB review
0.3	17-Oct-2014	Draft version released to accompany CCDB review of Supporting Document.
0.2	13-Oct-2014	Internal draft in response to public review comments, for iTC review
0.1	05-Sep-2014	Draft published for Public review

## Contents

<b>Error! Hyperlink reference not valid.</b>	<b>Acknowledgements</b>	<b>2</b>
<b>Error! Hyperlink reference not valid.</b>	<b>0</b>	<b>Preface</b>
<b>Error! Hyperlink reference not valid.</b>	<b>0.1</b>	<b>Objectives of Document</b>
<b>Error! Hyperlink reference not valid.</b>	<b>0.2</b>	<b>Scope of Document</b>
<b>Error! Hyperlink reference not valid.</b>	<b>0.3</b>	<b>Intended Readership</b>
<b>Error! Hyperlink reference not valid.</b>	<b>0.4</b>	<b>Related Documents</b>
<b>Error! Hyperlink reference not valid.</b>	<b>0.5</b>	<b>Revision History</b>
<b>Error! Hyperlink reference not valid.</b>	<b>1</b>	<b>PP Introduction</b>
<b>Error! Hyperlink reference not valid.</b>	<b>1.1</b>	<b>PP Reference Identification</b>
<b>Error! Hyperlink reference not valid.</b>	<b>1.2</b>	<b>TOE Overview</b>
<b>Error! Hyperlink reference not valid.</b>	<b>1.3</b>	<b>TOE Use Cases</b>
<b>Error! Hyperlink reference not valid.</b>	<b>2</b>	<b>CC Conformance</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3</b>	<b>Security Problem Definition</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1</b>	<b>Threats</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.1</b>	<b>Communications with the Network Device</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.1.1</b>	<b>T.UNAUTHORIZED_ADMINISTRATOR_ACCESS</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.1.2</b>	<b>T.WEAK_CRYPTOGRAPHY</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.1.3</b>	<b>T.UNTRUSTED_COMMUNICATION_CHANNELS</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.1.4</b>	<b>T.WEAK_AUTHENTICATION_ENDPOINTS</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.2</b>	<b>Valid Updates</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.2.1</b>	<b>T.UPDATE_COMPROMISE</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.3</b>	<b>Audited Activity</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.3.1</b>	<b>T.UNDETECTED_ACTIVITY</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.4</b>	<b>Administrator and Device Credentials and Data</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.4.1</b>	<b>T.SECURITY_FUNCTIONALITY_COMPROMISE</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.4.2</b>	<b>T.PASSWORD_CRACKING</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.5</b>	<b>Device Failure</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.1.5.1</b>	<b>T.SECURITY_FUNCTIONALITY_FAILURE</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.2</b>	<b>Assumptions</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.2.1</b>	<b>A.PHYSICAL_PROTECTION</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.2.2</b>	<b>A.LIMITED_FUNCTIONALITY</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.2.3</b>	<b>A.NO_THRU_TRAFFIC_PROTECTION</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.2.4</b>	<b>A.TRUSTED_ADMINISTRATOR</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.2.5</b>	<b>A.REGULAR_UPDATES</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.2.6</b>	<b>A.ADMIN_CREDENTIALS_SECURE</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.3</b>	<b>Organizational Security Policy</b>
<b>Error! Hyperlink reference not valid.</b>	<b>3.3.1</b>	<b>P.ACCESS_BANNER</b>
<b>Error! Hyperlink reference not valid.</b>	<b>4</b>	<b>Security Objectives</b>
<b>Error! Hyperlink reference not valid.</b>	<b>4.1</b>	<b>Security Objectives for the Operational Environment</b>
<b>Error! Hyperlink reference not valid.</b>	<b>4.1.1</b>	<b>OE.PHYSICAL</b>
<b>Error! Hyperlink reference not valid.</b>	<b>4.1.2</b>	<b>OE.NO_GENERAL_PURPOSE</b>
<b>Error! Hyperlink reference not valid.</b>	<b>4.1.3</b>	<b>OE.NO_THRU_TRAFFIC_PROTECTION</b>
<b>Error! Hyperlink reference not valid.</b>	<b>4.1.4</b>	<b>OE.TRUSTED_ADMIN</b>
<b>Error! Hyperlink reference not valid.</b>	<b>4.1.5</b>	<b>OE.UPDATES</b>
<b>Error! Hyperlink reference not valid.</b>	<b>4.1.6</b>	<b>OE.ADMIN_CREDENTIALS_SECURE</b>
<b>Error! Hyperlink reference not valid.</b>	<b>5</b>	<b>Security Functional Requirements</b>
<b>Error! Hyperlink reference not valid.</b>	<b>5.1</b>	<b>Conventions</b>
<b>Error! Hyperlink reference not valid.</b>	<b>5.2</b>	<b>SFR Architecture</b>
<b>Error! Hyperlink reference not valid.</b>	<b>5.3</b>	<b>Security Audit (FAU)</b>
<b>Error! Hyperlink reference not valid.</b>	<b>5.3.1</b>	<b>Security Audit Data generation (FAU_GEN)</b>
<b>Error! Hyperlink reference not valid.</b>	<b>5.3.1.1</b>	<b>FAU_GEN.1 Audit data generation</b>
<b>Error! Hyperlink reference not valid.</b>	<b>5.3.1.2</b>	<b>FAU_GEN.2 User identity association</b>
<b>Error! Hyperlink reference not valid.</b>	<b>5.3.2</b>	<b>Security audit event storage (Extended FAU_STG_EXT)</b>
<b>Error! Hyperlink reference not valid.</b>	<b>5.3.2.1</b>	<b>FAU_STG_EXT.1 Protected Audit Event Storage</b>

<b>Error! Hyperlink reference not valid.</b> 5.4 .....	Cryptographic Support (FCS) —	30
<b>Error! Hyperlink reference not valid.</b> 5.4.1 .....	Cryptographic Key Management (FCS_CKM) —	30
<b>Error! Hyperlink reference not valid.</b> 5.4.1.1FCS_CKM.1 .....	Cryptographic Key Generation (Refined) —	30
<b>Error! Hyperlink reference not valid.</b> 5.4.1.2FCS_CKM.2 .....	Cryptographic Key Establishment (Refined) —	31
<b>Error! Hyperlink reference not valid.</b> 5.4.1.3 FCS_CKM.4 .....	Cryptographic Key Destruction —	32
<b>Error! Hyperlink reference not valid.</b> 5.4.2 .....	Cryptographic Operation (FCS_COP) —	32
<b>Error! Hyperlink reference not valid.</b> 5.4.2.1 .....	FCS_COP.1 Cryptographic Operation —	32
<b>Error! Hyperlink reference not valid.</b> 5.4.3 .. Random Bit Generation (Extended) —	FCS_RBG_EXT) —	34
<b>Error! Hyperlink reference not valid.</b> 5.4.3.1 ... FCS_RBG_EXT.1 .....	Random Bit Generation —	34
<b>Error! Hyperlink reference not valid.</b> 5.5 .....	Identification and Authentication (FIA) —	35
<b>Error! Hyperlink reference not valid.</b> 5.5.1 ... Password Management (Extended) —	FIA_PMG_EXT) —	35
<b>Error! Hyperlink reference not valid.</b> 5.5.1.1 ...FIA_PMG_EXT.1 .....	Password Management —	35
<b>Error! Hyperlink reference not valid.</b> 5.5.2User Identification and Authentication (Extended) —	FIA_UIA_EXT) —	36
<b>Error! Hyperlink reference not valid.</b> 5.5.2.1FIA_UIA_EXT.1 .....	User Identification and Authentication —	36
<b>Error! Hyperlink reference not valid.</b> 5.5.3User authentication (FIA_UAU) (Extended) —	FIA_UAU_EXT) —	36
<b>Error! Hyperlink reference not valid.</b> 5.5.3.1FIA_UAU_EXT.2 .....	Password-based Authentication Mechanism —	37
<b>Error! Hyperlink reference not valid.</b> 5.5.3.2FIA_UAU.7 .....	Protected Authentication Feedback —	37
<b>Error! Hyperlink reference not valid.</b> 5.5.4Authentication using X.509 certificates (Extended) —	FIA_X509_EXT) —	38
<b>Error! Hyperlink reference not valid.</b> 5.5.4.1FIA_X509_EXT.1 .....	X.509 Certificate Validation —	37
<b>Error! Hyperlink reference not valid.</b> 5.5.4.2FIA_X509_EXT.2 .....	X.509 Certificate Authentication —	38
<b>Error! Hyperlink reference not valid.</b> 5.5.4.3FIA_X509_EXT.3 .....	X.509 Certificate Requests —	39
<b>Error! Hyperlink reference not valid.</b> 5.6 .....	Security Management (FMT) —	39
<b>Error! Hyperlink reference not valid.</b> 5.6.1 .....	Management of functions in TSF (FMT_MOF) —	39
<b>Error! Hyperlink reference not valid.</b> 5.6.1.1FMT_MOF.1(1)/TrustedUpdate .....	Management of security functions —	39
<b>Error! Hyperlink reference not valid.</b> 5.6.2 .....	Management of TSF Data (FMT_MTD) —	39
<b>Error! Hyperlink reference not valid.</b> 5.6.2.1 .....	FMT_MTD.1 Management of TSF Data —	39
<b>Error! Hyperlink reference not valid.</b> 5.6.3 .....	Specification of Management Functions (FMT_SMF) —	40
<b>Error! Hyperlink reference not valid.</b> 5.6.3.1FMT_SMF.1 .....	Specification of Management Functions —	40
<b>Error! Hyperlink reference not valid.</b> 5.6.4 .....	Security management roles (FMT_SMR) —	41
<b>Error! Hyperlink reference not valid.</b> 5.6.4.1 ... FMT_SMR.2 .....	Restrictions on security roles —	41
<b>Error! Hyperlink reference not valid.</b> 5.7 .....	Protection of the TSF (FPT) —	41
<b>Error! Hyperlink reference not valid.</b> 5.7.1 .... Protection of TSF Data (Extended) —	FPT_SKP_EXT) —	41
<b>Error! Hyperlink reference not valid.</b> 5.7.1.1FPT_SKP_EXT.1 .....	Protection of TSF Data (for reading of all system data) —	41
<b>Error! Hyperlink reference not valid.</b> 5.7.2Protection of Administrator Passwords (Extended) —	FPT_APW_EXT) —	42
<b>Error! Hyperlink reference not valid.</b> 5.7.2.1FPT_APW_EXT.1 .....	Protection of Administrator Passwords —	42
<b>Error! Hyperlink reference not valid.</b> 5.7.3 .....	TSF testing (Extended) —	42
<b>Error! Hyperlink reference not valid.</b> 5.7.3.1 ... FPT_TST_EXT.1 .....	TSF Testing (Extended) —	42
<b>Error! Hyperlink reference not valid.</b> 5.7.4 .....	Trusted Update (FPT_TUD_EXT) —	43
<b>Error! Hyperlink reference not valid.</b> 5.7.4.1 .....	FPT_TUD_EXT.1 Trusted Update —	43
<b>Error! Hyperlink reference not valid.</b> 5.7.5 .....	Time stamps (FPT_STM) —	44
<b>Error! Hyperlink reference not valid.</b> 5.7.5.1 .....	FPT_STM.1 Reliable Time Stamps —	44
<b>Error! Hyperlink reference not valid.</b> 5.8 .....	TOE Access (FTA) —	45
<b>Error! Hyperlink reference not valid.</b> 5.8.1TSF initiated Session Locking (Extended) —	FTA_SSL_EXT) —	45
<b>Error! Hyperlink reference not valid.</b> 5.8.1.1FTA_SSL_EXT.1 .....	TSF initiated Session Locking —	45
<b>Error! Hyperlink reference not valid.</b> 5.8.2 .....	Session locking and termination (FTA_SSL) —	45
<b>Error! Hyperlink reference not valid.</b> 5.8.2.1 .....	FTA_SSL.3 TSF initiated Termination —	45
<b>Error! Hyperlink reference not valid.</b> 5.8.2.2 .....	FTA_SSL.4 User initiated Termination —	45
<b>Error! Hyperlink reference not valid.</b> 5.8.3 .....	TOE access banners (FTA_TAB) —	46
<b>Error! Hyperlink reference not valid.</b> 5.8.3.1 ... FTA_TAB.1 .....	Default TOE Access Banners —	46
<b>Error! Hyperlink reference not valid.</b> 5.9 .....	Trusted path/channels (FTP) —	46
<b>Error! Hyperlink reference not valid.</b> 5.9.1 .....	Trusted Channel (FTP_ITC) —	46
<b>Error! Hyperlink reference not valid.</b> 5.9.1.1FTP_ITC.1 .....	Inter TSF trusted channel (Refined) —	46
<b>Error! Hyperlink reference not valid.</b> 5.9.2 .....	Trusted Path (FTP_TRP) —	47
<b>Error! Hyperlink reference not valid.</b> 5.9.2.1 .....	FTP_TRP.1 Trusted Path (Refinement) —	47
<b>Error! Hyperlink reference not valid.</b> 6 .....	Security Assurance Requirements —	49
<b>Error! Hyperlink reference not valid.</b> 6.1 .....	ASE: Security Target —	49
<b>Error! Hyperlink reference not valid.</b> 6.2 .....	ADV: Development —	50
<b>Error! Hyperlink reference not valid.</b> 6.2.1 .....	Basic Functional Specification (ADV_FSP.1) —	50

<b>Error! Hyperlink reference not valid.</b> 6.3 .....	AGD: Guidance Documentation	50
<b>Error! Hyperlink reference not valid.</b> 6.3.1 .....	Operational User Guidance (AGD_OPE.1)	51
<b>Error! Hyperlink reference not valid.</b> 6.3.2 .....	Preparative Procedures (AGD_PRE.1)	51
<b>Error! Hyperlink reference not valid.</b> 6.4 .....	Class ALC: Life cycle Support	51
<b>Error! Hyperlink reference not valid.</b> 6.4.1 .....	Labelling of the TOE (ALC_CMC.1)	51
<b>Error! Hyperlink reference not valid.</b> 6.4.2 .....	TOE CM Coverage (ALC_CMS.1)	51
<b>Error! Hyperlink reference not valid.</b> 6.5 .....	Class ATE: Tests	51
<b>Error! Hyperlink reference not valid.</b> 6.5.1 .....	Independent Testing – Conformance (ATE_IND.1)	52
<b>Error! Hyperlink reference not valid.</b> 6.6 .....	Class AVA: Vulnerability Assessment	52
<b>Error! Hyperlink reference not valid.</b> 6.6.1 .....	Vulnerability Survey (AVA_VAN.1)	52
<b>Error! Hyperlink reference not valid.</b> A .....	Optional Requirements	53
<b>Error! Hyperlink reference not valid.</b> A.1 .....	Audit Events for Optional SFRs	53
<b>Error! Hyperlink reference not valid.</b> A.2 .....	Security Audit (FAU)	54
<b>Error! Hyperlink reference not valid.</b> A.2.1 .....	Security audit event storage (FAU_STG.1 & Extended – FAU_STG_EXT.1)	54
<b>Error! Hyperlink reference not valid.</b> A.2.1.1 .....	FAU_STG.1 Protected audit trail storage	54
<b>Error! Hyperlink reference not valid.</b> A.2.1.2 .....	FAU_STG_EXT.2 Counting lost audit data	54
<b>Error! Hyperlink reference not valid.</b> A.2.1.3 .....	FAU_STG_EXT.3 Display warning for local storage space	54
<b>Error! Hyperlink reference not valid.</b> A.3 .....	Security Management (FMT)	55
<b>Error! Hyperlink reference not valid.</b> A.3.1 .....	Management of functions in TSF (FMT_MOF)	55
<b>Error! Hyperlink reference not valid.</b> A.3.1.1 .....	FMT_MOF.1 Management of security functions behaviour	55
<b>Error! Hyperlink reference not valid.</b> A.3.2 .....	Management of TSF data (FMT_MTD)	56
<b>Error! Hyperlink reference not valid.</b> A.3.2.1 .....	FMT_MTD.1/AdminAet Management of TSF data	56
<b>Error! Hyperlink reference not valid.</b> A.4 .....	Protection of the TSF (FPT)	57
<b>Error! Hyperlink reference not valid.</b> A.4.1 .....	Fail Secure (FPT_FLS)	57
<b>Error! Hyperlink reference not valid.</b> A.4.1.1 .....	FPT_FLS.1/LocSpace Failure with preservation of secure state	57
<b>Error! Hyperlink reference not valid.</b> B .....	Selection-Based Requirements	58
<b>Error! Hyperlink reference not valid.</b> B.1 .....	Audit Events for Selection-Based SFRs	58
<b>Error! Hyperlink reference not valid.</b> B.2 .....	Cryptographic Support (FCS)	59
<b>Error! Hyperlink reference not valid.</b> B.2.1 .....	Cryptographic Protocols (Extended – FCS_HTTPS_EXT, FCS_IPSEC_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT)	59
<b>Error! Hyperlink reference not valid.</b> B.2.1.1 .....	FCS_HTTPS_EXT.1 HTTPS Protocol	59
<b>Error! Hyperlink reference not valid.</b> B.2.1.2 .....	FCS_IPSEC_EXT.1 IPsec Protocol	59
<b>Error! Hyperlink reference not valid.</b> B.2.1.3 .....	FCS_SSHC_EXT.1 SSH Client Protocol	64
<b>Error! Hyperlink reference not valid.</b> B.2.1.4 .....	FCS_SSHS_EXT.1 SSH Server Protocol	65
<b>Error! Hyperlink reference not valid.</b> B.2.1.5 .....	FCS_TLSC_EXT.1 TLS Client Protocol	67
<b>Error! Hyperlink reference not valid.</b> B.2.1.6 .....	FCS_TLSC_EXT.2 TLS Client Protocol with authentication	67
<b>Error! Hyperlink reference not valid.</b> B.2.1.7 .....	FCS_TLSS_EXT.1 TLS Server Protocol	71
<b>Error! Hyperlink reference not valid.</b> B.2.1.8 .....	FCS_TLSS_EXT.2 TLS Server Protocol with mutual authentication	71
<b>Error! Hyperlink reference not valid.</b> B.3 .....	Protection of the TSF (FPT)	74
<b>Error! Hyperlink reference not valid.</b> B.3.1 .....	TSF self test (Extended)	74
<b>Error! Hyperlink reference not valid.</b> B.3.1.1 .....	FPT_TST_EXT.2 Self tests based on certificates	74
<b>Error! Hyperlink reference not valid.</b> B.3.2 .....	Trusted Update (FPT_TUD_EXT)	75
<b>Error! Hyperlink reference not valid.</b> B.3.2.1 .....	FPT_TUD_EXT.2 Trusted Update based on certificates	75
<b>Error! Hyperlink reference not valid.</b> B.4 .....	Security Management (FMT)	75
<b>Error! Hyperlink reference not valid.</b> B.4.1 .....	Management of functions in TSF (FMT_MOF)	75
<b>Error! Hyperlink reference not valid.</b> B.4.1.1 .....	FMT_MOF.1(2)/TrustedUpdate – Management of security functions	75
<b>Error! Hyperlink reference not valid.</b> C .....	Extended Component Definitions	76
<b>Error! Hyperlink reference not valid.</b> C.1 .....	Security Audit (FAU)	76
<b>Error! Hyperlink reference not valid.</b> C.1.1 .....	Protected audit event storage (FAU_STG_EXT)	76
<b>Error! Hyperlink reference not valid.</b> C.1.1.1 .....	FAU_STG_EXT.1 Protected Audit Event Storage	77
<b>Error! Hyperlink reference not valid.</b> C.1.1.2 .....	FAU_STG_EXT.2 Counting lost audit data	77
<b>Error! Hyperlink reference not valid.</b> C.1.1.3 .....	FAU_STG_EXT.3 Display warning for local storage space	77
<b>Error! Hyperlink reference not valid.</b> C.2 .....	Cryptographic Support (FCS)	78
<b>Error! Hyperlink reference not valid.</b> C.2.1 .....	Random Bit Generation (FCS_RBG_EXT)	78
<b>Error! Hyperlink reference not valid.</b> C.2.1.1 .....	FCS_RBG_EXT.1 Random Bit Generation	78
<b>Error! Hyperlink reference not valid.</b> C.2.2 .....	Cryptographic Protocols (Extended – FCS_HTTPS_EXT, FCS_IPSEC_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT)	78
<b>Error! Hyperlink reference not valid.</b> C.2.2.1 .....	FCS_HTTPS_EXT.1 HTTPS Protocol	79
<b>Error! Hyperlink reference not valid.</b> C.2.2.2 .....	FCS_IPSEC_EXT.1 IPsec Protocol	80



<b>Error! Hyperlink reference not valid.</b>	<b>C.2.2.3</b>	<b>FCS_SSHC_EXT.1 SSH Client</b>	<b>85</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.2.2.4</b>	<b>FCS_SSHS_EXT.1 SSH Server Protocol</b>	<b>87</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.2.2.5</b>	<b>FCS_TLSC_EXT TLS Client Protocol</b>	<b>88</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.2.2.6</b>	<b>FCS_TLSS_EXT TLS Server Protocol</b>	<b>92</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3</b>	<b>Identification and Authentication (FIA)</b>	<b>95</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3.1</b>	<b>Password Management (FIA_PMG_EXT)</b>	<b>95</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3.1.1</b>	<b>FIA_PMG_EXT.1 Password Management</b>	<b>95</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3.2</b>	<b>User Identification and Authentication (FIA_UIA_EXT)</b>	<b>96</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3.2.1</b>	<b>FIA_UIA_EXT.1 User Identification and Authentication</b>	<b>96</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3.3</b>	<b>User authentication (FIA_UAU) (FIA_UAU_EXT)</b>	<b>97</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3.3.1</b>	<b>FIA_UAU_EXT.2 Password-based Authentication Mechanism</b>	<b>98</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3.4</b>	<b>Authentication using X.509 certificates (Extended FIA_X509_EXT)</b>	<b>98</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3.4.1</b>	<b>FIA_X509_EXT.1 X.509 Certificate Validation</b>	<b>98</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3.4.2</b>	<b>FIA_X509_EXT.2 X.509 Certificate Authentication</b>	<b>99</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.3.4.3</b>	<b>FIA_X509_EXT.3 X.509 Certificate Requests</b>	<b>100</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.4</b>	<b>Protection of the TSF (FPT)</b>	<b>100</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.4.1</b>	<b>Protection of TSF Data (FPT_SKP_EXT)</b>	<b>100</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.4.1.1</b>	<b>FPT_SKP_EXT.1 Protection of TSF Data (for reading of all sy</b>	<b>100</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.4.2</b>	<b>Protection of Administrator Passwords (FPT_APW_EXT)</b>	<b>102</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.4.2.1</b>	<b>FPT_APW_EXT.1 Protection of Administrator Passwords</b>	<b>102</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.4.3</b>	<b>TSF self test</b>	<b>102</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.4.3.1</b>	<b>FPT_TST_EXT.1 TSF Testing</b>	<b>102</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.4.4</b>	<b>Trusted Update (FPT_TUD_EXT)</b>	<b>104</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.4.4.1</b>	<b>FPT_TUD_EXT.1 Trusted Update</b>	<b>105</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.4.4.2</b>	<b>FPT_TUD_EXT.2 Trusted Update based on certificates</b>	<b>106</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.5</b>	<b>TOE Access (FTA)</b>	<b>107</b>
<b>Error! Hyperlink reference not valid.</b>	<b>C.5.1</b>	<b>FTA_SSL_EXT.1 TSF initiated Session Locking</b>	<b>107</b>
<b>Error! Hyperlink reference not valid.</b>	<b>D</b>	<b>Entropy Documentation And Assessment</b>	<b>109</b>
<b>Error! Hyperlink reference not valid.</b>	<b>D.1</b>	<b>Design Description</b>	<b>109</b>
<b>Error! Hyperlink reference not valid.</b>	<b>D.2</b>	<b>Entropy Justification</b>	<b>109</b>
<b>Error! Hyperlink reference not valid.</b>	<b>D.3</b>	<b>Operating Conditions</b>	<b>110</b>
<b>Error! Hyperlink reference not valid.</b>	<b>D.4</b>	<b>Health Testing</b>	<b>110</b>
<b>Error! Hyperlink reference not valid.</b>	<b>E</b>	<b>Glossary</b>	<b>111</b>
<b>Error! Hyperlink reference not valid.</b>	<b>F</b>	<b>Acronyms</b>	<b>112</b>
Acknowledgements			2
0. Preface			3
0.1 Objectives of Document			3
0.2 Scope of Document			3
0.3 Intended Readership			3
0.4 Related Documents			3
0.5 Revision History			5
1. PP Introduction			15
1.1 PP Reference Identification			15
1.2 TOE Overview			15
1.3 TOE Use Cases			16
2. CC Conformance			17
3. Introduction to Distributed TOEs			18
3.1 Supported Distributed TOE Use Cases			18
3.2 Unsupported Distributed TOE Use Cases			22
3.3 Registration of components of a distributed TOE			23
3.4 Allocation of Requirements in Distributed TOEs			26
4. Security Problem Definition			31
4.1 Threats			31
4.1.1 Communications with the Network Device			31
4.1.1.1 T.UNAUTHORIZED ADMINISTRATOR ACCESS			32

4.1.1.2	T.WEAK CRYPTOGRAPHY .....	32
4.1.1.3	T.UNTRUSTED COMMUNICATION CHANNELS .....	33
4.1.1.4	T.WEAK AUTHENTICATION ENDPOINTS .....	33
4.1.2	Valid Updates .....	34
4.1.2.1	T.UPDATE COMPROMISE .....	34
4.1.3	Audited Activity .....	34
4.1.3.1	T.UNDETECTED ACTIVITY .....	35
4.1.4	Administrator and Device Credentials and Data .....	35
4.1.4.1	T.SECURITY FUNCTIONALITY COMPROMISE .....	36
4.1.4.2	T.PASSWORD CRACKING .....	36
4.1.5	Device Failure .....	36
4.1.5.1	T.SECURITY FUNCTIONALITY FAILURE .....	37
4.2	Assumptions .....	37
4.2.1	A.PHYSICAL PROTECTION .....	37
4.2.2	A.LIMITED FUNCTIONALITY .....	37
4.2.3	A.NO THRU TRAFFIC PROTECTION .....	37
4.2.4	A.TRUSTED ADMINISTRATOR .....	38
4.2.5	A.REGULAR UPDATES .....	38
4.2.6	A.ADMIN CREDENTIALS SECURE .....	38
4.2.7	A.COMPONENTS RUNNING (applies to distributed TOEs only) .....	38
4.2.8	A.RESIDUAL INFORMATION .....	38
4.3	Organizational Security Policy .....	39
4.3.1	P.ACCESS BANNER .....	39
5.	Security Objectives .....	40
5.1	Security Objectives for the Operational Environment .....	40
5.1.1	OE.PHYSICAL .....	40
5.1.2	OE.NO GENERAL PURPOSE .....	40
5.1.3	OE.NO THRU TRAFFIC PROTECTION .....	40
5.1.4	OE.TRUSTED ADMIN .....	40
5.1.5	OE.UPDATES .....	40
5.1.6	OE.ADMIN CREDENTIALS SECURE .....	40
5.1.7	OE.COMPONENTS RUNNING (applies to distributed TOEs only) .....	40
5.1.8	OE.RESIDUAL INFORMATION .....	40
6.	Security Functional Requirements .....	42
6.1	Conventions .....	42
6.2	SFR Architecture .....	42
6.3	Security Audit (FAU) .....	50
6.3.1	Security Audit Data generation (FAU GEN) .....	50
6.3.1.1	FAU_GEN.1 Audit data generation .....	51
6.3.1.2	FAU_GEN.2 User identity association .....	55
6.3.2	Security audit event storage (Extended – FAU STG EXT) .....	55
6.3.2.1	FAU_STG_EXT.1 Protected Audit Event Storage .....	55
6.3.3	Security Audit for Distributed TOEs .....	56
6.4	Cryptographic Support (FCS) .....	57
6.4.1	Cryptographic Key Management (FCS CKM) .....	57
6.4.1.1	FCS_CKM.1 Cryptographic Key Generation (Refinement) .....	57
6.4.1.2	FCS_CKM.2 Cryptographic Key Establishment (Refinement) .....	58
6.4.1.3	FCS_CKM.4 Cryptographic Key Destruction .....	59
6.4.2	Cryptographic Operation (FCS COP) .....	60
6.4.2.1	FCS_COP.1 Cryptographic Operation .....	60
6.4.3	Random Bit Generation (Extended – FCS RBG EXT) .....	62
6.4.3.1	FCS_RBG_EXT.1 Random Bit Generation .....	62
6.5	Identification and Authentication (FIA) .....	63
6.5.1	Authentication Failure Management (FIA AFL) .....	63
6.5.1.1	FIA_AFL.1 Authentication Failure Management (Refinement) .....	63
6.5.2	Password Management (Extended – FIA PMG EXT) .....	64
6.5.2.1	FIA_PMG_EXT.1 Password Management .....	64

6.5.3	User Identification and Authentication (Extended – FIA_UIA_EXT).....	64
6.5.3.1	FIA_UIA_EXT.1 User Identification and Authentication.....	64
6.5.4	User authentication (FIA_UAU) (Extended – FIA_UAU_EXT).....	65
6.5.4.1	FIA_UAU_EXT.2 Password-based Authentication Mechanism.....	65
6.5.4.2	FIA_UAU.7 Protected Authentication Feedback .....	65
6.6	Security Management (FMT) .....	66
6.6.1	Management of functions in TSF (FMT_MOF) .....	68
6.6.1.1	FMT_MOF.1/ManualUpdate Management of security functions behaviour.....	68
6.6.2	Management of TSF Data (FMT_MTD).....	68
6.6.2.1	FMT_MTD.1/CoreData Management of TSF Data.....	68
6.6.3	Specification of Management Functions (FMT_SMF) .....	69
6.6.3.1	FMT_SMF.1 Specification of Management Functions .....	69
6.6.4	Security management roles (FMT_SMR).....	70
6.6.4.1	FMT_SMR.2 Restrictions on security roles .....	70
6.7	Protection of the TSF (FPT) .....	71
6.7.1	Protection of TSF Data (Extended – FPT_SKP_EXT) .....	71
6.7.1.1	FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys) ..	71
6.7.2	Protection of Administrator Passwords (Extended – FPT_APW_EXT).....	72
6.7.2.1	FPT_APW_EXT.1 Protection of Administrator Passwords .....	72
6.7.3	TSF testing (Extended – FPT_TST_EXT).....	72
6.7.3.1	FPT_TST_EXT.1 TSF Testing (Extended).....	72
6.7.4	Trusted Update (FPT_TUD_EXT).....	73
6.7.4.1	FPT_TUD_EXT.1 Trusted Update.....	73
6.7.5	Time stamps (FPT_STM).....	76
6.7.5.1	FPT_STM.1 Reliable Time Stamps .....	76
6.8	TOE Access (FTA).....	76
6.8.1	TSF-initiated Session Locking (Extended – FTA_SSL_EXT) .....	77
6.8.1.1	FTA_SSL_EXT.1 TSF-initiated Session Locking.....	77
6.8.2	Session locking and termination (FTA_SSL).....	77
6.8.2.1	FTA_SSL.3 TSF-initiated Termination (Refinement) .....	77
6.8.2.2	FTA_SSL.4 User-initiated Termination (Refinement) .....	77
6.8.3	TOE access banners (FTA_TAB) .....	77
6.8.3.1	FTA_TAB.1 Default TOE Access Banners (Refinement).....	77
6.9	Trusted path/channels (FTP).....	78
6.9.1	Trusted Channel (FTP_ITC) .....	78
6.9.1.1	FTP_ITC.1 Inter-TSF trusted channel (Refinement) .....	78
6.9.2	Trusted Path (FTP_TRP).....	79
6.9.2.1	FTP_TRP.1/Admin Trusted Path (Refinement).....	79
7.	Security Assurance Requirements.....	81
7.1	ASE: Security Target.....	81
7.2	ADV: Development .....	82
7.2.1	Basic Functional Specification (ADV_FSP.1) .....	82
7.3	AGD: Guidance Documentation.....	82
7.3.1	Operational User Guidance (AGD_OPE.1) .....	83
7.3.2	Preparative Procedures (AGD_PRE.1) .....	83
7.4	Class ALC: Life-cycle Support.....	85
7.4.1	Labelling of the TOE (ALC_CMC.1) .....	85
7.4.2	TOE CM Coverage (ALC_CMS.1).....	85
7.5	Class ATE: Tests .....	85
7.5.1	Independent Testing – Conformance (ATE_IND.1) .....	85
7.6	Class AVA: Vulnerability Assessment .....	85
7.6.1	Vulnerability Survey (AVA_VAN.1) .....	86
A.	Optional Requirements .....	87
A.1	Audit Events for Optional SFRs .....	87
A.2	Security Audit (FAU) .....	89
A.2.1	Security audit event storage (FAU_STG.1 & Extended – FAU_STG_EXT) .....	89
A.2.1.1	FAU_STG.1 Protected audit trail storage .....	89

A.2.1.2	FAU_STG_EXT.2/LocSpace Counting lost audit data .....	89
A.2.1.3	FAU_STG.3/LocSpace Action in case of possible audit data loss Display warning for local storage space .....	90
A.3	Identification and Authentication (FIA) .....	91
A.1.1	Authentication using X.509 certificates (Extended – FIA_X509_EXT).....	91
A.3.1.1	FIA_X509_EXT.1 Certificate Validation.....	91
A.4	Security Management (FMT) .....	92
A.4.1	Management of functions in TSF (FMT_MOF) .....	92
A.4.1.1	FMT_MOF.1/Services Management of security functions behaviour.....	92
A.4.1.2	FMT_MOF.1/Functions Management of security functions behaviour .....	92
A.4.2	Management of TSF data (FMT_MTD) .....	94
A.4.2.1	FMT_MTD.1/CryptoKeys Management of TSF data .....	94
A.5	Protection of the TSF (FPT) .....	94
A.5.1	Fail Secure (FPT_FLS) .....	94
A.5.1.1	FPT_FLS.1/LocSpace Failure with preservation of secure state .....	94
A.5.2	Internal TOE TSF data transfer (FPT_ITT) .....	95
A.5.2.1	FPT_ITT.1 Basic internal TSF data transfer protection (Refinement) .....	95
A.6	Trusted Path/Channels (FTP).....	95
A.6.1	Trusted Path (FTP_TRP).....	95
A.6.1.1	FTP_TRP.1/Join Trusted Path (Refinement) .....	95
A.7	Communication (FCO).....	96
A.7.1	Communication Partner Control (FCO_CPC_EXT).....	96
A.7.1.1	FCO_CPC_EXT.1 Component Registration Channel Definition.....	97
B.	Selection-Based Requirements.....	99
B.1	Audit Events for Selection-Based SFRs .....	99
B.2	Cryptographic Support (FCS).....	100
B.2.1	Cryptographic Protocols (Extended – FCS_HTTPS_EXT, FCS_IPSEC_EXT, FCS_SSHC_EXT, FCS_SSHS_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT) .....	100
B.2.1.1	FCS_HTTPS_EXT.1 HTTPS Protocol.....	100
B.2.1.2	FCS_IPSEC_EXT.1 IPsec Protocol.....	101
B.2.1.3	FCS_SSHC_EXT.1 SSH Client Protocol .....	106
B.2.1.4	FCS_SSHS_EXT.1 SSH Server Protocol.....	108
B.2.1.5	FCS_TLSC_EXT.1 TLS Client Protocol.....	110
B.2.1.6	FCS_TLSC_EXT.2 TLS Client Protocol with authentication .....	112
B.2.1.7	FCS_TLSS_EXT.1 TLS Server Protocol .....	115
B.2.1.8	FCS_TLSS_EXT.2 TLS Server Protocol with mutual authentication.....	117
B.3	Identification and Authentication (FIA) .....	119
B.3.1	Authentication using X.509 certificates (Extended – FIA_X509_EXT).....	119
B.3.1.1	FIA_X509_EXT.1 X.509 Certificate Validation .....	119
B.3.1.2	FIA_X509_EXT.2 X.509 Certificate Authentication .....	120
B.3.1.3	FIA_X509_EXT.3 X.509 Certificate Requests .....	121
B.4	Protection of the TSF (FPT) .....	121
B.4.1	TSF self test (Extended).....	121
B.4.1.1	FPT_TST_EXT.2 Self tests based on certificates.....	121
B.4.2	Trusted Update (FPT_TUD_EXT).....	122
B.4.2.1	FPT_TUD_EXT.2 Trusted Update based on certificates.....	122
B.5	Security Management (FMT) .....	122
B.5.1	Management of functions in TSF (FMT_MOF) .....	122
B.5.1.1	FMT_MOF.1/AutoUpdate Management of security functions behaviour.....	122
C.	Extended Component Definitions.....	124
C.1	Security Audit (FAU) .....	124
C.1.1	Protected audit event storage (FAU_STG_EXT).....	124
C.1.1.1	FAU_STG_EXT.1 Protected Audit Event Storage.....	125
C.1.1.2	FAU_STG_EXT.2 Counting lost audit data .....	126
C.2	Cryptographic Support (FCS) .....	127
C.2.1	Random Bit Generation (FCS_RBG_EXT).....	127
C.2.1.1	FCS_RBG_EXT.1 Random Bit Generation .....	127

C.2.2	Cryptographic Protocols (Extended – FCS HTTPS_EXT, FCS IPSEC_EXT, FCS SSHC_EXT, FCS SSHS_EXT, FCS TLSC_EXT, FCS TLSS_EXT) .....	128
C.2.2.1	FCS HTTPS_EXT.1 HTTPS Protocol.....	128
C.2.2.2	FCS IPSEC_EXT.1 IPsec Protocol.....	129
C.2.2.3	FCS SSHC_EXT.1 SSH Client.....	133
C.2.2.4	FCS SSHS_EXT.1 SSH Server Protocol .....	136
C.2.2.5	FCS TLSC_EXT TLS Client Protocol.....	138
C.2.2.6	FCS TLSS_EXT TLS Server Protocol .....	142
C.3	Identification and Authentication (FIA) .....	146
C.3.1	Password Management (FIA_PMG_EXT) .....	146
C.3.1.1	FIA_PMG_EXT.1 Password Management.....	146
C.3.2	User Identification and Authentication (FIA_UIA_EXT).....	147
C.3.2.1	FIA_UIA_EXT.1 User Identification and Authentication .....	148
C.3.3	User authentication (FIA_UAU) (FIA_UAU_EXT).....	148
C.3.3.1	FIA_UAU_EXT.2 Password-based Authentication Mechanism.....	149
C.3.4	Authentication using X.509 certificates (Extended – FIA_X509_EXT).....	150
C.3.4.1	FIA_X509_EXT.1 X.509 Certificate Validation .....	151
C.3.4.2	FIA_X509_EXT.2 X509 Certificate Authentication .....	152
C.3.4.3	FIA_X509_EXT.3 X.509 Certificate Requests .....	152
C.4	Protection of the TSF (FPT) .....	153
C.4.1	Protection of TSF Data (FPT_SKP_EXT) .....	153
C.4.1.1	FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys) 154	
C.4.2	Protection of Administrator Passwords (FPT_APW_EXT).....	155
C.4.2.1	FPT_APW_EXT.1 Protection of Administrator Passwords .....	155
C.4.3	TSF self test .....	155
C.4.3.1	FPT_TST_EXT.1 TSF Testing .....	155
C.4.4	Trusted Update (FPT_TUD_EXT).....	157
C.4.4.1	FPT_TUD_EXT.1 Trusted Update .....	158
C.4.4.2	FPT_TUD_EXT.2 Trusted Update based on certificates.....	161
C.5	TOE Access (FTA).....	161
C.5.1	FTA_SSL_EXT.1 TSF-initiated Session Locking .....	161
C.6	Communication (FCO) .....	163
C.6.1	Communication Partner Control (FCO_CPC_EXT).....	163
A.1.1.1.1	FCO_CPC_EXT.1 Component Registration Channel Definition.....	163
D.	Entropy Documentation And Assessment.....	165
D.1	Design Description .....	165
D.2	Entropy Justification .....	165
D.3	Operating Conditions.....	166
D.4	Health Testing.....	166
E.	Rationales.....	167
E.1	SFR Dependencies Analysis.....	167
	Glossary .....	175
	Acronyms .....	176

## Figures / Tables

Figure 1: <u>Generalized Distributed TOE Model</u> .....	18
Figure 2: <u>Basic distributed TOE use case</u> .....	19
Figure 3: <u>Non-distributed TOE use case</u> .....	19
Figure 4: <u>Distributed TOE use case with Management Component out of scope</u> .....	20
Figure 5: <u>Management Component required to fulfil cPP requirements</u> .....	20
Figure 6: <u>Distributed Network Devices plus Management Component required to fulfil cPP requirements</u> .....	21
Figure 7 <u>Distributed TOE extended through equivalency argument</u> .....	21
Figure 8 <u>Unsupported Enterprise Management use case</u> .....	22
Figure 9 <u>Unsupported use case with Multiple Management Components</u> .....	23
Figure 10 <u>Distributed TOE registration using channel satisfying FPT ITT.1 or FTP ITC.1</u> .....	24
Figure 11 <u>Distributed TOE registration using channel satisfying FTP TRP.1/Join</u> .....	24
Figure 12 <u>Distributed TOE registration without a registration channel</u> .....	25
Figure 13 <u>Joiner enablement options for Distributed TOEs</u> .....	25
Table 1: <u>Security Functional Requirements for Distributed TOEs</u> .....	30
<u>Figure 14: Protected Communications SFR Architecture</u> .....	<u>444</u>
Figure <u>215: Administrator Authentication SFR Architecture</u> .....	<u>4542</u>
Figure <u>316: Correct Operation SFR Architecture</u> .....	<u>4642</u>
Figure <u>417: Trusted Updated and Audit SFR Architecture</u> .....	<u>4843</u>
Figure <u>518: Management SFR Architecture</u> .....	<u>4944</u>
<u>Figure 19: Distributed TOE SFR Architecture</u> .....	<u>50</u>
Table <u>42: Security Functional Requirements and Auditable Events</u> .....	<u>5448</u>
Table <u>23: Security Assurance Requirements</u> .....	<u>8172</u>
Table <u>34: TOE Optional SFRs and Auditable Events</u> .....	<u>8878</u>
Table <u>45: Selection-Based SFRs and Auditable Events</u> .....	<u>10059100</u>
Table 6: <u>SFR Dependencies Rationale for Mandatory SFRs</u> .....	170
Table 7: <u>SFR Dependencies Rationale for Optional SFRs</u> .....	171
Table 8: <u>SFR Dependencies Rationale for Selection-Based SFRs</u> .....	174

# 1. PP Introduction

## 1.1 PP Reference Identification

PP Reference: collaborative Protection Profile for Network Devices

PP Version: 1.01

PP Date: ~~27-Feb-2015~~21-Jul-2016

## 1.2 TOE Overview

This is a Collaborative Protection Profile (cPP) whose Target of Evaluation (TOE) is a network device. It provides a minimal set of security requirements expected by all network devices that target the mitigation of a set of defined threats. This baseline set of requirements will be built upon by future cPPs to provide an overall set of security solutions for networks up to carrier and enterprise scale. A network device in the context of this cPP is a device composed of both hardware and software that is connected to the network and has an infrastructure role within the network. The TOE may be standalone or distributed, where a distributed TOE is one that requires multiple distinct components to operate as a logical whole in order to fulfill the requirements of this cPP (a more extensive description of distributed network device TOEs is given in section 3).

A Virtual Network Device (vND) is a software implementation of network device functionality that run inside a virtual machine. This cPP expressly excludes evaluation of vNDs unless the products is able to meet all the requirements and assumptions of a physical ND as required in this cPP

This means:

- The virtualization layer (or hypervisor or Virtual Machine Manager (VMM)) is considered part of the ND's software stack, and thus is part of the TOE. vNDs that can run on multiple VMMs must be tested on each claimed VMM unless the vendor can successfully argue equivalence.
- The physical hardware is likewise included in the TOE (like in the example included above). vNDs must be tested for each claimed hardware platform unless the vendor can successfully argue equivalence.
- There is only one vND instance for each physical hardware platform.
- There are no other guest VMs on the physical platform providing non-network device functionality.

The intent of this document is to define the baseline set of common security functionality expected by all network devices, regardless of their ultimate security purpose or any additional security functionality the device may employ. This baseline set includes securing any remote management path, providing identification and authentication services for both



local and remote logins, auditing security-related events, cryptographically validating the source of any update, and offering some protection against common network-based attacks.

The aim is that any network device that meets this cPP will “behave well” on the network and can be trusted to do no harm. To accomplish this, the network device is expected to employ standards-based tunneling protocols to include IPsec, TLS, or SSH to protect the communication paths to external entities.~~It, and in the case of a distributed TOE, to protect the communications between TOE components. For most of the allowed secure channel protocol selections it~~ is also required that X.509 certificates be used for authentication purposes; use of certificates is supported as an option for code signing/digital signatures.

~~Distributed TOEs are outside the scope of the current version of this cPP, but are expected to be included in the scope of the next version.~~

Additional security functionality that a network device may employ is outside the scope of this cPP, ~~as these and such functionality~~ will be specified in other device-type specific cPPs. Also considered out of scope is virus and emailing scanning, intrusion detection/prevention capabilities, Network Address Translation (NAT) as a security function, and virtualized network functions~~, except in the case outlined above.~~ It is expected that this cPP will be updated to expand the desired security functionality to increase resiliency, allow for varying implementations (such as software-only network devices), and keep current with technology enhancements. At this time, however, Exact ~~Compliance~~Conformance<sup>2</sup> with the cPP is required, and no additional functionality will be evaluated.

### 1.3 TOE Use Cases

The essence of the requirements for network device TOEs is that the devices can be remotely managed in a secure manner and that any software updates applied are from a trusted source.

Examples of network devices that are covered by requirements in this cPP include routers, firewalls, VPN gateways, IDSs, and switches. Where such devices include significant additional functionality with its own distinct security requirements then a separate cPP may be created to be used for those devices, with that cPP containing a superset of the network device cPP requirements. For example, a separate cPP of this sort has been created for Stateful Traffic Filter Firewalls.

Examples of devices that connect to a network but are not included to be evaluated against this cPP include mobile devices, end-user workstations, and virtualized network device functionality.

---

<sup>2</sup> Exact ~~Compliance~~Conformance is specified as a subset of ~~strict-compliance~~Strict Conformance – see the definition in section 22.



## 2. CC Conformance

As defined by the references [CC1], [CC2] and [CC3], this cPP:

- conforms to the requirements of Common Criteria v3.1, Revision 4
- is Part 2 extended, Part 3 conformant
- does not claim conformance to any other PP.

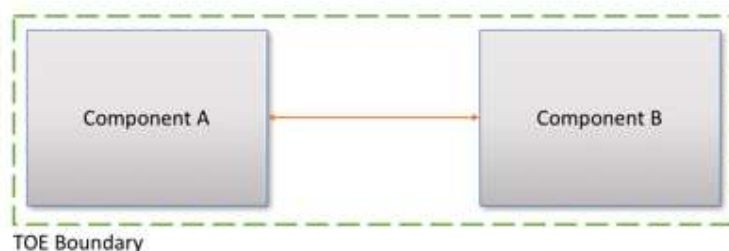
The methodology applied for the cPP evaluation is defined in [CEM]. This cPP satisfies the following Assurance Families: APE\_CCL.1, APE\_ECD.1, APE\_INT.1, APE\_OBJ.1, APE\_REQ.1 and APE\_SPD.1.

In order to be conformant to this cPP, a TOE must demonstrate Exact Conformance. Exact Conformance, as a subset of Strict Conformance as defined by the CC, is defined as the ST containing all of the requirements in section ~~56~~ (these are the mandatory requirements) of the this cPP, and potentially requirements from Appendix ~~AA~~ (these are optional SFRs) or Appendix ~~BB~~ (these are selection-based SFRs, some of which will be mandatory according to the selections made in other SFRs) of this cPP. While iteration is allowed, no additional requirements (from the CC parts 2 or 3, or definitions of extended components not already included in this cPP) are allowed to be included in the ST. Further, no requirements in section ~~56~~ of this cPP are allowed to be omitted.

### **3. Introduction to Distributed TOEs**

This cPP includes support for distributed network device TOEs. Network devices can sometimes be composed of multiple components operating as a logical whole. Oftentimes we see this architecture when dealing with products where a centralized management console is used to provide administration to dispersed components.

There are a number of different architectures, but fundamentally, they are variations of the following model where the SFRs of this cPP can only be fulfilled if the two components are deployed and operate together.

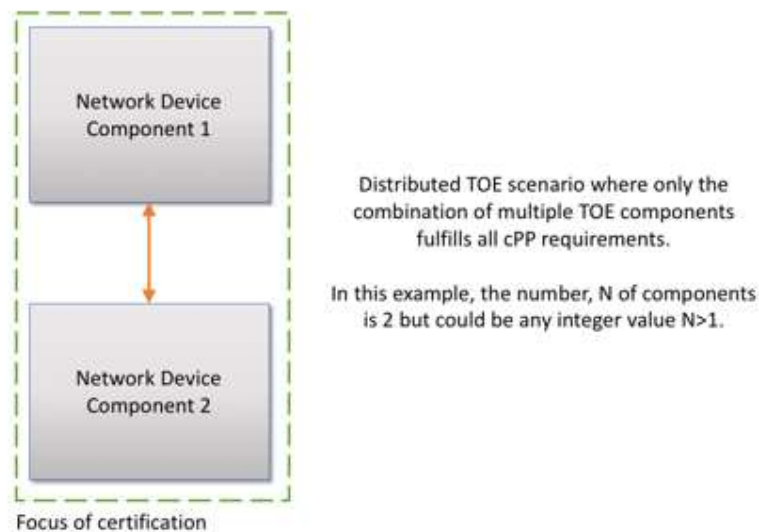


*Figure 1: Generalized Distributed TOE Model*

#### **3.1 Supported Distributed TOE Use Cases**

The following discussion provides guidance over the supported distributed TOE use cases in this version of the cPP.

**Case 1: cPP requirements can only be fulfilled if several TOE components work together**

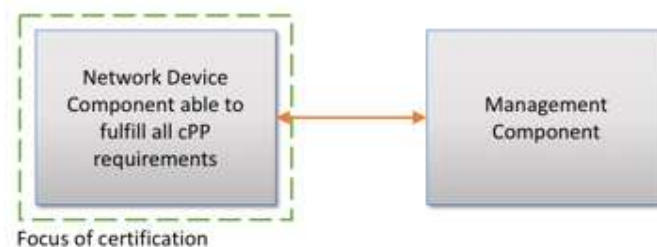


*Figure 2: Basic distributed TOE use case*

The first and most basic use case is where multiple interconnected network device components need to operate together to fulfil the requirements of the cPP. To be considered a distributed TOE, a minimum of 2 interconnected components are required.

**Case 2: cPP requirements can be fulfilled without Management component.**

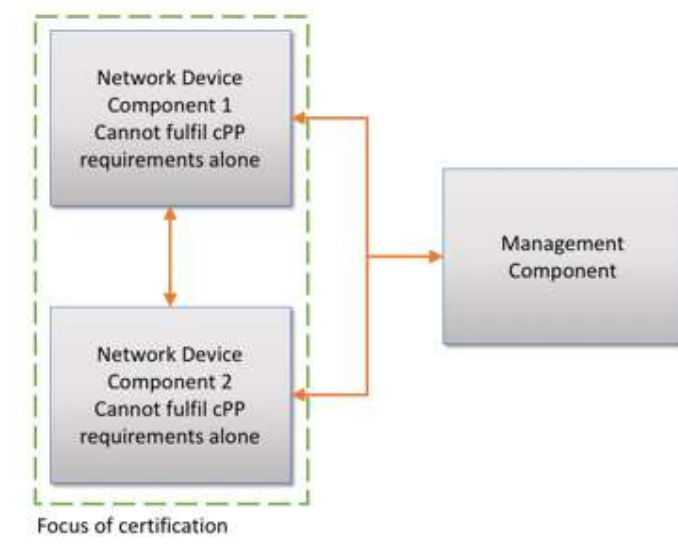
Some network devices are designed to operate alongside a Management Component. A network device that operates in this manner but can satisfy all SFRs of the cPP without the Management Component shall not be regarded as a distributed TOE and shall be certified according to this cPP without the Management Component



*Figure 3: Non-distributed TOE use case*

Alternatively, a Network Device may require more than one component in order to fulfil all of the requirements of the cPP. In addition to the components required to fulfil the cPP a Management Component may also be offered for use with the TOE. However, as with the

case shown in Figure 3 above, certification shall not include the Management Component in this case. This situation is depicted in Figure 4.

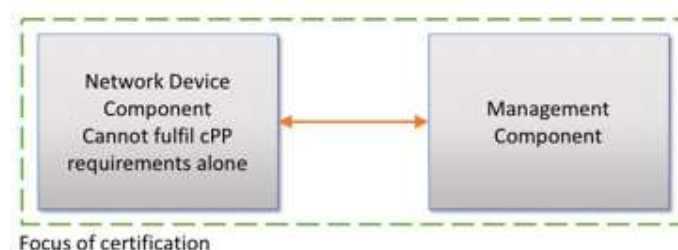


*Figure 4: Distributed TOE use case with Management Component out of scope*

For the cases in both Figure 3 and Figure 4, the Management Component may be certified separately according to a different (c)PP.

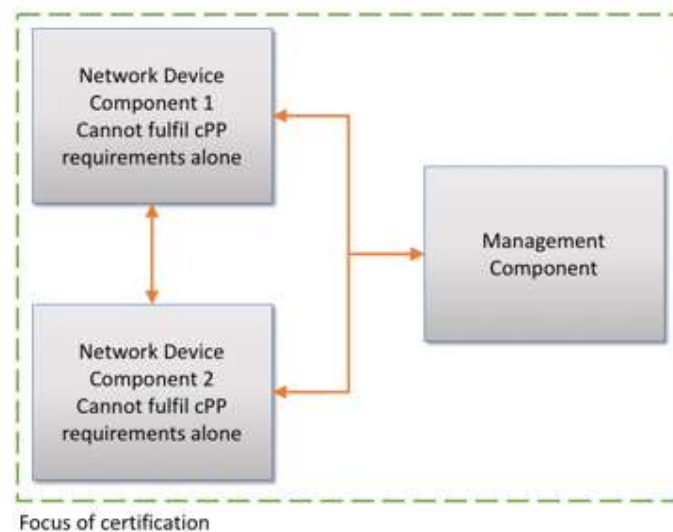
### **Case 3: cPP requirements cannot be fulfilled without Management Component**

A Network Device that requires the Management Component to satisfy all SFRs of the cPP shall be considered to be a distributed TOE and be certified according to this cPP together with the Management Component.



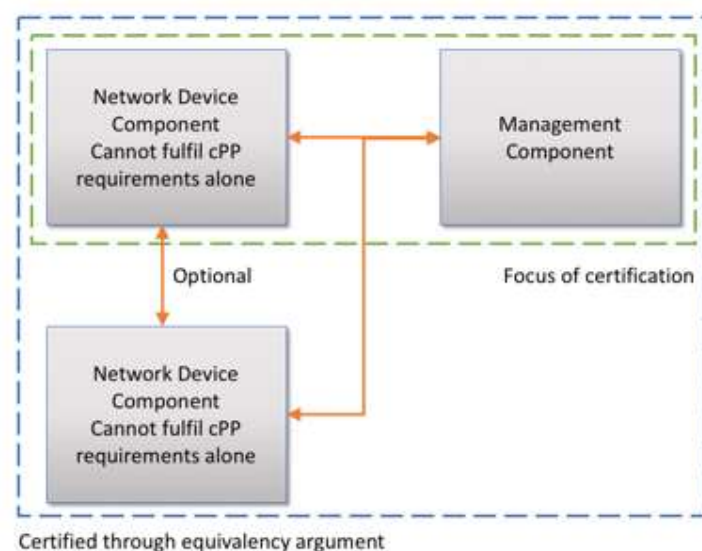
*Figure 5: Management Component required to fulfil cPP requirements*

A Management Component may also be considered part of the distributed TOE alongside multiple distributed Network Devices if it is required to fulfil all SFRs of this cPP.



*Figure 6: Distributed Network Devices plus Management Component required to fulfil cPP requirements*

Where several Network Devices are managed by one Management Component, the TOE may also be considered to be distributed but the focus of the certification should be restricted to the simplest combination of Network Device and Management Component. By the use of an equivalency argument, the combination of multiple Network Devices together with one Management Component can then be regarded as certified solution<sup>3</sup>.



*Figure 7 Distributed TOE extended through equivalency argument*

<sup>3</sup> [SD, B.4] describes how to define the components of a distributed TOE in terms of a ‘minimum configuration’ and allowance for iteration of equivalent components.

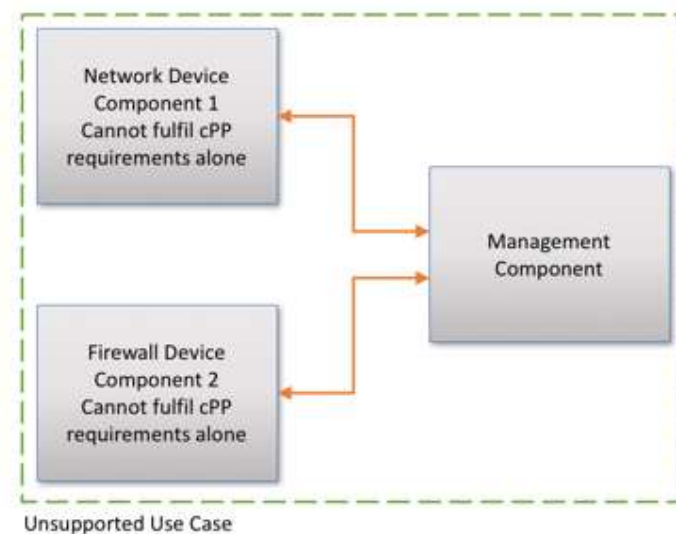
In this model the individual Network Device components rely on functionality within the Management Component to fulfil the requirements of this cPP and therefore a direct relationship between Network Device components themselves is optional.

More than one Management Component may be used if it is for the sole purpose of redundancy.

### **3.2 Unsupported Distributed TOE Use Cases**

The following discussion provides guidance for the distributed TOE use cases that are not supported by this version of the cPP.

#### **Case 4: cPP requirements cannot be fulfilled without Management Component shared with other devices**

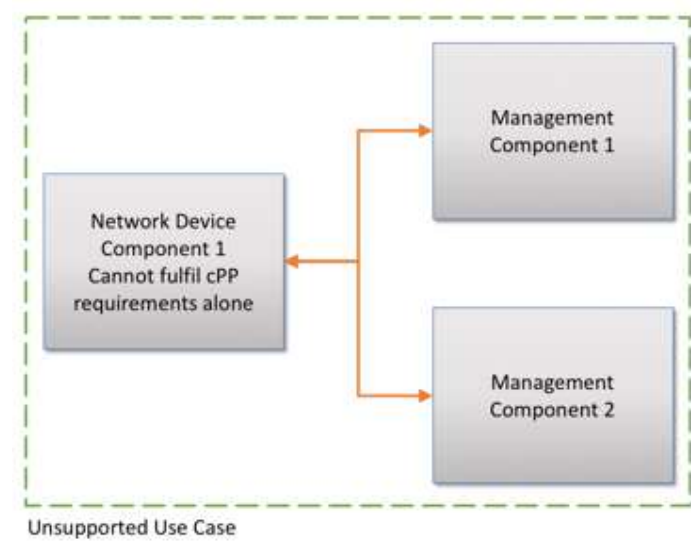


*Figure 8 Unsupported Enterprise Management use case*

This case is similar to Case 3 above but consists of different types of devices that form multiple distinct products, together with one Management Component. In this case the Management Component is considered to be an ‘Enterprise Manager’ (central management components for different types of devices). This use case is not supported by this version of the cPP.

#### **Case 5: cPP requirements cannot be fulfilled without multiple Management Components**

The case where one device, distributed TOE or combination of TOEs according to Case 3 above are managed by more than one Management Component (except for the purpose of redundancy) is not covered by this version of the cPP.

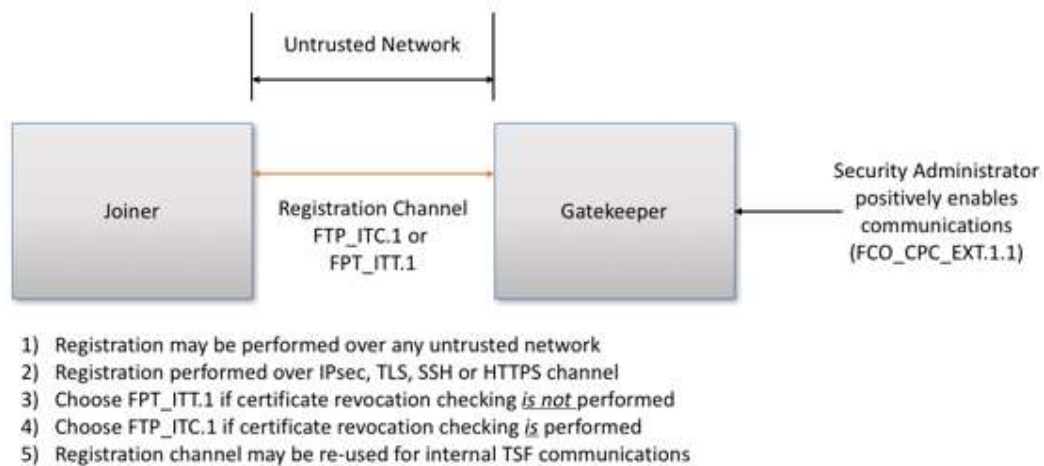


*Figure 9 Unsupported use case with Multiple Management Components*

### **3.3 Registration of components of a distributed TOE**

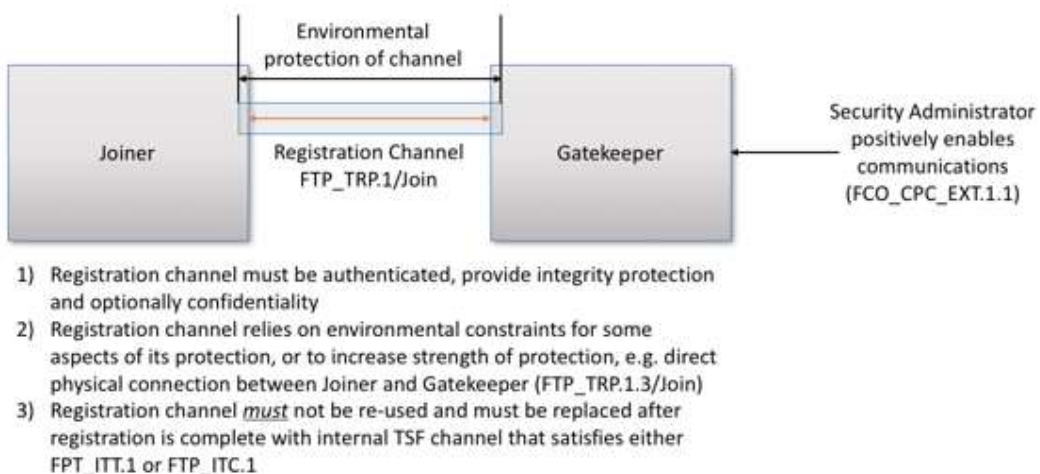
When dealing with a distributed TOE, a number of separate components need to be brought together in the operational environment in order to create the TOE: this requires that trusted communications channels are set up between certain pairs of components (it is assumed that all components need to communicate with at least one other component, but not that all components need to communicate with all other components).

The underlying model for creation of the TOE is to have a 'registration process' in which components 'join' the TOE. The registration process starts with two components, one of which (the 'joiner') is about to join an existing TOE by registering with the other (the 'gatekeeper'). The two components will use one or more specified authentication and communication channel options so that the components authenticate each other and protect any sensitive data that is transmitted during the registration process (e.g. a key might be sent by a gatekeeper to the joiner as a result of the registration). The following figures illustrate the three supported registration models. Figure 10 illustrates a distributed TOE registration approach which uses an instance of FPT ITT.1 or FTP ITC.1 to protect the registration exchange.



*Figure 10 Distributed TOE registration using channel satisfying FPT\_ITT.1 or FTP\_ITC.1*

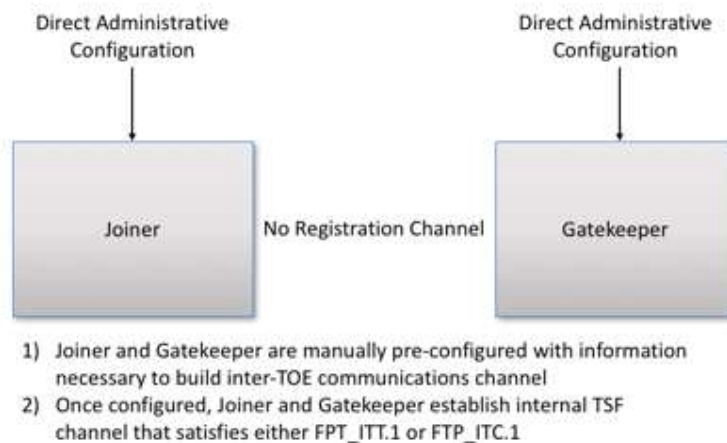
The second approach (Figure 11) utilises an alternative registration channel and supports use-cases where the channel relies on environmental security constraints to provide the necessary protection of the registration exchange.



*Figure 11 Distributed TOE registration using channel satisfying FTP\_TRP.1/Join*

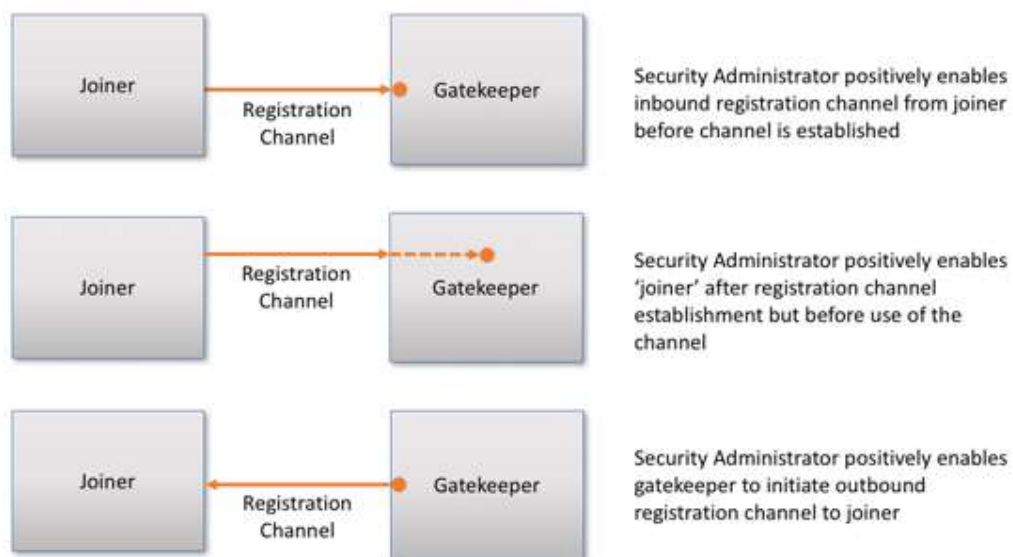
The final approach (Figure 12) supports use-cases where registration is performed manually through direct configuration of both the Joiner and Gatekeeper devices. Once configured, the two components establish an internal TSF channel that satisfies FPT\_ITT.1 or FTP\_ITC.1.





*Figure 12 Distributed TOE registration without a registration channel*

In each case, during the registration process, the Security Administrator must positively enable the joining components before it can act as part of the TSF. The following figure illustrates the approaches that this enablement step may take;



*Figure 13 Joiner enablement options for Distributed TOEs*

Note that in the case where no registration channel is required, that is that the joiner and gatekeeper are directly configured (Figure 12), enablement is implied as part of this direct configuration process.

After registration the components will communicate between themselves using a normal SSH/TLS/IPsec/HTTPS channel (which is specified in an ST as an instance of FTP\_ITC.1 or FPT\_ITT.1 in terms of section 6 and appendix A). This channel for inter-component communications is specified at the top level with the new (extended) SFR FCO\_CPC\_EXT.1 (see section A.7.1) and is in addition to the other communication channels required for

communication with entities outside the TOE (which are specified in an ST as instances of FTP ITC.1 and FPT TRP.1).

### **3.4 Allocation of Requirements in Distributed TOEs**

For a distributed TOE, the security functional requirements in this cPP need to be met by the TOE as a whole, but not all SFRs will necessarily be implemented by all components. The following categories are defined in order to specify when each SFR must be implemented by a component:

- **All Components ('All')** – All components that comprise the distributed TOE must independently satisfy the requirement.
- **At least one Component ('One')** – This requirement must be fulfilled by at least one component within the distributed TOE.
- **Feature Dependent ('Feature Dependent')** – These requirements will only be fulfilled where the feature is implemented by the distributed TOE component (note that the requirement to meet the cPP as a whole requires that at least one component implements these requirements if they are specified in section 6).

Table 1 specifies how each of the SFRs in this cPP must be met, using the categories above.

<u>Requirement</u>	<u>Description</u>	<u>Distributed TOE SFR Allocation</u>
<u>FAU_GEN.1</u>	<u>Audit Data Generation</u>	<u>All</u>
<u>FAU_GEN.2</u>	<u>User Identity Association</u>	<u>All</u>
<u>FAU_STG_EXT.1</u>	<u>Protected Audit Event Storage</u>	<u>All</u>
<u>FAU_STG.1</u>	<u>Protected Audit Trail Storage</u>	<u>One</u>
<u>FAU_STG_EXT.2/LocSpace</u>	<u>Counting Lost Audit Data</u>	<u>One</u>
<u>FAU_STG.3/LocSpace</u>	<u>Display warning for local storage space</u>	<u>One</u>

<u>Requirement</u>	<u>Description</u>	<u>Distributed TOE SFR Allocation</u>
<u>FCO_CPC_EXT.1</u>	<u>Communication Partner Control</u>	<u>One</u>
<u>FCS_CKM.1</u>	<u>Cryptographic Key Generation</u>	<u>One<sup>4</sup></u>
<u>FCS_CKM.2</u>	<u>Cryptographic Key Establishment</u>	<u>All</u>
<u>FCS_CKM.4</u>	<u>Cryptographic Key Destruction</u>	<u>All</u>
<u>FCS_COP.1/DataEncryption</u>	<u>Cryptographic Operation (AES Data Encryption/Decryption)</u>	<u>All</u>
<u>FCS_COP.1/SigGen</u>	<u>Cryptographic Operation (Signature Verification)</u>	<u>All</u>
<u>FCS_COP.1/Hash</u>	<u>Cryptographic Operation (Hash Algorithm)</u>	<u>All</u>
<u>FCS_COP.1/KeyedHash</u>	<u>Cryptographic Operation (Keyed Hash Algorithm)</u>	<u>All</u>
<u>FCS_HTTPS_EXT.1</u>	<u>HTTPS Protocol</u>	<u>Feature Dependent</u>
<u>FCS_IPSEC_EXT.1</u>	<u>IPsec Protocol</u>	<u>Feature Dependent</u>
<u>FCS_SSHC_EXT.1</u>	<u>SSH Client</u>	<u>Feature Dependent</u>
<u>FCS_SSHS_EXT.1</u>	<u>SSH Server</u>	<u>Feature Dependent</u>
<u>FCS_TLSC_EXT.1</u>	<u>TLS Client</u>	<u>Feature Dependent</u>

<sup>4</sup> Each component of a distributed TOE will be required either to perform on-board key generation and (if the TOE uses X.509 certificates as in Appendix B.3.1) RFC 2986 Certificate Request generation, or else to receive its keys and certificates, generated on some other component of the TOE, using a secure registration channel at the point where the component is joined to the TOE. Certificate request generation will be required from either the component that generates the key or the component that receives the key.

<u>Requirement</u>	<u>Description</u>	<u>Distributed TOE SFR Allocation</u>
<u>FCS_TLSC_EXT.2</u>	<u>TLS Client with authentication</u>	<u>Feature Dependent</u>
<u>FCS_TLSS_EXT.1</u>	<u>TLS Server</u>	<u>Feature Dependent</u>
<u>FCS_TLSS_EXT.2</u>	<u>TLS Server with mutual authentication</u>	<u>Feature Dependent</u>
<u>FCS_RBG_EXT.1</u>	<u>Random Bit Generation</u>	<u>All</u>
<u>FIA_AFL.1</u>	<u>Authentication Failure Management</u>	<u>One</u>
<u>FIA_PMG_EXT.1</u>	<u>Password Management</u>	<u>One</u>
<u>FIA_UIA_EXT.1</u>	<u>User Identification and Authentication</u>	<u>One</u>
<u>FIA_UAU_EXT.2</u>	<u>Password-based Authentication Mechanism</u>	<u>One</u>
<u>FIA_UAU.7</u>	<u>Protected Authentication Feedback</u>	<u>Feature Dependent</u>
<u>FIA_X509_EXT.1/Rev</u>	<u>X.509 Certification Validation</u>	<u>Feature Dependent</u>
<u>FIA_X509_EXT.1/ITT</u>	<u>X.509 Certification Validation</u>	<u>Feature Dependent</u>
<u>FIA_X509_EXT.2</u>	<u>X.509 Certificate Authentication</u>	<u>Feature Dependent</u>
<u>FIA_X509_EXT.3</u>	<u>Certificate Requests</u>	<u>Feature Dependent<sup>4</sup></u>
<u>FMT_MOF.1/ManualUpdate</u>	<u>Trusted Update - Management of Security Functions behaviour</u>	<u>All</u>
<u>FMT_MOF.1/Services</u>	<u>Trusted Update - Management of TSF Data</u>	<u>All</u>

<u>Requirement</u>	<u>Description</u>	<u>Distributed TOE SFR Allocation</u>
<u>FMT MOF.1/Functions</u>	<u>Management of security functions behaviour</u>	<u>All</u>
<u>FMT MTD.1/CoreData</u>	<u>Management of TSF Data</u>	<u>All</u>
<u>FMT MTD.1/CryptoKeys</u>	<u>Management of TSF Data</u>	<u>Feature Dependent</u>
<u>FMT SMF.1</u>	<u>Specification of Management Functions</u>	<u>Feature Dependent</u>
<u>FMT SMR.2</u>	<u>Restrictions on Security Roles</u>	<u>One</u>
<u>FPT SKP_EXT.1</u>	<u>Protection of TSF Data (for reading of all symmetric keys)</u>	<u>All</u>
<u>FPT APW_EXT.1</u>	<u>Protection of Administrator Passwords</u>	<u>Feature Dependent</u>
<u>FPT TST_EXT.1</u>	<u>Testing (Extended)</u>	<u>All</u>
<u>FPT_FLS.1/LocSpace</u>	<u>Failure with preservation of secure state</u>	<u>All</u>
<u>FPT ITT.1</u>	<u>Basic internal TSF data transfer protection</u>	<u>Feature Dependent<sup>5</sup></u>
<u>FPT STM.1</u>	<u>Reliable Time Stamps</u>	<u>All</u>
<u>FPT TST_EXT.2</u>	<u>Self-Test Based on Certificates</u>	<u>Feature Dependent</u>
<u>FPT TUD_EXT.1</u>	<u>Trusted Update</u>	<u>All</u>

<sup>5</sup> To protect inter-TSF data transfer, FPT ITT.1 or FTP ITC.1 must be fulfilled by each distributed TOE component. This is in addition to an iteration of FTP ITC.1 to protect communications with external entities.

<u>Requirement</u>	<u>Description</u>	<u>Distributed TOE SFR Allocation</u>
<u>FPT_TUD_EXT.2</u>	<u>Trusted Update based on Certificates</u>	<u>Feature Dependent</u>
<u>FTA_SSL.3</u>	<u>TSF-initiated Termination</u>	<u>Feature Dependent</u>
<u>FTA_SSL.4</u>	<u>User-Initiated Termination</u>	<u>Feature Dependent</u>
<u>FTA_SSL_EXT.1</u>	<u>TSF-Initiated Session Locking</u>	<u>Feature Dependent</u>
<u>FTA_TAB.1</u>	<u>Default TOE Access Banner</u>	<u>One</u>
<u>FTP_ITC.1</u>	<u>Inter-TSF Trusted Channel (Refinement)</u>	<u>One</u>
<u>FTP_TRP.1/Admin</u>	<u>Trusted Path (Refinement)</u>	<u>One</u>
<u>FTP_TRP.1/Join</u>	<u>Trusted Path</u>	<u>Feature Dependent</u>
<u>FMT_MTD.1/CryptoKeys</u>	<u>Management of TSF Data</u>	<u>Feature Dependent</u>
<u>FMT_MOF.1/ManualUpdate</u>	<u>Management of security functions behaviour</u>	<u>Feature Dependent</u>
<u>FMT_MOF.1/AutoUpdate</u>	<u>Management of security functions behaviour</u>	<u>Feature Dependent</u>

*Table 1: Security Functional Requirements for Distributed TOEs*

The ST for a distributed TOE must include a mapping of SFRs to each of the components of the TOE. (Note that this deliverable is examined as part of the ASE\_TSS.1 and AVA\_VAN.1 Evaluation Activities as described in [SD, 5.1.2] and [SD, 5.6.1.1] respectively. The ST for a distributed TOE may also introduce a ‘minimum configuration’ and identify components that may have instances added to an operational configuration without affecting the validity of the CC certification. [SD, B.4] describes Evaluation Activities relating to these equivalency aspects of a distributed TOE (and hence what is expected in the ST).

### **3.4. Security Problem Definition**

A network device has a network infrastructure role that it is designed to provide. In doing so, the network device communicates with other network devices and other network entities (~~an~~ entity~~i.e. entities~~ not defined as ~~a~~network device~~devices~~ because they do not have an infrastructure role) over the network. ~~-~~ At the same time, it must provide a minimal set of common security functionality expected by all network devices. ~~-~~The security problem to be addressed by a compliant network device is defined as this set of common security functionality that addresses the threats that are common to network devices, as opposed to those that might be targeting the specific functionality of a specific type of network device. The set of common security functionality addresses communication with the network device, both authorized and unauthorized, the ability to perform valid ~~or~~and secure updates, the ability to audit device activity, the ability to securely store and utilize device and administrator credentials and data, and the ability to self-test critical device components for failures.

#### **3.14.1 Threats**

The threats for the Network Device are grouped according to functional areas of the device in the sections below. The description of each threat is then followed by a rationale describing how it is addressed by the SFRs in section 6, appendix A, and appendix B.

##### **3.1.14.1.1 Communications with the Network Device**

A network device communicates with other network devices and other network entities. The endpoints of this communication can be geographically and logically distant and may pass through a variety of other systems. The intermediate systems may be untrusted providing an opportunity for unauthorized communication with the network device or for authorized communication to be compromised. The security functionality of the network device must be able to protect any critical network traffic (administration traffic, authentication traffic, audit traffic, etc.). The communication with the network device falls into two categories: authorized communication and unauthorized communication.

Authorized communication includes network traffic allowable by policy destined to and originating from the network device as it was designed and intended. This includes critical network traffic, such as network device administration and communication with an authentication or audit logging server, which requires a secure channel to protect the communication. The security functionality of the network device includes the capability to ensure that only authorized communications are allowed and the capability to provide a secure channel for critical network traffic. Any other communication is considered unauthorized communication.

The primary threats to network device communications addressed in this cPP focus on an external, unauthorized entity attempting to access, modify, or otherwise disclose the critical network traffic. A poor choice of cryptographic algorithms or the use of non-standardized tunneling protocols along with weak administrator credentials, such as an easily guessable password or use of a default password, will allow a threat agent unauthorized access to the device. Weak or no cryptography provides little to no protection of the traffic allowing a threat agent to read, manipulate and/or control the critical data with little effort. Non-

standardized tunneling protocols not only limit the interoperability of the device but lack the assurance and confidence standardization provides through peer review.

#### **3.1.1.14.1.1.1 T.UNAUTHORIZED\_ADMINISTRATOR\_ACCESS**

Threat agents may attempt to gain administrator access to the network device by nefarious means such as masquerading as an administrator to the device, masquerading as the device to an administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between network devices. Successfully gaining administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.

##### SFR Rationale:

- The administrator role is defined in FMT\_SMR.2 and the relevant administration capabilities are defined in FMT\_SMF.1 and FMT\_MTD.1/CoreData, with optional additional capabilities in FMT\_MOF.1/Services and FMT\_MOF.1/Functions
- The actions allowed before authentication of an administrator are constrained by FIA\_UIA\_EXT.1, and include the advisory notice and consent warning message displayed according to FTA\_TAB.1
- The requirement for the administrator authentication process is described in FIA\_UAU\_EXT.2
- Locking of administrator sessions is ensured by FTA\_SSL\_EXT.1 (for local sessions), FTA\_SSL.3 (for remote sessions), and FTA\_SSL.4 (for all interactive sessions)
- The secure channel used for remote administrator connections is specified in FPT\_TRP.1/Admin
- (Malicious actions carried out from an administrator session are separately addressed by T.UNDETECTED\_ACTIVITY)
- (Protection of the administrator credentials is separately addressed by T.PASSWORD\_CRACKING).

#### **3.1.1.24.1.1.2 T.WEAK\_CRYPTOGRAPHY**

Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.

##### SFR Rationale:

- Requirements for key generation and key distribution are set in FCS\_CKM.1 and FCS\_CKM.2 respectively



- Requirements for use of cryptographic schemes are set in FCS\_COP.1/DataEncryption, FCS\_COP.1/SigGen, FCS\_COP.1/Hash, and FCS\_COP.1/KeyedHash
- Requirements for random bit generation to support key generation and secure protocols (see SFRs resulting from T.UNTRUSTED\_COMMUNICATION\_CHANNELS) are set in FCS\_RBG\_EXT.1
- Management of cryptographic functions is specified in FMT\_SMF.1

#### **3.1.1.34.1.1.3 T.UNTRUSTED\_COMMUNICATION\_CHANNELS**

Threat agents may attempt to target network devices that do not use standardized secure tunneling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the network device itself.

##### SFR Rationale:

- The general use of secure protocols for identified communication channels is described at the top level in FTP\_ITC.1 and FTP\_TRP.1/Admin; for distributed TOEs the requirements for inter-component communications are addressed by the requirements in FPT\_ITT.1
- Requirements for the use of secure communication protocols are set for all the allowed protocols in FCS\_HTTPS\_EXT.1, FCS\_IPSEC\_EXT.1, FCS\_SSHC\_EXT.1, FCS\_SSHS\_EXT.1, FCS\_TLSC\_EXT.1, FCS\_TLSC\_EXT.2, FCS\_TLSS\_EXT.1, FCS\_TLSS\_EXT.2
- Optional and selection-based requirements for use of public key certificates to support secure protocols are defined in FIA\_X509\_EXT.1, FIA\_X509\_EXT.2, FIA\_X509\_EXT.3

#### **3.1.1.44.1.1.4 T.WEAK\_AUTHENTICATION\_ENDPOINTS**

Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised.

##### SFR Rationale:

- The use of appropriate secure protocols to provide authentication of endpoints (as in the SFRs addressing T.UNTRUSTED\_COMMUNICATION\_CHANNELS) are

ensured by the requirements in FTP\_ITC.1 and FTP\_TRP.1/Admin; for distributed TOEs the authentication requirements for endpoints in inter-component communications are addressed by the requirements in FPT\_ITT.1

- Additional possible special cases of secure authentication during registration of distributed TOE components are addressed by FCO\_CPC\_EXT.1 and FTP\_TRP.1/Join.

### **3.1.24.1.2 Valid Updates**

Updating network device software and firmware is necessary to ensure that the security functionality of the network device is maintained. The source and content of an update to be applied must be validated by cryptographic means; otherwise, an invalid source can write their own firmware or software updates that circumvents the security functionality of the network device. Methods of validating the source and content of a software or firmware update by cryptographic means typically involve cryptographic signature schemes where hashes of the updates are digitally signed.

Unpatched versions of software or firmware leave the network device susceptible to threat agents attempting to circumvent the security functionality using known vulnerabilities. Non-validated updates or updates validated using non-secure or weak cryptography leave the updated software or firmware vulnerable to threat agents attempting to modify the software or firmware to their advantage.

### **3.1.2.14.1.2.1 T.UPDATE\_COMPROMISE**

Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.

#### **SFR Rationale:**

- Requirements for protection of updates are set in FPT\_TUD\_EXT.1
- Additional optional use of certificate-based protection of signatures can be specified using FPT\_TUD\_EXT.2, supported by the X.509 certificate processing requirements in FIA\_X509\_EXT.1, FIA\_X509\_EXT.2 and FIA\_X509\_EXT.3
- Requirements for management of updates are defined in FMT\_SMF.1 and (for manual updates) in FMT\_MOF.1/ManualUpdate, with optional requirements for automatic updates in FMT\_MOF.1/AutoUpdate

### **3.1.34.1.3 Audited Activity**

Auditing of network device activities is a valuable tool for administrators to monitor the status of the device. It provides the means for administrator accountability, security functionality activity reporting, reconstruction of events, and problem analysis. Processing performed in response to device activities may give indications of a failure or compromise of

the security functionality. When indications of activity that impact the security functionality are not generated and monitored, it is possible for such activities to occur without administrator awareness. Further, if records are not generated and retained, reconstruction of the network and the ability to understand the extent of any compromise could be negatively affected. Additional concerns are the protection of the audit data that is recorded from alteration or unauthorized deletion. This could occur within the TOE, or while the audit data is in transit to an external storage device.

Note this cPP requires that the network device generate the audit data and have the capability to send the audit data to a trusted network entity (e.g., a syslog server).

### **3.1.3.14.1.3.1 T.UNDETECTED\_ACTIVITY**

Threat agents may attempt to access, change, and/or modify the security functionality of the network device without administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the administrator would have no knowledge that the device has been compromised.

#### **SFR Rationale:**

- Requirements for basic auditing capabilities are specified in FAU\_GEN.1 and FAU\_GEN.2, with timestamps provided according to FPT\_STM.1
- Requirements for protecting audit records stored on the TOE are specified in FAU\_STG.1
- Requirements for secure transmission of local audit records to an external IT entity via a secure channel are specified in FAU\_STG\_EXT.1
- Optional additional requirements for dealing with potential loss of locally stored audit records are specified in FAU\_STG\_EXT.2/LocSpace, FAU\_STG.3/LocSpace and FPT\_FLS.1/LocSpace
- If (optionally) configuration of the audit functionality is provided by the TOE then this is specified in FMT\_SMF.1, and confining this functionality to Security Administrators is required by FMT\_MOF.1/Functions.

### **3.1.44.1.4 Administrator and Device Credentials and Data**

A network device contains data and credentials which must be securely stored and must appropriately restrict access to authorized entities. Examples include the device firmware, software, configuration authentication credentials for secure channels, and administrator credentials. Device and administrator keys, key material, and authentication credentials need to be protected from unauthorized disclosure and modification. Furthermore, the security functionality of the device needs to require default authentication credentials, such as administrator passwords, be changed.

Lack of secure storage and improper handling of credentials and data, such as unencrypted credentials inside configuration files or access to secure channel session keys, can allow an attacker to not only gain access to the network device, but also compromise the security of the network through seemingly authorized modifications to configuration or through man-in-

the-middle attacks. These attacks allow an unauthorized entity to gain access and perform administrative functions using the Security Administrator's credentials and to intercept all traffic as an authorized endpoint. This results in difficulty in detection of security compromise and in reconstruction of the network, potentially allowing continued unauthorized access to administrator and device data.

#### **3.1.4.14.1.4.1 T.SECURITY\_FUNCTIONALITY\_COMPROMISE**

Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.

##### SFR Rationale:

- Protection of secret/private keys against compromise is specified in FPT\_SKP\_EXT.1
- Secure destruction of keys is specified in FCS\_CKM.4
- If (optionally) management of keys is provided by the TOE then this is specified in FMT\_SMF.1, and confining this functionality to Security Administrators is required by FMT\_MTD.1/CryptoKeys
- (Protection of passwords is separately covered under T.PASSWORD\_CRACKING).

#### **3.1.4.24.1.4.2 T.PASSWORD\_CRACKING**

Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.

##### SFR Rationale:

- Requirements for password lengths and available characters are set in FIA\_PMG\_EXT.1
- Protection of password entry by providing only obscured feedback is specified in FIA\_UAU.7
- Actions on reaching a threshold number of consecutive password failures are specified in FIA\_AFL.1
- Requirements for secure storage of passwords are set in FPT\_APW\_EXT.1.

#### **3.1.54.1.5 Device Failure**

Security mechanisms of the network device generally build up from roots of trust to more complex sets of mechanisms. Failures could result in a compromise to the security functionality of the device. A network device self-testing its security critical components at both start-up and during run-time ensures the reliability of the device's security functionality.

### 3.1.5.14.1.5.1 T.SECURITY\_FUNCTIONALITY\_FAILURE

A component of the network device may fail during start-up or during operations causing a compromise or failure in the security functionality of the network device, leaving the device susceptible to attackers.

#### SFR Rationale:

- Requirements for running self-test are defined in FPT\_TST\_EXT.1
- Optional use of certificates to support self-test is defined in FPT\_TST\_EXT.2 (with support for the use of certificates in FIA\_X509\_EXT.1, FIA\_X509\_EXT.2, and FIA\_X509\_EXT.3).

### 3.2.2 Assumptions

This section describes the assumptions made in identification of the threats and security requirements for network devices. The network device is not expected to provide assurance in any of these areas, and as a result, requirements are not included to mitigate the threats associated.

#### 3.2.14.2.1 A.PHYSICAL\_PROTECTION

The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.

[OE.PHYSICAL]

#### 3.2.24.2.2 A.LIMITED\_FUNCTIONALITY

The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example the device should not provide computing platform for general purpose applications (unrelated to networking functionality).

[OE.NO\_GENERAL\_PURPOSE]

#### 3.2.34.2.3 A.NO\_THRU\_TRAFFIC\_PROTECTION

A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by

the ND cPP. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g, firewall).

[OE.NO\_THRU\_TRAFFIC\_PROTECTION]

#### **3.2.44.2.4 A.TRUSTED\_ADMINISTRATOR**

The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious administrator that actively works to bypass or compromise the security of the device.

[OE.TRUSTED\_ADMIN]

#### **3.2.54.2.5 A.REGULAR\_UPDATES**

The network device firmware and software is assumed to be updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

[OE.UPDATES]

#### **3.2.64.2.6 A.ADMIN\_CREDENTIALS\_SECURE**

The administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.

[OE.ADMIN\_CREDENTIALS\_SECURE]

#### **4.2.7 A.COMPONENTS\_RUNNING (applies to distributed TOEs only)**

For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.

[OE.COMPONENTS\_RUNNING]

#### **4.2.8 A.RESIDUAL\_INFORMATION**

The administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

[OE.RESIDUAL\_INFORMATION]

### **3.3.4.3 Organizational Security Policy**

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. ~~For the purposes of this ePP a single policy is described in the section below.~~ The description of each policy is then followed by a rationale describing how it is addressed by the SFRs in section 6, appendix A, and appendix B.

#### **3.3.14.3.1 P.ACCESS\_BANNER**

The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

~~{FTA\_TAB.1}~~

#### SFR Rationale:

- An advisory notice and consent warning message is required to be displayed by FTA TAB.1

## **4.5. Security Objectives**

### **4.15.1 Security Objectives for the Operational Environment**

The following subsections describe objectives for the Operational Environment.

#### **4.1.15.1.1 OE.PHYSICAL**

Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

#### **4.1.25.1.2 OE.NO\_GENERAL\_PURPOSE**

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

#### **4.1.35.1.3 OE.NO\_THRU\_TRAFFIC\_PROTECTION**

The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

#### **4.1.45.1.4 OE.TRUSTED\_ADMIN**

TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

#### **4.1.55.1.5 OE.UPDATES**

The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

#### **4.1.65.1.6 OE.ADMIN\_CREDENTIALS\_SECURE**

The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

### **5.1.7 OE.COMPONENTS RUNNING (applies to distributed TOEs only)**

For distributed TOEs the TOE Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The TOE Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.

### **5.1.8 OE.RESIDUAL INFORMATION**

The TOE administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on



networking equipment when the equipment is discarded or removed from its operational environment.

## 5.6. Security Functional Requirements

The individual security functional requirements are specified in the sections below. SFRs in this section are mandatory SFRs that any conformant TOE must meet. Based on selections made in these SFRs it will also be necessary to include some of the selection-based SFRs in Appendix [BB](#). Additional optional SFRs may also be adopted from those listed in Appendix [AA](#).

For a distributed TOE, the ST author should reference Table 1 for guidance on how each SFR should be met. The table details whether SFRs should be met by all TOE components, by at least one TOE component or whether they are dependent upon the feature being implemented by the TOE component. The ST for a distributed TOE must include a mapping of SFRs to each of the components of the TOE. (Note that this deliverable is examined as part of the ASE TSS.1 and AVA VAN.1 Evaluation Activities as described in [SD, 5.1.2] and [SD, 5.6.1.1] respectively.

The Evaluation Activities defined in [SD] describe actions that the evaluator will take in order to determine compliance of a particular TOE with the SFRs. The content of these Evaluation Activities will therefore provide more insight into deliverables required from TOE Developers.

### 5.16.1 Conventions

The conventions used in descriptions of the SFRs are as follows:

- Assignment: Indicated with *italicized text*;
- Refinement made by PP author: Indicated with **bold text** and ~~strikethroughs~~, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined text*;
- Iteration: Indicated by ~~appending the iteration number in parenthesis, e.g., (1), (2), (3)~~ and/or by adding a string starting with “/” (e.g. “FCS COP.1/Hash”).

Extended SFRs are identified by having a label ‘EXT’ at the end of the SFR name.

Where compliance to RFCs is referred to in SFRs, this is intended to be demonstrated by completing the corresponding evaluation activities in [SD] for the relevant SFR.

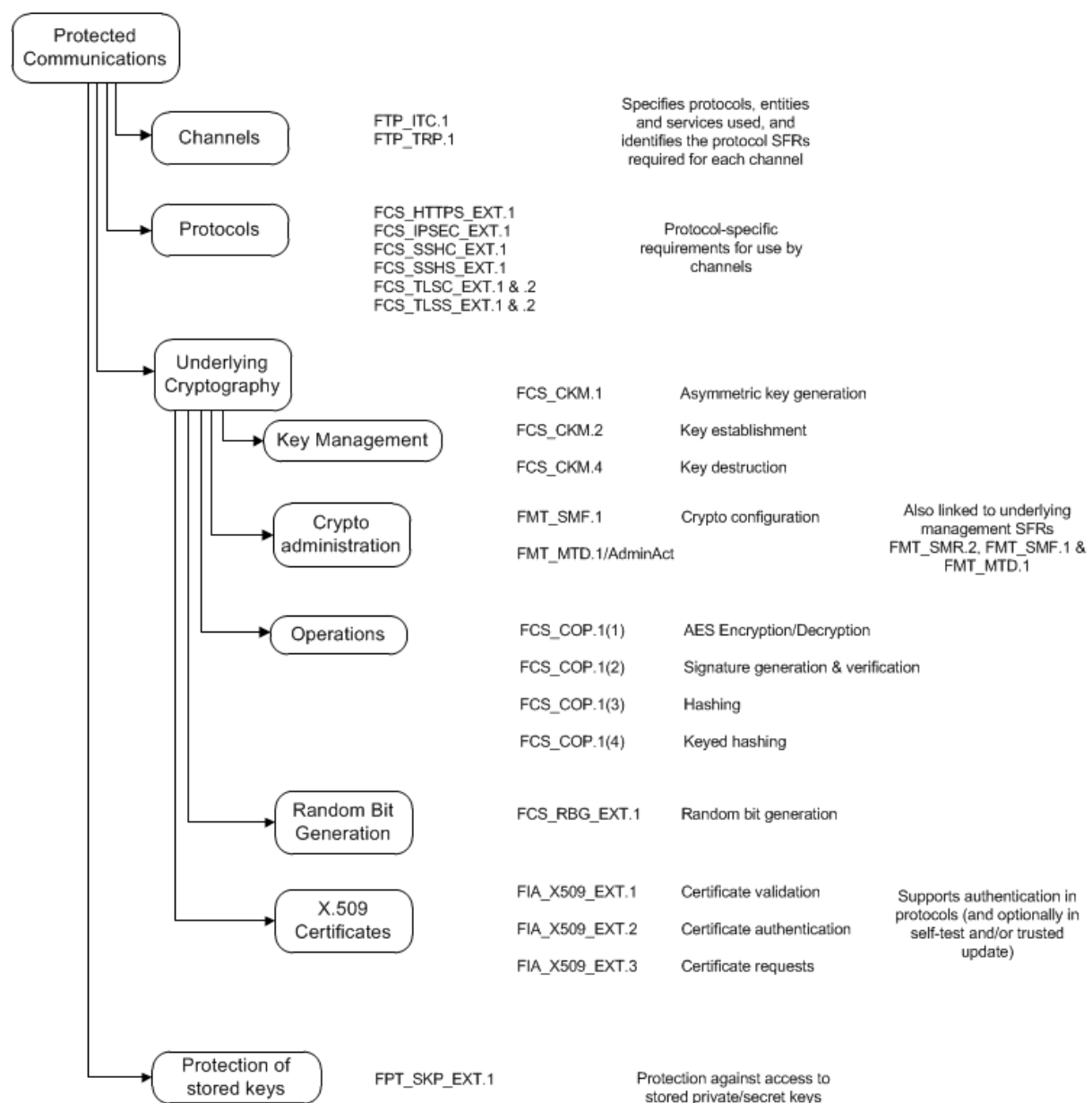
### 5.26.2 SFR Architecture

~~Figure 14~~Figure 1, ~~Figure 15~~Figure 2, ~~Figure 16~~Figure 3, ~~Figure 17~~Figure 4 and ~~Figure 18~~Figure 5 and ~~Figure 19~~ give a graphical presentation of the connections between the Security Functional Requirements in sections ~~6.35.3-6.95.9~~, Appendix [AA](#) and Appendix [BB](#), and the underlying functional areas and operations that the TOE provides. The diagrams

provide a context for SFRs that relates to their use in the TOE, whereas other sections define the SFRs grouped by the abstract class and family groupings in [CC2].

In general, the SFRs from Appendix BB that are required by an ST are determined by the selections made in other SFRs. For example: FTP\_ITC.1 and FTP\_TRP.1/Admin (in sections 6.9.1.15-9.1.1 and 6.9.2.15-9.2.1 respectively) each contain selections of a protocol to be used for the type of secure channel described by the SFR. The selection of the protocol(s) here determines which of the protocol-specific SFRs in section B.2.1B.2.1 are also required in the ST.

SFRs in Appendix AA can be included in the ST if they are provided by the TOE, but are not mandatory in order for a TOE to claim conformance to this cPP.



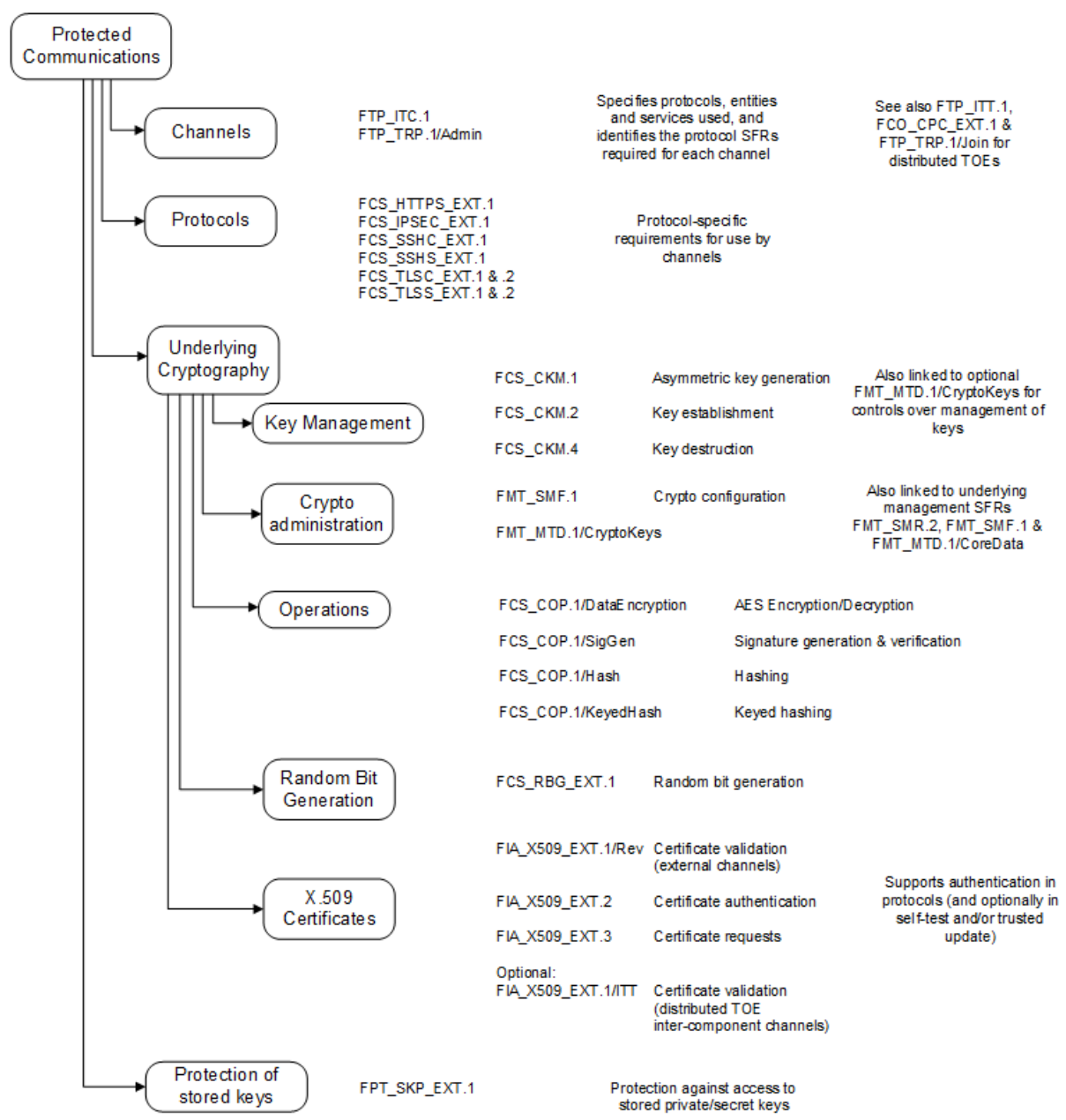


Figure 14: Protected Communications SFR Architecture

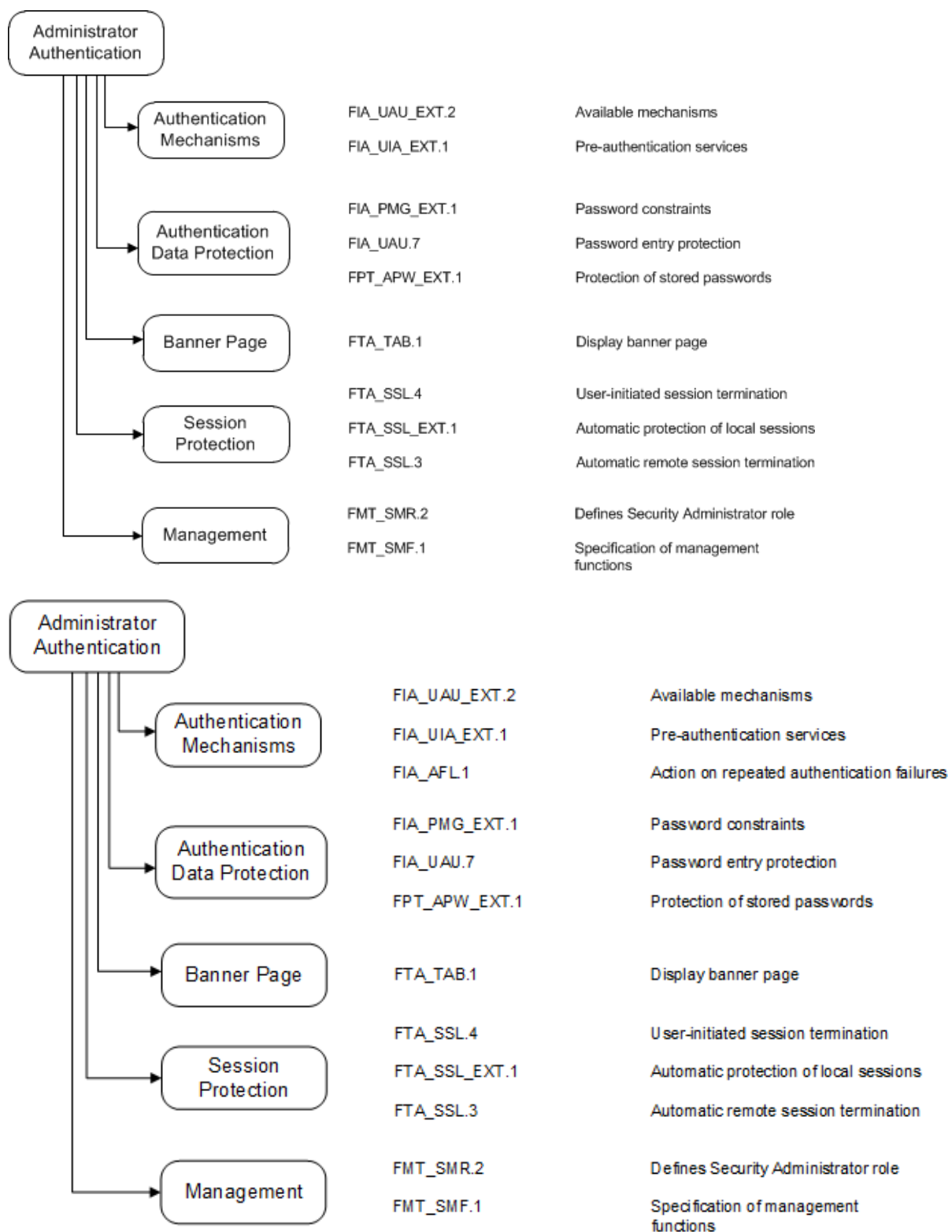


Figure 152: Administrator Authentication SFR Architecture

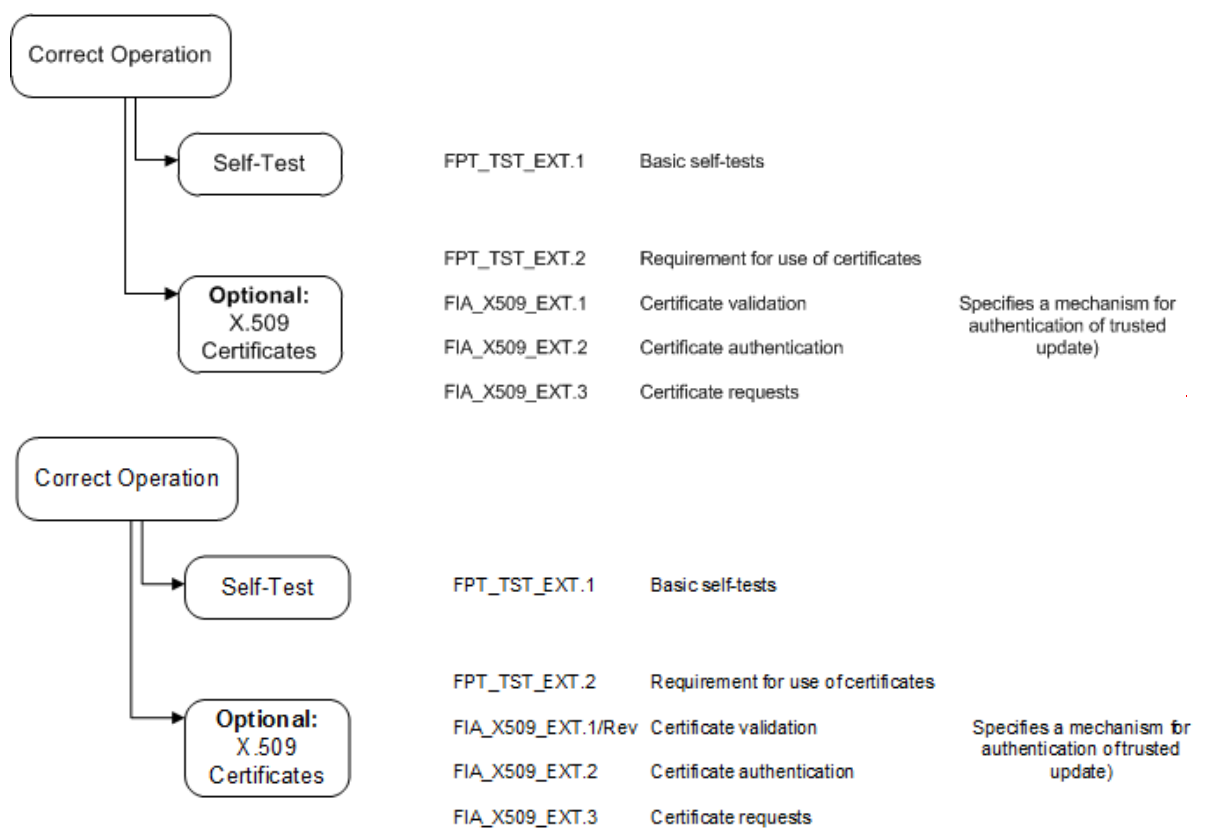
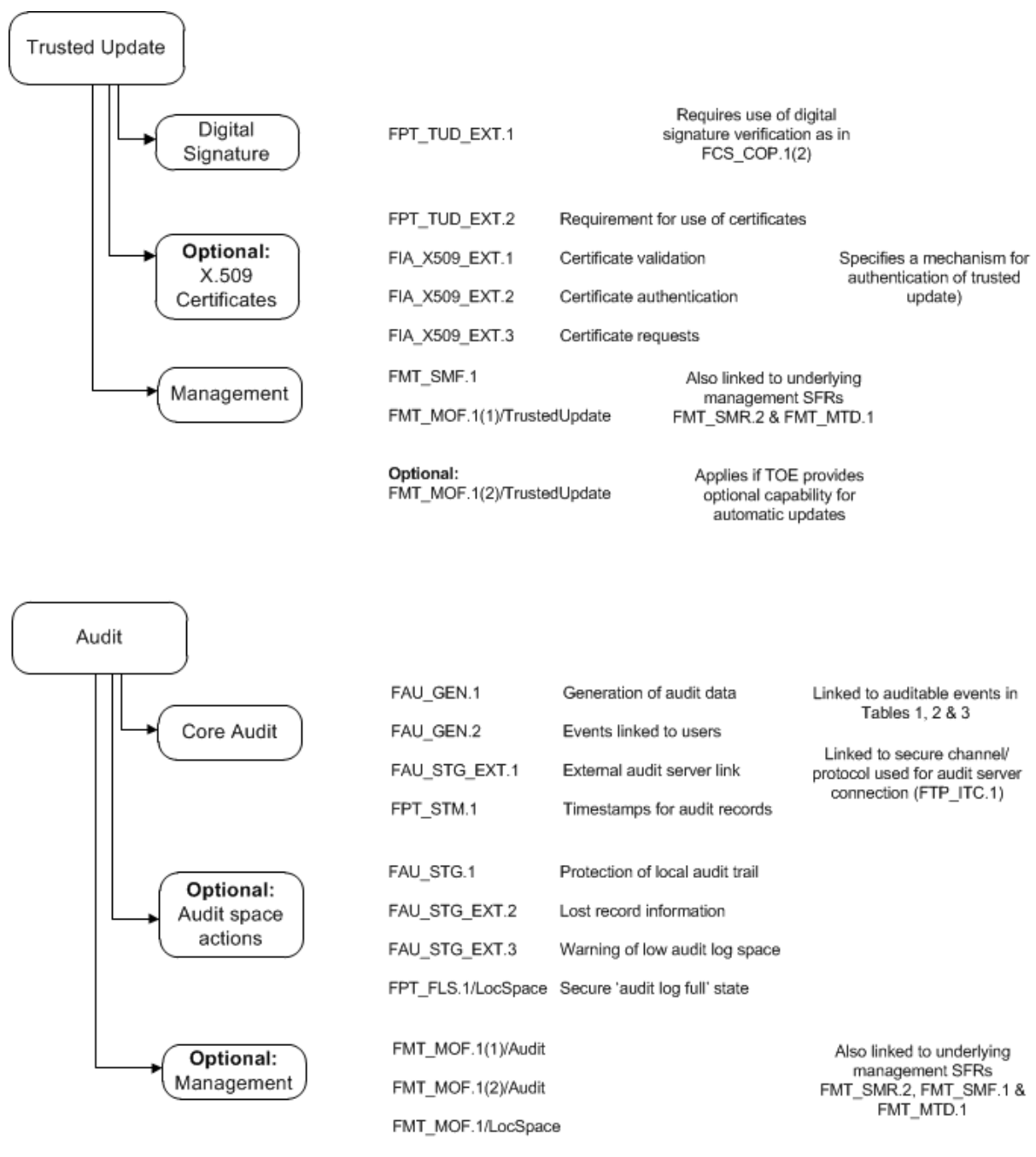


Figure 163: Correct Operation SFR Architecture



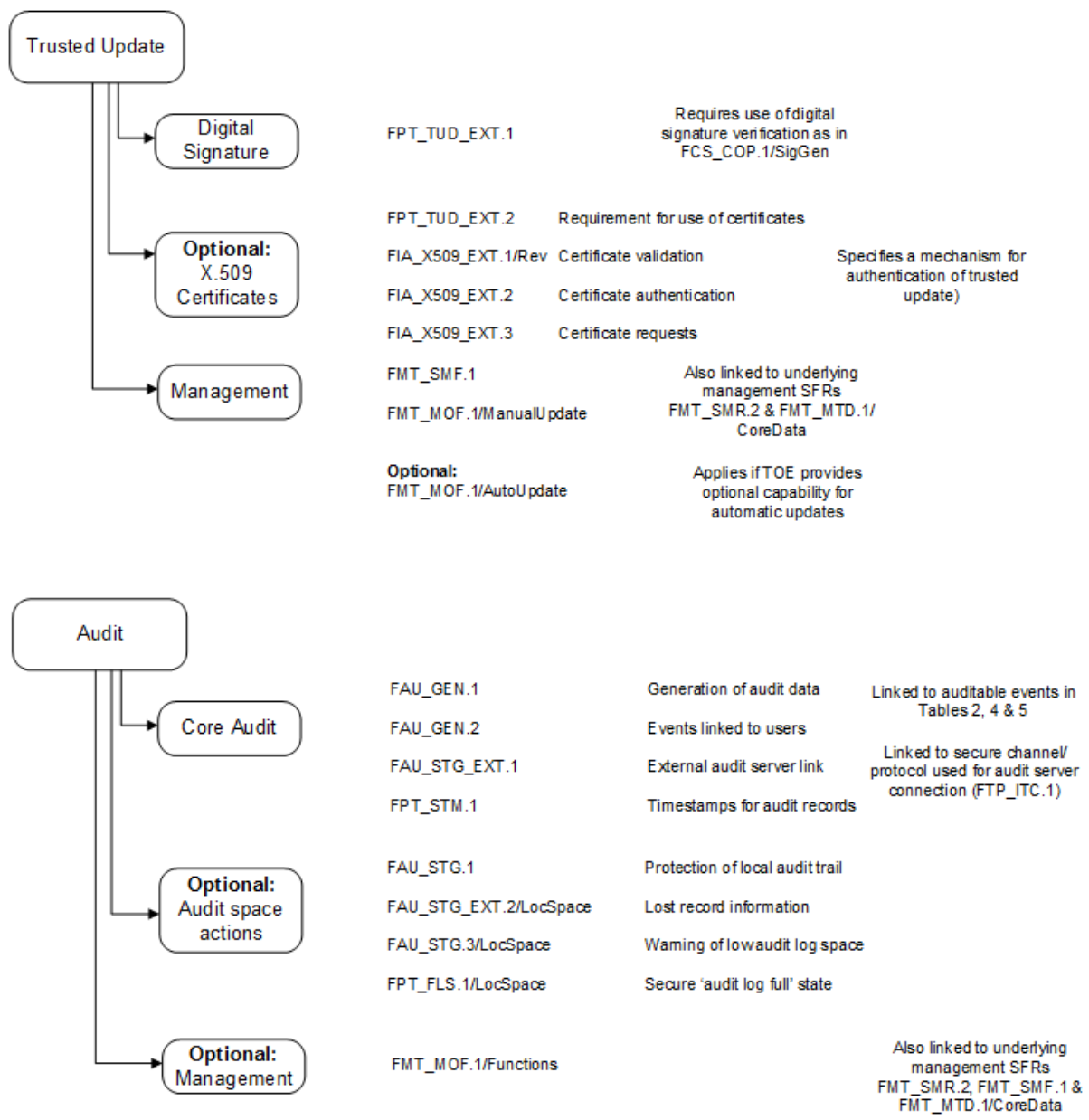


Figure 174: Trusted Updated and Audit SFR Architecture



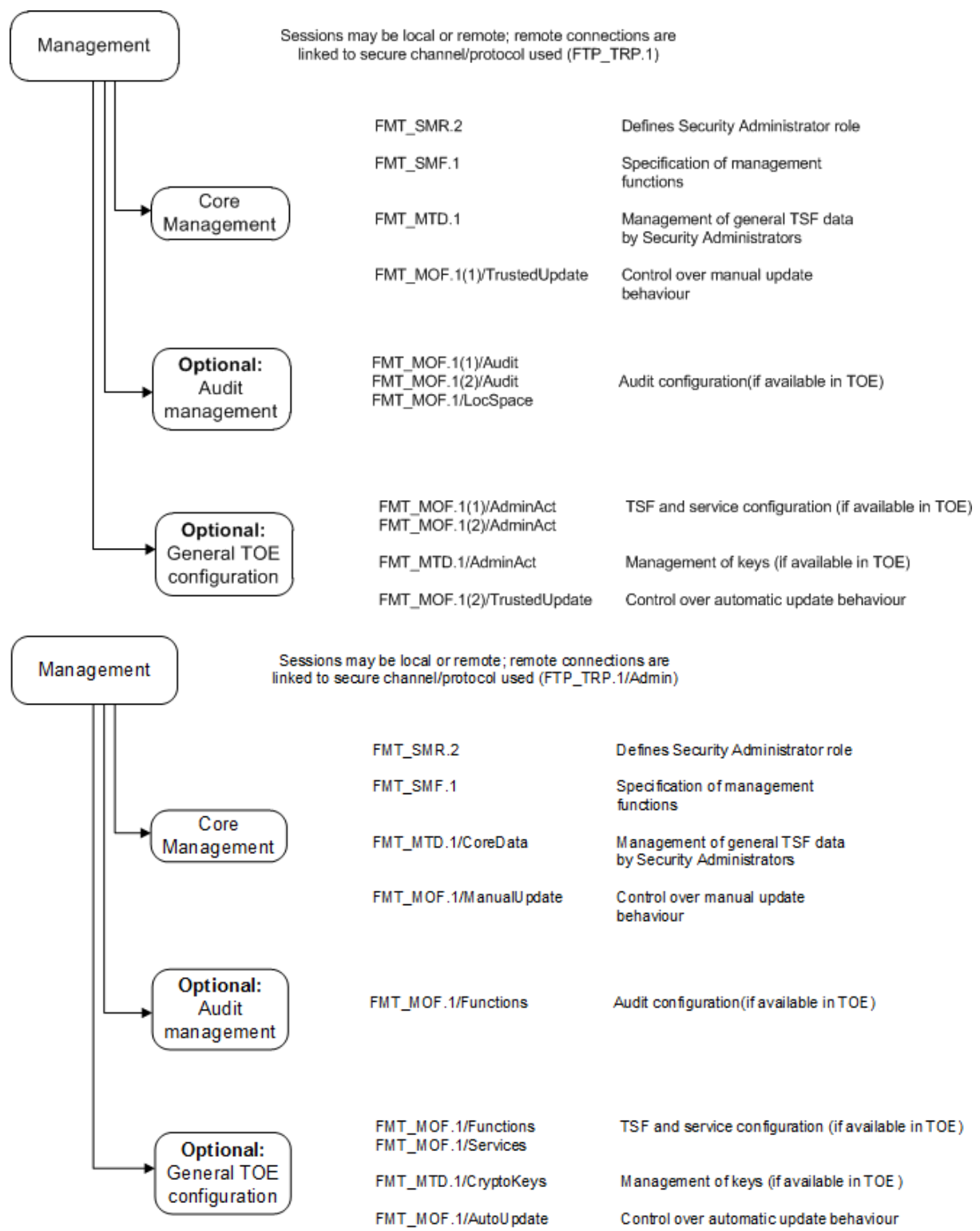
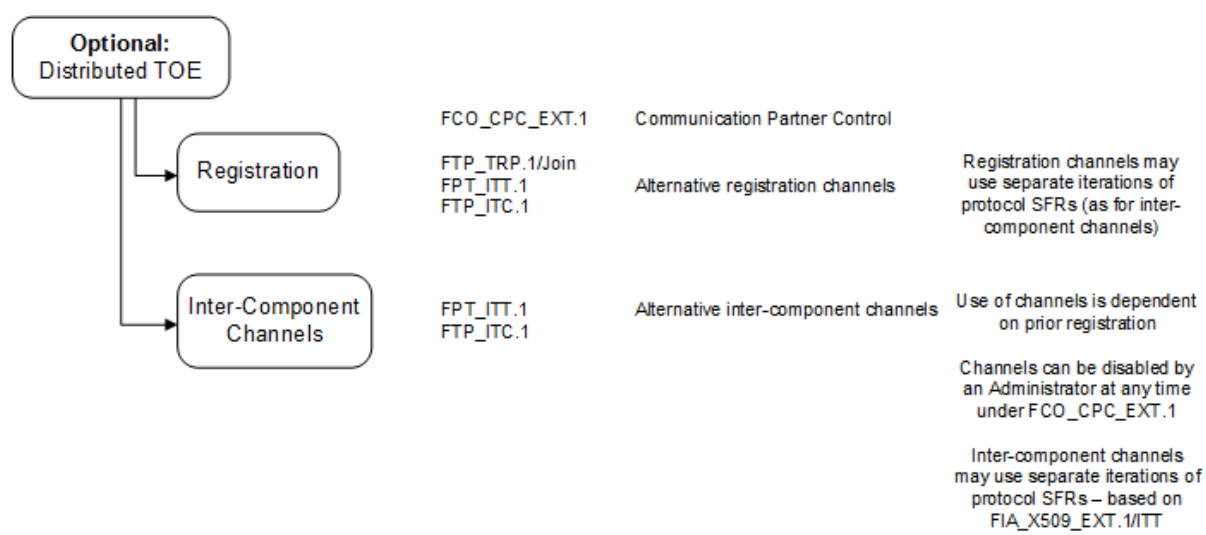


Figure 185: Management SFR Architecture



*Figure 19: Distributed TOE SFR Architecture*

### 5.36.3 Security Audit (FAU)

#### 5.3.16.3.1 Security Audit Data generation (FAU\_GEN)

In order to assure that information exists that allows Security Administrators to discover intentional and unintentional issues with the configuration and/or operation of the system, compliant TOEs have the capability of generating audit data targeted at detecting such activity. Auditing of administrative activities provides information that may be used to hasten corrective action should the system be configured incorrectly. Audit of select system events can provide an indication of failure of critical portions of the TOE (e.g. a cryptographic provider process not running) or anomalous activity (e.g. establishment of an administrative session at a suspicious time, repeated failures to establish sessions or authenticate to the system) of a suspicious nature.

In some instances there may be a large amount of audit information produced that could overwhelm the TOE or administrators in charge of reviewing the audit information. The TOE must be capable of sending audit information to an external trusted entity. This information must carry reliable timestamps, which will help order the information when sent to the external device.

Loss of communication with the audit server is problematic. While there are several potential mitigations to this threat, this cPP does not mandate that a specific action takes place; the degree to which this action preserves the audit information and still allows the TOE to meet its functionality responsibilities should drive decisions on the suitability of the TOE in a particular environment.

~~5.3.1.16.3.1.1~~ FAU\_GEN.1 Audit data generation

FAU_GEN.1	Audit Data Generation
-----------	-----------------------

**FAU\_GEN.1.1** The TSF shall be able to generate an audit record of the following auditable events:

- a) Start-up and shut-down of the audit functions;
- b) All auditable events for the not specified level of audit; and
- c) *All administrative actions comprising:*
  - *Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).*
  - *Security related configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
  - *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
  - *Resetting passwords (name of related user account shall be logged).*
  - *Starting and stopping services (if applicable)*
  - *Selection: [no other actions, assignment: [list of other uses of privileges]];*
- d) Specifically defined auditable events listed in Table 2~~Table 1~~.

**Application Note 1**

*If the list of 'administrative actions' appears to be incomplete, the assignment in the selection should be used to list additional administrative actions which are audited.*

*The ST author replaces the cross-reference to the table of audit events with an appropriate cross-reference for the ST. This must also include the relevant parts of Table 4~~Table 3~~ and Table 5~~Table 4~~ for optional and selection-based SFRs included in the ST.*

*For distributed TOEs each component must generate an audit record for each of the SFRs that it implements. If more than one TOE component is involved when an audit event is triggered, the event has to be audited on each component (e.g. rejection of a connection by one component while attempting to establish a secure communication channel between two components should result in an audit event being generated by both components). This is not limited to error cases but also includes events about successful actions like successful build up/tear down of a secure communication channel between TOE components.*

**Application Note 2**

*The ST author can include other auditable events directly in the table; they are not limited to the list presented.*

*The TSS should identify what information is logged to identify the relevant key for the administrative task of generating/import of, changing, or deleting of cryptographic keys.*

*With respect to FAU\_GEN.1.1 the term ‘services’ refers to trusted path and trusted channel communications, on demand self-tests, trusted update and administrator sessions (that exist under the trusted path) (e.g. netconf).*

**FAU\_GEN.1.2** The TSF shall record within each audit record at least the following information:

- a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
- b) For each audit event type, based on the auditable event definitions of the functional components included in the **PPcPP/ST**, *information specified in column three of Table 2~~Table 1~~.*

### **Application Note 3**

*The ST author replaces the cross-reference to the table of audit events with an appropriate cross-reference for the ST. This must also include the relevant parts of Table 4~~Table 3~~ and Table 5~~Table 4~~ for optional and selection-based SFRs included in the ST.*

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1 <del>(1)</del> <u>/DataEncryption</u>	None.	None.
FCS_COP.1 <del>(2)</del> <u>/SigGen</u>	None.	None.
FCS_COP.1 <del>(3)</del> <u>/Hash</u>	None.	None.
FCS_COP.1 <del>(4)</del> <u>/KeyedHash</u>	None.	None.
FCS_RBG_EXT.1	None.	None.
<u>FIA_AFL.1</u>	<u>Any breach of unsuccessful login</u>	<u>Provided user identity, origin of the attempt (e.g.,</u>

	<u>attempt limits</u>	<u>IP address).</u>
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
<del>FIA_X509_EXT.1</del>	<del>Unsuccessful attempt to validate a certificate</del>	<del>Reason for failure</del>
<del>FIA_X509_EXT.2</del>	<del>None</del>	<del>None</del>
<del>FIA_X509_EXT.3</del>	<del>None</del>	<del>None</del>
FMT_MOF.1(1)/ <u>TrustedUpdate/ManualUpdate</u>	Any attempt to initiate a manual update	None.
FMT_MTD.1/ <u>CoreData</u>	All management activities of TSF data.	None.
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	No additional information.
FPT_STM.1	Changes to time.	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1	Any attempts at unlocking of an	None.

	interactive session.	
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel.  Termination of the trusted channel.  Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/ <u>Admin</u>	Initiation of the trusted path.  Termination of the trusted path.  Failure of the trusted path functions.	Identification of the claimed user identity.

Table ~~2~~1: Security Functional Requirements and Auditable Events

#### Application Note 4

Additional audit events will apply to the TOE depending on the optional and selection-based requirements adopted from Appendix ~~AA~~ and Appendix ~~BB~~. The ST author must therefore include the relevant additional events specified in the tables in ~~Table 4~~Table 3 and ~~Table 5~~Table 4.

~~The audit event for FIA\_X509\_EXT.1 is based on the TOE not being able to complete the certificate validation by ensuring the following:~~

- ~~• the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.~~
- ~~• Verification of the digital signature of the trusted hierarchical CA~~
- ~~• read/access the CRL or access the OCSP server.~~

~~If any of these checks fails, then an audit event with the failure should be written to the audit log~~

**5.3.1.26.3.1.2 FAU\_GEN.2 User identity association**

<b>FAU_GEN.2</b>	<b>User identity association</b>
------------------	----------------------------------

**FAU\_GEN.2.1** For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

**Application Note 5**

*Where an auditable event is triggered by another component, the component that records the event must associate it with the identity of the initiating component that caused the event (applies to distributed TOEs only).*

**5.3.26.3.2 Security audit event storage (Extended – FAU\_STG\_EXT)**

A network device TOE is not expected to take responsibility for all audit storage itself. Although it is required to store data locally at the time of generation, and to take some appropriate action if this local storage capacity is exceeded, the TOE is also required to be able to establish a secure link to an external audit server to enable external audit trail storage.

**5.3.2.16.3.2.1 FAU\_STG\_EXT.1 Protected Audit Event Storage**

<b>FAU_STG_EXT.1</b>	<b>Protected Audit Event Storage</b>
----------------------	--------------------------------------

**FAU\_STG\_EXT.1.1** The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP\_ITC.1.

**Application Note 5**~~Application Note 6~~

*For selecting the option of transmission of generated audit data to an external IT entity the TOE relies on a non-TOE audit server for storage and review of audit records. The storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment in that case. Since the external audit server is not part of the TOE, there are no requirements on it except the capabilities for ITC transport for audit data. No requirements are placed upon the format or underlying protocol of the audit data being transferred. The TOE must be capable of being configured to transfer audit data to an external IT entity without administrator intervention. Manual transfer would not meet the requirements. Transmission could be done in real-time or periodically. If the transmission is not done in real-time then the TSS describes what event stimulates the transmission to be made and what range of frequencies the TOE supports for making transfers of audit data to the audit server; the TSS also suggests typical acceptable frequencies for the transfer.*

*For distributed TOEs each component must be able to export audit data across a protected channel external (FTP ITC.1) or intercomponent (FPT ITT.1 or FPT ITC.1) as appropriate. At least one component of the TOE must be able to export audit records via FTP ITC.1 such that all TOE audit records can be exported to an external IT entity.*

**FAU\_STG\_EXT.1.2** The TSF shall be able to store generated audit data on the TOE itself.

**FAU\_STG\_EXT.1.3** The TSF shall [selection: *drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]*] when the local storage space for audit data is full.

~~Application Note 6~~Application Note 7

*The external log server might be used as alternative storage space in case the local storage space is full. The 'other action' could in this case be defined as 'send the new audit data to an external IT entity'.*

For distributed TOEs each component must provide some amount of local storage to ensure that audit records are preserved in case of network connectivity issues. The behaviour when local storage is exhausted must be described for each component.

### 6.3.3 Security Audit for Distributed TOEs

For distributed TOEs the handling of audit information might be more complicated than for TOEs consisting only of one component. Basically there are a few basic requirements to be fulfilled:

- Every component must be able to generate audit information.
- Every component must either be able to buffer audit information and forward it to another TOE component or to store audit information locally.
- For the overall TOE it must be possible to store all audit information locally.
- For the overall TOE it must be possible to send out audit information to an external audit server.

In general, every component must be able to generate its own audit information. It would be possible that every component also stores its own audit information locally as well as every component could be able to send out audit data to an external audit server. But instead of this it would also be sufficient that every component would be able to generate its own audit data and buffer it locally before the information is sent out to one or more other TOE components for local storage and/or transmission to an external audit server. For the transfer of audit records between TOE components the secure connection via FTP ITC.1 or FPT ITT.1 must be used.

Such a solution would still be suitable to fulfil the requirement that all audit-related SFRs have to be fulfilled by all TOE components, although formally not every component would support local storage or transfer to an external audit server itself.

Regarding the establishment of Inter-TOE communication, error conditions as well as successful connection/teardown events should be captured by both ends of the connection.

Although all TOE components shall be able to generate their own audit data according to FAU\_GEN.1 for all the SFRs that they implement, not all TOE components have to provide audit data about all events. For distributed TOEs a mapping shall be provided to show which auditable events according to FAU\_GEN.1 are covered by which components (also giving a



justification that the records generated by each component cover all the SFRs that it implements). The overall TOE has to provide audit information about all events defined for FAU\_GEN.1. As a result at least one TOE component has to be assigned to every auditable event defined for FAU\_GEN.1. The part of the mapping related to Table 2 shall be consistent with the mapping of SFRs to TOE components for ASE\_TSS.1 in the sense that all components defined as generating audit information for a particular SFR should also contribute to that SFR in the mapping for ASE\_TSS.1. This applies not only to audit events defined for mandatory SFRs but also to all audit events for optional and selection-based SFRs as defined in Appendix A and Appendix B.

If one or more of the optional audit components FAU\_STG.1, FAU\_STG\_EXT.2/LocSpace and FAU\_STG.3/LocSpace or FPT\_FLS.1/LocSpace are selected in the Security Target derived from this cPP, then the SFR mapping for ASE\_TSS.1 must include a specific identification of the TOE components to which they apply.

#### 5.46.4 Cryptographic Support (FCS)

This section defines cryptographic requirements that underlie the other security properties of the TOE, covering key generation and random bit generation, key establishment methods, key destruction, and the various types of cryptographic operation to provide AES encryption/decryption, signature verification, hash generation, and keyed hash generation.

These SFRs support the implementation of the selection-based protocol-level SFRs in Appendix ~~B~~**B**.

##### 5.4.16.4.1 Cryptographic Key Management (FCS\_CKM)

##### 5.4.1.16.4.1.1 FCS\_CKM.1 Cryptographic Key Generation (~~Refined~~**Refinement**)

FCS_CKM.1	Cryptographic Key Generation
-----------	------------------------------

**FCS\_CKM.1.1** The TSF shall **generate asymmetric** cryptographic keys in accordance with a specified cryptographic key **generation** algorithm: [selection:

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3;*
- *ECC schemes using “NIST curves” [selection: P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4;*
- *FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.1*

] and specified cryptographic key sizes [assignment: ~~cryptographic key sizes~~] that meet the following: [assignment: ~~list of standards~~].

~~Application Note 7~~ Application Note 8

The ST author selects all key generation schemes used for key establishment and device authentication. When key generation is used for key establishment, the schemes in FCS\_CKM.2.1 and selected cryptographic protocols must match the selection. When key generation is used for device authentication, other than ssh-rsa, the public key is expected to be associated with an X.509v3 certificate.

~~If in a distributed TOE, if~~ the TOE component acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.

~~5.4.1.26.4.1.2~~ FCS\_CKM.2 Cryptographic Key Establishment (RefinedRefinement)

<b>FCS_CKM.2</b>	<b>Cryptographic Key Establishment</b>
------------------	--

**FCS\_CKM.2.1** The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [selection:

- **RSA-based key establishment schemes that meetsmeet the following: NIST Special Publication 800-56B, “Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography”;**
- **Elliptic curve-based key establishment schemes that meetsmeet the following: NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;**
- **Finite field-based key establishment schemes that meetsmeet the following: NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”**

~~] that meets the following: [assignment: list of standards].~~

~~Application Note 8~~ Application Note 9

This is a refinement of the SFR FCS\_CKM.2 to deal with key establishment rather than key distribution.

The ST author selects all key establishment schemes used for the selected cryptographic protocols.

The RSA-based key establishment schemes are described in Section 9 of NIST SP 800-56B; however, Section 9 relies on implementation of other sections in SP 800-56B. ~~If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.~~

The elliptic curves used for the key establishment scheme correlate with the curves specified in FCS\_CKM.1.1.

The domain parameters used for the finite field-based key establishment scheme are specified by the key generation according to FCS\_CKM.1.1.

#### 5.4.1.36.4.1.3 FCS\_CKM.4 Cryptographic Key Destruction

##### FCS\_CKM.4 Cryptographic Key Destruction

**FCS\_CKM.4.1** The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method ~~[selection:]~~:

- For plaintext keys in volatile memory storage, the destruction shall be executed by a ~~[selection:]~~ single ~~direct~~ overwrite ~~[selection:]~~ consisting of ~~[selection:]~~ a pseudo-random pattern using the TSF's RBG, ~~consisting of zeroes, ones, a new value of the key, [assignment: a value that does not contain any CSP]],~~ destruction of reference to the key directly followed by a ~~read-verify request for garbage collection];~~
  - ~~If the read-verification of the overwritten data fails, the process shall be repeated again.~~
- For plaintext keys in non-volatile EEPROM storage, the destruction shall be executed by ~~at~~ the invocation of an interface provided by a part of the TSF that ~~[selection:]~~
  - ~~logically addresses the storage location of the key and performs a [selection:] single, direct [assignment: number of passes]-pass] overwrite consisting of [selection:] a pseudo-random pattern using the TSF's RBG (as specified in FCS\_RBG\_EXT.1), followed by a read-verify, zeroes, ones, a new value of the key, [assignment: a value that does not contain any CSP]];~~
    - ~~If the read-verification of the overwritten data fails, the process shall be repeated again.~~
- ~~For non-volatile flash memory, the destruction shall be executed by [selection:] a single, direct overwrite consisting of zeroes, a block erase] followed by a read-verify.~~
  - ~~If instructs a part of the read-verification of TSF to destroy the overwritten data fails, abstraction that represents the process shall be repeated again.~~
- ~~For non-volatile memory other than EEPROM and flash, the destruction shall be executed by overwriting three or more times with a random pattern that is changed before each write.~~

†

that meets the following: No Standard.

#### Application Note 10

In parts of the selections where keys are identified as being destroyed by "a part of the TSF", the TSS identifies the relevant part and the interface involved. The interface referenced in the requirement could take different forms for different TOEs, the most likely of which is an application programming interface to an OS kernel. There may be various levels of abstraction visible. For instance, in a given implementation the application may have access to the file system details and may be able to logically address specific memory locations. In another implementation the application may simply have a handle to a resource and can only ask another part of the TSF such as the interpreter or OS to delete the resource.

*Where different key destruction methods are used for different keys and/or different destruction situations then the different methods and the keys/situations they apply to are described in the TSS (and the ST may use separate iterations of the SFR to aid clarity). The TSS describes all relevant keys used in the implementation of SFRs, including cases where the keys are stored in a non-plaintext form. In the case of non-plaintext storage, the encryption method and relevant key-encrypting-key are identified in the TSS.*

*Some selections allow assignment of “a value that does not contain any CSP”. This means that the TOE uses some specified data not drawn from an RBG meeting FCS RBG EXT requirements, and not being any of the particular values listed as other selection options. The point of the phrase “does not contain any CSP” is to ensure that the overwritten data is carefully selected, and not taken from a general pool that might contain current or residual data that itself requires confidentiality protection.*

*Key destruction does not apply to the public component of asymmetric key pairs.*

#### **5.4.26.4.2 Cryptographic Operation (FCS\_COP)**

##### **5.4.2.16.4.2.1 FCS\_COP.1 Cryptographic Operation**

<b>FCS_COP.1(1) <del>/DataEncryption</del> Cryptographic Operation (AES Data Encryption/ Decryption)</b>
--

**FCS\_COP.1.1(1) ~~/DataEncryption~~** The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in [selection: CBC, CTR, GCM] mode* and cryptographic key sizes *[selection: 128 bits, 192 bits, 256 bits]* that meet the following: *AES as specified in ISO 18033-3, [selection: CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772].*

#### **~~Application Note 9~~ Application Note 11**

*For the first selection of FCS\_COP.1.1(1) ~~/DataEncryption~~, the ST author should choose the mode or modes in which AES operates. For the second selection, the ST author should choose the key sizes that are supported by this functionality. The modes and key sizes selected here correspond to the cipher suite selections made in the trusted channel requirements.*

<b>FCS_COP.1(2) <u>SigGen</u> Cryptographic Operation (Signature Generation and Verification)</b>
---

**FCS\_COP.1.1(2) SigGen** The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm *[selection:*

- *RSA Digital Signature Algorithm and cryptographic key sizes (**modulus**) [assignment: 2048 bits or greater],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [assignment: 256 bits or greater]*

]

that meet the following: [selection:

- For RSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-~~PKCS2v1~~PKCS1v1\_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,
- For ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” ~~P-256, P-384, and~~ [selection: P-256, P-384, P-521, no other curves]; ISO/IEC 14888-3, Section 6.4

].

#### ~~Application Note 10~~Application Note 12

The ST Author should choose the algorithm implemented to perform digital signatures. For the algorithm(s) chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm. The ST author ensures that the assignments and selections for this SFR include all of the parameter values necessary for the cipher suites selected for the protocol SFRs (see Appendix B.2.1) that are included in the ST. The ST Author should check for consistency of selections with other FCS requirements, especially when supporting elliptic curves.

<b>FCS_COP.1(3)/Hash</b>	<b>Cryptographic Operation (Hash Algorithm)</b>
--------------------------	---

**FCS\_COP.1.1(3)/Hash** The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [selection: SHA-1, SHA-256, SHA-384, SHA-512] and cryptographic key sizes ~~[assignment: cryptographic key sizes]~~ and message digest sizes [selection: 160, 256, 384, 512] bits that meet the following: ISO/IEC 10118-3:2004.

#### ~~Application Note 11~~Application Note 13

Vendors are strongly encouraged to implement updated protocols that support the SHA-2 family; until updated protocols are supported, this ~~PPcPP~~ allows support for SHA-1 implementations in compliance with SP 800-131A. In a future version of this cPP, SHA-256 will be the minimum requirement for all TOEs.

The hash selection should be consistent with the overall strength of the algorithm used for ~~FCS\_COP.1(1)/DataEncryption~~ and ~~FCS\_COP.1(2)/SigGen~~ (for example, SHA 256 for 128-bit keys).

<b>FCS_COP.1(4)/KeyedHash</b>	<b>Cryptographic Operation (Keyed Hash Algorithm)</b>
-------------------------------	---

**FCS\_COP.1.1(4)/KeyedHash** The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [selection: HMAC-SHA-1, HMAC-SHA-

256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes [assignment: key size (in bits) used in HMAC] and message digest sizes [selection: 160, 256, 384, 512] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.

~~Application Note 12~~ Application Note 14

The key size [k] in the assignment 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  $L2 \leq k \leq L1$ .

~~5.4.3.6.4.3~~ Random Bit Generation (Extended – FCS\_RBG\_EXT)

~~5.4.3.16.4.3.1~~ FCS\_RBG\_EXT.1 Random Bit Generation

FCS_RBG_EXT.1	Random Bit Generation
---------------	-----------------------

**FCS\_RBG\_EXT.1.1** The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [selection: Hash\_DRBG (any), HMAC\_DRBG (any), CTR\_DRBG (AES)].

**FCS\_RBG\_EXT.1.2** The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [selection: [assignment: number of software-based sources] software-based noise source, [assignment: number of hardware-based sources] hardware-based noise source] with a minimum of [selection: 128 bits, 192 bits, 256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

~~Application Note 13~~ Application Note 15

For the first selection in FCS\_RBG\_EXT.1.2, the ST ~~author~~ selects at least one of the types of noise sources. If the TOE contains multiple noise sources of the same type, the ST author fills the assignment with the appropriate number for each type of source (e.g., 2 software-based noise sources, 1 hardware-based noise source). The documentation and tests required in the Evaluation Activity for this element ~~neessarily~~ should be repeated to cover each source indicated in the ST.

ISO/IEC 18031:2011 contains three different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used, and include the specific underlying cryptographic primitives used in the requirement. While any of the identified hash functions (SHA-1, ~~SHA-224,~~ SHA-256, SHA-384, SHA-512) are allowed for Hash\_DRBG or HMAC\_DRBG, only AES-based implementations for CTR\_DRBG are allowed.

If the key length for the AES implementation used here is different than that used to encrypt the user data, then FCS\_COP.1 may have to be adjusted or iterated to reflect the different key length. For the selection in FCS\_RBG\_EXT.1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG.



## **5.56.5 Identification and Authentication (FIA)**

In order to provide a trusted means for administrators to interact with the TOE, the TOE provides a password-based logon mechanism. The administrator must have the capability to compose a strong password, and have mechanisms in place so that the password must be changed regularly. To avoid attacks where an attacker might observe a password being typed by an administrator, passwords must be obscured during logon. Session locking or termination must also be implemented to mitigate the risk of an account being used illegitimately. Passwords must be stored in an obscured form, and there must be no interface provided for specifically reading the password or password file such that the passwords are displayed in plain text.

### **6.5.1 Authentication Failure Management (FIA AFL)**

#### **6.5.1.1 FIA AFL.1 Authentication Failure Management (Refinement)**

<b><u>FIA AFL.1 Authentication Failure Management</u></b>
---

FIA AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [assignment: range of acceptable values] unsuccessful authentication attempts occur related to administrators attempting to authenticate remotely.

FIA AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall [selection, choose one of: prevent the offending remote administrator from successfully authenticating until [assignment: action] is taken by a local Administrator; prevent the offending remote administrator from successfully authenticating until an Administrator defined time period has elapsed].

#### **Application Note 16**

This requirement does not apply to an administrator at the local console, since it does not make sense to lock a local administrator's account in this fashion. This could be addressed by (for example) requiring a separate account for local administrators or having the authentication mechanism implementation distinguish local and remote login attempts. The "action" taken by a local administrator is implementation specific and would be defined in the administrator guidance (for example, lockout reset or password reset). The ST author chooses one of the selections for handling of authentication failures depending on how the TOE has implemented this handler.

The TSS describes how the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking). The Operational Guidance describes, and identifies the importance of, any actions that are required in order to ensure that administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA AFL.1.

## ~~5.5.16.5.2~~ Password Management (Extended – FIA\_PMG\_EXT)

### ~~5.5.1.16.5.2.1~~ FIA\_PMG\_EXT.1 Password Management

FIA_PMG_EXT.1	Password Management
---------------	---------------------

**FIA\_PMG\_EXT.1.1** The TSF shall provide the following password management capabilities for administrative passwords:

- a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [selection: “!”, “@”, “#”, “\$”, “%”, “^”, “&”, “\*”, “(”, “)”, [assignment: other characters]];
- b) Minimum password length shall be settable by the Security Administrator, and shall support passwords of 15 characters or greater.

#### ~~Application Note 14~~Application Note 17

*The ST author selects the special characters that are supported by TOE; they may optionally list additional special characters supported using the assignment. "Administrative passwords" refers to passwords used by administrators at the local console, over protocols that support passwords, such as SSH and HTTPS, or to grant configuration data that supports other SFRs in the Security Target.*

## ~~5.5.26.5.3~~ User Identification and Authentication (Extended – FIA\_UIA\_EXT)

### ~~5.5.2.16.5.3.1~~ FIA\_UIA\_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1	User Identification and Authentication
---------------	--

**FIA\_UIA\_EXT.1.1** The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA\_TAB.1;
- [selection: no other actions, [assignment: list of services, actions performed by the TSF in response to non-TOE requests.]]

**FIA\_UIA\_EXT.1.2** The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

#### ~~Application Note 15~~Application Note 18

*This requirement applies to users (administrators and external IT entities) of services available from the TOE directly, and not services available by connecting through the TOE. While it should be the case that few or no services are available to external entities prior to identification and authentication, if there are some available (perhaps ICMP echo) these*



should be listed in the assignment statement; otherwise “no other actions” should be selected.

Authentication can be password-based through the local console or through a protocol that supports passwords (such as SSH), or be certificate based (such as SSH, TLS).

For communications with external IT entities (e.g., an audit server or NTP server, for instance), such connections must be performed in accordance with FTP\_ITC.1, whose protocols perform identification and authentication. This means that such communications (e.g., establishing the IPsec connection to the authentication server) would not have to be specified in the assignment, since establishing the connection “counts” as initiating the identification and authentication process.

According to the application note for FMT SMR.2, for distributed TOEs at least one TOE component has to support the authentication of Security Administrators according to FIA\_UIA\_EXT.1 and FIA\_UAU\_EXT.2 but not necessarily all TOE components. In case not all TOE components support this way of authentication for Security Administrators the TSS shall describe how Security Administrators are authenticated and identified.

#### **5.5.36.5.4 User authentication (FIA\_UAU) (Extended – FIA\_UAU\_EXT)**

##### **5.5.3.16.5.4.1 FIA\_UAU\_EXT.2 Password-based Authentication Mechanism**

<b>FIA_UAU_EXT.2</b>	<b>Password-based Authentication Mechanism</b>
----------------------	--

**FIA\_UAU\_EXT.2.1** The TSF shall provide a local password-based authentication mechanism, and [selection: [assignment: other authentication mechanism(s)], ~~no~~no other authentication mechanism] to perform administrative user authentication.

#### ~~Application Note 16~~Application Note 19

The assignment should be used to identify any additional local authentication mechanisms supported. Local authentication mechanisms are defined as those that occur through the local console; remote administrative sessions (and their associated authentication mechanisms) are specified in FTP\_TRP.1/Admin.

According to the application note for FMT SMR.2, for distributed TOEs at least one TOE component has to support the authentication of Security Administrators according to FIA\_UIA\_EXT.1 and FIA\_UAU\_EXT.2 but not necessarily all TOE components. In case not all TOE components support this way of authentication for Security Administrators the TSS shall describe how Security Administrators are authenticated and identified.

##### **5.5.3.26.5.4.2 FIA\_UAU.7 Protected Authentication Feedback**

<b>FIA_UAU.7</b>	<b>Protected Authentication Feedback</b>
------------------	--

**FIA\_UAU.7.1** The TSF shall provide only *obscured feedback* to the administrative user while the authentication is in progress at the local console.

~~Application Note 17~~ Application Note 20

“Obscured feedback” implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user that may provide any indication of the authentication data.

~~5.5.4 Authentication using X.509 certificates (Extended FIA\_X509\_EXT)~~

~~5.5.4.1 FIA\_X509\_EXT.1 X.509 Certificate Validation~~

~~FIA\_X509\_EXT.1~~ **X.509 Certificate Validation**

~~FIA\_X509\_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:~~

- ~~• RFC 5280 certificate validation and certificate path validation.~~
- ~~• The certificate path must terminate with a trusted CA certificate.~~
- ~~• The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.~~
- ~~The TSF shall validate the revocation status of the certificate using [selection: the~~  
*Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate*  
*Revocation List (CRL) as specified in RFC 5759].*
- ~~• The TSF shall validate the extendedKeyUsage field according to the following rules:~~
  - ~~○ Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.~~
  - ~~○ Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.~~
  - ~~○ Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.~~
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

~~Application Note 18~~

~~FIA\_X509\_EXT.1.1 lists the rules for validating certificates. The ST author selects whether revocation status is verified using OCSP or CRLs. The trusted channel/path protocols require that certificates are used; this use requires that the extendedKeyUsage rules are verified.~~

~~The validation is expected to end in a trusted root CA certificate in a root store managed by the platform.~~

~~**FIA\_X509\_EXT.1.2** The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.~~

~~**Application Note 19**~~~~**Application Note 1**~~

~~This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.~~

~~**5.5.4.2A.1.1.1 FIA\_X509\_EXT.2 X.509 Certificate Authentication**~~

~~**FIA\_X509\_EXT.2**~~ ~~**X.509 Certificate Authentication**~~

~~**FIA\_X509\_EXT.2.1** The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection: IPsec, TLS, HTTPS, SSH], and [selection: code signing for system software updates, code signing for integrity verification, [assignment: other uses], no additional uses].~~

~~**Application Note 20**~~

~~The ST author's selection matches the selection of FTP\_ITC.1.1. Certificates may optionally be used for trusted updates of system software (FPT\_TUD\_EXT.1) and for integrity verification (FPT\_TST\_EXT.2).~~

~~**FIA\_X509\_EXT.2.2** When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate].~~

~~**Application Note 21**~~

~~Often a connection must be established to check the revocation status of a certificate either to download a CRL or to perform a lookup using OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA\_X509\_EXT.1, the behavior indicated in the selection determines the validity. The TOE must not accept the certificate if it fails any of the other validation rules in FIA\_X509\_EXT.1. If the administrator configured option is selected by the ST Author, the ST Author also selects the corresponding function in FMT\_SMF.1.~~

~~**5.5.4.3**~~ ~~**FIA\_X509\_EXT.3 X.509 Certificate Requests**~~

~~**FIA\_X509\_EXT.3.1** The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: device specific information, Common Name, Organization, Organizational Unit, Country].~~

~~**Application Note 22**~~~~**Application Note 1**~~

~~The public key is the public key portion of the public private key pair generated by the TOE as specified in FCS\_CKM.1(1).~~

~~FIA\_X509\_EXT.3.2 The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.~~

## 5.66.6 Security Management (FMT)

Management functions required in this section describe required capabilities to support a Security Administrator role and basic set of security management functions dealing with management of configurable aspects included in other SFRs (FMT\_SMF.1), general management of TSF data (FMT\_MTD.1/CoreData), and enabling TOE updates (FMT\_MOF.1(~~1~~)/TrustedUpdate)/ManualUpdate).

For distributed TOEs security management of TOE components could be realized for every TOE component directly or through other TOE components. The TSS shall describe which management SFRs and management functions apply to each TOE component (applies only to distributed TOEs).

These core management requirements are supplemented by optional requirements in section A.4A.3 and selection-based requirements in section B.5B.4, according to the TOE capabilities.

### 5.6.16.6.1 Management of functions in TSF (FMT\_MOF)

#### 5.6.1.16.6.1.1 FMT\_MOF.1(~~1~~)/TrustedUpdate/ManualUpdate Management of security functions behaviour

<b>FMT_MOF.1(<del>1</del>)/<u>TrustedUpdate</u>/<u>ManualUpdate</u> functions behaviour</b>	<b>Management of security functions behaviour</b>
---	---

**FMT\_MOF.1.1(~~1~~)/TrustedUpdate/ManualUpdate** The TSF shall restrict the ability to enable the functions *to perform manual update to Security Administrators*.

#### Application Note 23~~Application Note 21~~

*FMT\_MOF.1(~~1~~)/TrustedUpdate/ManualUpdate restricts the initiation of manual updates to Security Administrators.*

### 5.6.26.6.2 Management of TSF Data (FMT\_MTD)

#### 5.6.2.16.6.2.1 FMT\_MTD.1/CoreData Management of TSF Data

<b>FMT_MTD.1/<u>CoreData</u> Management of TSF Data</b>
---

**FMT\_MTD.1.1/CoreData** The TSF shall restrict the ability to manage the *TSF data* to *Security Administrators*.

~~Application Note 24~~Application Note 22

The word “manage” includes but is not limited to create, initialize, view, change default, modify, delete, clear, and append. This SFR includes also the resetting of user passwords by the Security Administrator. The identifier ‘CoreData’ has been added here to separate this iteration of FMT\_MTD.1 from the optional iteration of FMT\_MTD.1 defined in Appendix A.4.2.1 (FMT\_MTD.1/CryptoKeys).

~~5.6.36.6.3~~ **5.6.36.6.3 Specification of Management Functions (FMT\_SMF)**

~~5.6.316.6.3.1~~ **5.6.316.6.3.1 FMT\_SMF.1 Specification of Management Functions**

<b>FMT_SMF.1</b>	<b>Specification of Management Functions</b>
------------------	--

**FMT\_SMF.1.1** The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [selection: digital signature, hash comparison] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA\_AFL.1;
- [selection:
  - Ability to configure audit behavior;
  - Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA\_UIA\_EXT.1;
  - Ability to configure the cryptographic functionality;
  - Ability to configure thresholds for SSH rekeying;
  - Ability to configure the interaction between TOE components, if applicable;
  - Ability to re-enable an administrator account;
  - No other capabilities.]

~~Application Note 25~~Application Note 23

The TOE must provide functionality for both local and remote administration, including the ability to configure the access banner for FTA\_TAB.1 and the session inactivity time(s) for ~~FTA\_SSL.3 & EXT.1 and~~ FTA\_SSL.43. The item “Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates” includes the relevant management functions from ~~FMT\_MOF.1(1)/TrustedUpdate/ManualUpdate,~~ FMT\_MOF.1(2)/TrustedUpdate/AutoUpdate (if included in the ST), FIA\_X509\_EXT.2.2 and FPT\_TUD\_EXT.1.2 and FPT\_TUD\_EXT.2.2 (if included in the ST and if they include an administrator-configurable action). Similarly, the selection “Ability to configure audit behavior” includes the relevant management functions from ~~FMT\_MOF.1(1)/Audit,~~ FMT\_MOF.1(2)/Audit, ~~FMT\_MOF.1.1(1)/AdminAct,~~ FMT\_MOF.1.1(2)/AdminAct/Services and FMT\_MOF.1/LeeSpaceFunctions, (for all of these SFRs that are included in the ST). If the TOE offers the ability for a remote administrator account to be disabled inline with

FIA AFL.1 then the ST author should select "Ability to re-enable an administrator account" to allow the account to be re-enabled by a local administrator. If the TOE offers the ability for the administrator to configure the audit behaviour, configure the services available prior to identification or authentication, or if any of the cryptographic functionality on the TOE can be configured, or if the ST is describing a distributed TOE, then the ST author makes the appropriate choice or choices in the second selection, otherwise select "No other capabilities:-" (in the latter case the selection may alternatively be left blank in the ST).

The selection 'Ability to configure thresholds for SSH rekeying' shall be included in the ST if the TOE supports configuration of the thresholds for the mechanisms used to fulfil FCS\_SSHC\_EXT.1.8 or FCS\_SSHS\_EXT.1.8 (such configuration then requires the inclusion of FMT\_MOF.1/Functions in the ST). If the TOE places limits on the values accepted for the thresholds then this is stated in the TSS.

For distributed TOEs the interaction between TOE components will be configurable (see FCO\_CPC\_EXT.1). Therefore the ST author includes the selection "Ability to configure the interaction between TOE components" for distributed TOEs. A simple example would be the change of communication protocol according to FPT\_ITT.1. Another example would be changing the management of a TOE component from direct remote administration to remote administration through another TOE component. A more complex use case would be if the realization of an SFR is achieved through two or more TOE components and the responsibilities between the two or more components could be modified.

For distributed TOEs that implement a registration channel (as described in FCO\_CPC\_EXT.1.2), the ST author uses the selection 'Ability to configure the cryptographic functionality' in this SFR, and its corresponding mapping in the TSS, to describe the configuration of any cryptographic aspects of the registration channel that can be modified by the operational environment in order to improve the channel security (cf. AGD\_PRE.1 refinement item 2).

#### **5.6.46.6.4 Security management roles (FMT\_SMR)**

##### **5.6.416.6.4.1 FMT\_SMR.2 Restrictions on security roles**

<b>FMT_SMR.2</b>	<b>Restrictions on Security Roles</b>
------------------	---------------------------------------

**FMT\_SMR.2.1** The TSF shall maintain the roles:

- *Security Administrator.*

**FMT\_SMR.2.2** The TSF shall be able to associate users with roles.

**FMT\_SMR.2.3** The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely*

are satisfied.



~~Application Note 26~~Application Note 24

FMT\_SMR.2.3 requires that a Security Administrator be able to administer the TOE through the local console and through a remote mechanism (IPsec, SSH, TLS, HTTPS).

For distributed TOEs not every TOE component is required to implement its own user management to fulfill this SFR. At least one component has to support authentication and identification of Security Administrators according to FIA\_UIA\_EXT.1 and FIA\_UAU\_EXT.2. For the other TOE components authentication as Security Administrator can be realized through the use of a trusted channel (either according to FTP\_ITC.1 or FPT\_ITT.1) from a component that supports the authentication of Security Administrators according to FIA\_UIA\_EXT.1 and FIA\_UAU\_EXT.2. The identification of users according to FIA\_UIA\_EXT.1.2 and the association of users with roles according to FMT\_SMR.2.2 is done through the components that support the authentication of Security Administrators according to FIA\_UIA\_EXT.1 and FIA\_UAU\_EXT.2. TOE components that authenticate Security Administrators through the use of a trusted channel are not required to support local administration of the component as defined in FMT\_SMR.2.3.

## **5.7.6.7 Protection of the TSF (FPT)**

This section defines requirements for the TOE to protect critical security data such as keys and passwords, to provide self-tests that monitor continued correct operation of the TOE (including detection of failures of firmware or software integrity), and to provide trusted methods for updates to the TOE firmware/software. In addition, the TOE is required to provide reliable timestamps in order to support accurate audit recording under the FAU\_GEN family.

### **~~5.7.16.7.1~~ Protection of TSF Data (Extended – FPT\_SKP\_EXT)**

#### **~~5.7.1.16.7.1.1~~ FPT\_SKP\_EXT.1 Protection of TSF Data (for reading of all symmetric keys)**

<b>FPT_SKP_EXT.1</b>	<b>Protection of TSF Data (for reading of all symmetric keys)</b>
----------------------	---

**FPT\_SKP\_EXT.1.1** The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

~~Application Note 27~~Application Note 25

*The intent of this requirement is for the device to protect keys, key material, and authentication credentials from unauthorized disclosure. This data should only be accessed for the purposes of their assigned security functionality, and there is no need for them to be displayed/accessed at any other time. This requirement does not prevent the device from providing indication that these exist, are in use, or are still valid. It does, however, restrict the reading of the values outright.*

~~5.7.26.7.2~~ **Protection of Administrator Passwords (Extended – FPT\_APW\_EXT)**

~~5.7.2.16.7.2.1~~ **FPT\_APW\_EXT.1 Protection of Administrator Passwords**

<b>FPT_APW_EXT.1</b>	<b>Protection of Administrator Passwords</b>
----------------------	--

**FPT\_APW\_EXT.1.1** The TSF shall store passwords in non-plaintext form.

**FPT\_APW\_EXT.1.2** The TSF shall prevent the reading of plaintext passwords.

~~Application Note 28~~ Application Note 26

*The intent of the requirement is that raw password authentication data are not stored in the clear, and that no user or administrator is able to read the plaintext password through “normal” interfaces. An all-powerful administrator of course could directly read memory to capture a password but is trusted not to do so. Passwords should be obscured during entry on the local console in accordance with FIA\_UAU.7.*

~~5.7.36.7.3~~ **TSF testing (Extended – FPT\_TST\_EXT)**

In order to detect some number of failures of underlying security mechanisms used by the TSF, the TSF will perform self-tests. The extent of this self-testing is left to the product developer, but a more comprehensive set of self-tests should result in a more trustworthy platform on which to develop enterprise architecture.

(For this component, selection-based requirements exist in Appendix ~~BB~~)

~~5.7.3.16.7.3.1~~ **FPT\_TST\_EXT.1 TSF Testing (Extended)**

<b>FPT_TST_EXT.1</b>	<b>TSF testing</b>
----------------------	--------------------

**FPT\_TST\_EXT.1.1** The TSF shall run a suite of the following self-tests [selection: *during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions* [assignment: *conditions under which self-tests should occur*]] to demonstrate the correct operation of the TSF: [assignment: *list of self-tests run by the TSF*].

~~Application Note 29~~ Application Note 27

*It is expected that self-tests are carried out during initial start-up (on power on). Other options should only be used if the developer can justify why they are not carried out during initial start-up. It is expected that at least self-tests for verification of the integrity of the firmware and software as well as for the correct operation of cryptographic functions necessary to fulfil the SFRs will be performed. If not all self-test are performed during start-up multiple iterations of this SFR are used with the appropriate options selected. In future versions of this cPP the suite of self-tests will be required to contain at least mechanisms for measured boot including self-tests of the components which perform the measurement.*

*Non-distributed TOEs may internally consist of several components that contribute to enforcing SFRs. Self-testing shall cover all components that contribute to enforcing SFRs and*



verification of integrity shall cover all software that contributes to enforcing SFRs on all components.

For distributed TOEs all TOE components have to perform self-tests. This does not necessarily mean that each TOE component has to carry out the same self-tests: the ST describes the applicability of the selection (i.e. when self-tests are run) and the final assignment (i.e. which self-tests are carried out) to each TOE component.

Application Note 30~~Application Note 28~~

If certificates are used by the self-test mechanism (e.g. for verification of signatures for integrity verification), certificates are validated in accordance with FIA\_X509\_EXT.1/Rev and should be selected in FIA\_X509\_EXT.2.1. Additionally, FPT\_TST\_EXT.2 must be included in the ST.

5.7.46.7.4 **Trusted Update (FPT\_TUD\_EXT)**

Failure by the Security Administrator to verify that updates to the system can be trusted may lead to compromise of the entire system. To establish trust in the source of the updates, the system can provide cryptographic mechanisms and procedures to procure the update, check the update cryptographically through the TOE-provided digital signature mechanism, and install the update on the system. While there is no requirement that this process be completely automated, guidance documentation will detail any procedures that must be performed manually, as well as the manner in which the administrator ensures that the signature on the update is valid.

(For this family, selection-based requirements exist in Appendix BB)

5.7.416.7.4.1 **FPT\_TUD\_EXT.1 Trusted Update**

<b>FPT_TUD_EXT.1</b>	<b>Trusted update</b>
----------------------	-----------------------

**FPT\_TUD\_EXT.1.1** The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

Application Note 31~~Application Note 29~~

*The version currently running (being executed) may not be the version most recently installed. For instance, maybe the update was installed but the system requires a reboot before this update will run. Therefore, it needs to be clear that the query should indicate both the most recently executed version as well as the most recently installed update.*

For a distributed TOE, the method of determining the installed versions on each component of the TOE is described in the operational guidance.

**FPT\_TUD\_EXT.1.2** The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [selection: *support automatic checking for updates, support automatic updates, no other update mechanism*].

~~Application Note 32~~Application Note 30

The selection in FPT\_TUD\_EXT.1.2 distinguishes the support of automatic checking for updates and support of automatic updates. The first option refers to a TOE that checks whether a new update is available, communicates this to the administrator (e.g. through a message during an administrator session, through log files) but requires some action by the administrator to actually perform the update. The second option refers to a TOE that checks for updates and automatically installs them upon availability.

The TSS explains what actions are involved in the TOE support when using the 'support automatic checking for updates' or 'support automatic updates' selections.

When published hash values (see FPT\_TUD\_EXT.1.3) are used to protect the trusted update mechanism, the TOE must not automatically download the update file(s) together with the hash value (either integrated in the update file(s) or separately) and automatically install the update without any active authorization by the Security Administrator, even when the calculated hash value matches the published hash value. When using published hash values to protect the trusted update mechanism, the option 'support of automatic updates' must not be used (automated checking for updates is permitted, though). The TOE may automatically download the update file(s) themselves but not to the hash value. For the published hash approach, it is intended that a Security Administrator is always required to give active authorisation for installation of an update (as described in more detail under FPT\_TUD\_EXT.1.3) below. Due to this, the type of update mechanism is regarded as 'manually initiated update', even if the update file(s) may be downloaded automatically. A fully automated approach (without Security Administrator intervention) can only be used when 'digital signature mechanism' is selected in FPT\_TUD\_EXT.1.3 below.

**FPT\_TUD\_EXT.1.3** The TSF shall provide means to authenticate firmware/software updates to the TOE using a [selection: digital signature mechanism, published hash] prior to installing those updates.

~~Application Note 33~~Application Note 31

The digital signature mechanism referenced in the selection of FPT\_TUD\_EXT.1.3 is one of the algorithms specified in FCS\_COP.1(2)/SigGen. The published hash referenced in FPT\_TUD\_EXT.1.3 is generated by one of the functions specified in FCS\_COP.1(3)/Hash... The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms.

When published hash values are used to secure the trusted update mechanism, an active authorization of the update process by the Security Administrator is always required. The secure transmission of an authentic hash value from the developer to the Security Administrator is one of the key factors to protect the trusted update mechanism when using published hashes and the guidance documentation needs to describe how this transfer has to be performed. For the verification of the trusted hash value by the Security Administrator different use cases are possible. The Security Administrator could obtain the published hash value as well as the update file(s) and perform the verification outside the TOE while the hashing of the update file(s) could be done by the TOE or by other means. Authentication as Security Administrator and initiation of the trusted update would in this case be regarded as 'active authorization' of the trusted update. Alternatively, the Administrator could provide the

*TOE with the published hash value together with the update file(s) and the hashing and hash comparison is performed by the TOE. In case of successful hash verification the TOE can perform the update without any additional step by the Security Administrator. Authentication as Security Administrator and sending the hash value to the TOE is regarded as 'active authorization' of the trusted update (in case of successful hash verification), because the administrator is expected to load the hash value only to the TOE when intending to perform the update. As long as the transfer of the hash value to the TOE is performed by the Security Administrator, loading of the update file(s) can be performed by the Security Administrator or can be automatically downloaded by the TOE from a repository.*

*If the digital signature mechanism is selected, the verification of the signature shall be performed by the TOE itself. For the published hash option, the verification can be done by the TOE itself as well as by the Security Administrator. In the latter case use of TOE functionality for the verification is not mandated, so verification could be done using non-TOE functionality of the device containing the TOE or without using the device containing the TOE.*

*For distributed TOEs all TOE components shall support Trusted Update. The verification of the signature or hash on the update shall either be done by each TOE component itself (signature verification) or for each TOE component (hash verification).*

*Updating a distributed TOE might lead to the situation where different TOE components are running different software versions. Depending on the differences between the different software versions the impact of a mixture of different software versions might be no problem at all or critical to the proper functioning of the TOE. The TSS shall detail the mechanisms that support the continuous proper functioning of the TOE during trusted update of distributed TOEs.*

#### ~~Application Note 34~~Application Note 32

---

*Future versions of this cPP will mandate the use of a digital signature mechanism for trusted updates.*

#### ~~Application Note 35~~Application Note 33

---

*If certificates are used by the update verification mechanism, certificates are validated in accordance with FIA\_X509\_EXT.1/Rev and should be selected in FIA\_X509\_EXT.2.1. Additionally, FPT\_TUD\_EXT.2 must be included in the ST.*

#### ~~Application Note 36~~Application Note 34

---

*“Update” in the context of this SFR refers to the process of replacing a non-volatile, system resident software component with another. The former is referred to as the NV image, and the latter is the update image. While the update image is typically newer than the NV image, this is not a requirement. There are legitimate cases where the system owner may want to rollback a component to an older version (e.g. when the component manufacturer releases a faulty update, or when the system relies on an undocumented feature no longer present in the update). Likewise, the owner may want to update with the same version as the NV image to recover from faulty storage.*

*All discrete software ~~components~~elements (e.g. applications, drivers, kernel, firmware) of the TSF, need to be protected, i.e. they should either be digitally signed by the corresponding manufacturer and subsequently verified by the mechanism performing the update. ~~Since it is recognized that components may be signed by different manufacturers, it is essential that the~~*

~~update process verify that both the update and NV images were produced by the same manufacturer (e.g. by comparing public keys) or signed by legitimate signing keys (e.g. successful verification of certificates when using X.509 certificates). or a hash should be published for them which needs to be verified before the update.~~

## ~~5.7.5~~6.7.5 Time stamps (FPT\_STM)

### ~~5.7.5.1~~6.7.5.1 FPT\_STM.1 Reliable Time Stamps

<b>FPT_STM.1</b>	<b>Reliable Time Stamps</b>
------------------	-----------------------------

**FPT\_STM.1.1** The TSF shall be able to provide reliable time stamps.

#### ~~Application Note 37~~Application Note 35

*Reliable time stamps are expected to be used with other TSF, e.g. for the generation of audit data to allow the Security Administrator to investigate incidents by checking the order of events and to determine the actual local time when events occurred. ~~The TSF does not provide reliable~~decision about the required level of accuracy of that information ~~about the current time at the TOE's location by itself, but~~is up to the administrator. The TOE depends on external time and date information, either provided manually by the ~~administrator~~Security Administrator or through the use of an NTP server. The use of a local real-time clock and the automatic synchronization with an NTP server is recommended but not mandated. The ST author describes in the TSS how the external time and date information is received by the TOE and how this information is maintained.*

*The term 'reliable time stamps' refers to the strict use of the time and date information, that is provided externally, and the logging of all changes to the time settings including information about the old and new time. With this information the real time for all audit data can be ~~calculated~~determined.*

*For distributed TOEs it is expected that the Security Administrator ensures synchronization between the time settings of different TOE components. All TOE components shall either be in sync (e.g. through synchronization between TOE components or through synchronization of different TOE components with external NTP servers) or the offset should be known to the administrator for every pair of TOE components. This includes TOE components synchronized to different time zones..*

## 5.86.8TOE Access (FTA)

This section specifies requirements associated with security of administration sessions carried out on the TOE. In particular, both local and remote sessions are monitored for inactivity and either locked or terminated when a threshold time period is reached. Administrators must also be able to positively terminate their own interactive sessions, and must have an advisory notice displayed at the start of each session.

~~5.8.16.8.1~~ TSF-initiated Session Locking (Extended – FTA\_SSL\_EXT)

~~5.8.1.16.8.1.1~~ FTA\_SSL\_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1	TSF-initiated Session Locking
---------------	-------------------------------

FTA\_SSL\_EXT.1.1 The TSF shall, for local interactive sessions, [selection:

- *lock the session - disable any activity of the user's data access/display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session;*
- *terminate the session]*

after a Security Administrator-specified time period of inactivity.

~~5.8.26.8.2~~ Session locking and termination (FTA\_SSL)

~~5.8.2.16.8.2.1~~ FTA\_SSL.3 TSF-initiated Termination **(Refinement)**

FTA_SSL.3	TSF-initiated Termination
-----------	---------------------------

FTA\_SSL.3.1-**Refinement**: The TSF shall terminate a **remote** interactive session after a Security Administrator-configurable time interval of session inactivity.

~~5.8.2.26.8.2.2~~ FTA\_SSL.4 User-initiated Termination **(Refinement)**

FTA_SSL.4	User-initiated Termination
-----------	----------------------------

FTA\_SSL.4.1-**Refinement**: The TSF shall allow **Administrator**-initiated termination of the **Administrator's** own interactive session.

~~5.8.36.8.3~~ TOE access banners (FTA\_TAB)

~~5.8.3.16.8.3.1~~ FTA\_TAB.1 Default TOE Access Banners **(Refinement)**

FTA_TAB.1	Default TOE Access Banners
-----------	----------------------------

FTA\_TAB.1.1-**Refinement**: Before establishing an **administrative user** session the TSF shall display a **Security Administrator-specified** advisory **notice and consent** warning message regarding use of the TOE.

~~Application Note 38~~Application Note 36

*This requirement is intended to apply to interactive sessions between a human user and a TOE. IT entities establishing connections or programmatic connections (e.g., remote procedure calls over a network) are not required to be covered by this requirement.*

## 5.9.6.9 Trusted path/channels (FTP)

To address the issues concerning transmitting sensitive data to and from the TOE, compliant TOEs will provide encryption for these communication paths between themselves and the endpoint. These channels are implemented using one (or more) of four standard protocols: IPsec, TLS, HTTPS, and SSH. These protocols are specified by RFCs that offer a variety of implementation choices. Requirements have been imposed on some of these choices (particularly those for cryptographic primitives) to provide interoperability and resistance to cryptographic attack.

In addition to providing protection from disclosure (and detection of modification) for the communications, each of the protocols described (IPsec, SSH, and TLS, HTTPS) offer two-way authentication of each endpoint in a cryptographically secure manner, meaning that even if there was a malicious attacker between the two endpoints, any attempt to represent themselves to either endpoint of the communications path as the other communicating party would be detected.

### 5.9.16.9.1 Trusted Channel (FTP\_ITC)

#### 5.9.1.16.9.1.1 FTP\_ITC.1 Inter-TSF trusted channel (~~Refined~~Refinement)

FTP_ITC.1	Inter-TSF trusted channel
-----------	---------------------------

**FTP\_ITC.1.1** The TSF shall be **capable of using [selection: *IPsec, SSH, TLS, HTTPS*] to provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [selection: *authentication server, assignment: [other capabilities]*** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.**

**FTP\_ITC.1.2** The TSF shall permit **the TSF, or the authorized IT entities** to initiate communication via the trusted channel.

**FTP\_ITC.1.3** The TSF shall initiate communication via the trusted channel for [assignment: *list of services for which the TSF is able to initiate communications*].

~~Application Note 39~~Application Note 37

*The intent of the above requirement is to provide a means by which a cryptographic protocol may be used to protect external communications with authorized IT entities that the TOE interacts with to perform its functions. The TOE uses at least one of the listed protocols for communications with the server that collects the audit information. If it communicates with an authentication server (e.g., RADIUS), then the ST author chooses "authentication server"*



in FTP\_ITC.1.1 and this connection must be capable of being protected by one of the listed protocols. If other authorized IT entities (e.g., NTP server) are protected, the ST author makes the appropriate assignments (for those entities) and selections (for the protocols that are used to protect those connections). The ST author selects the mechanism or mechanisms supported by the TOE, and then ensures that the detailed protocol requirements in Appendix ~~BB~~ corresponding to their selection are included in the ST. If TLS is selected, the ST author will claim FCS\_TLSC\_EXT.2 instead of FCS\_TLSC\_EXT.1.

While there are no requirements on the party initiating the communication, the ST author lists in the assignment for FTP\_ITC.1.3 the services for which the TOE can initiate the communication with the authorized IT entity.

The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.

Where public key certificates are used in support of an FTP ITC.1 channel, FIA X509 EXT.1/Rev is to be used (this requires checking certificate revocation), and not the iteration FIA X509 EXT.1/ITT which is only for use in inter-component channels of a distributed TOE.

## 5.9.26.9.2 Trusted Path (FTP\_TRP)

### 5.9.2.16.9.2.1 FTP\_TRP.1/Admin Trusted Path (Refinement)

FTP_TRP.1/ <u>Admin</u>	Trusted Path
-------------------------	--------------

**FTP\_TRP.1.1/Admin** The TSF shall be capable of using [selection: *IPsec, SSH, TLS, HTTPS*] to provide a communication path between itself and **authorized remote administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *disclosure and provides detection of modification of the channel data*.

**FTP\_TRP.1.2/Admin** The TSF shall permit **remote administrators** to initiate communication via the trusted path.

**FTP\_TRP.1.3/Admin** The TSF shall require the use of the trusted path for **initial administrator authentication and all remote administration actions**.

### ~~Application Note 40~~Application Note 38

*This requirement ensures that authorized remote administrators initiate all communication with the TOE via a trusted path, and that all communication with the TOE by remote administrators is performed over this path. The data passed in this trusted communication channel are encrypted as defined by the protocol chosen in the first selection. The ST author selects the mechanism or mechanisms supported by the TOE, and then ensures that the*

*detailed protocol requirements in Appendix ~~BB~~ corresponding to their selection are included in the ST.*



## 6.7. Security Assurance Requirements

This cPP identifies the Security Assurance Requirements (SARs) to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

This section lists the set of SARs from CC part 3 that are required in evaluations against this cPP. Individual Evaluation Activities to be performed are specified in [SD].

The general model for evaluation of TOEs against STs written to conform to this cPP is as follows: after the ST has been approved for evaluation, the ITSEF will obtain the TOE, supporting environmental IT (if required), and the guidance documentation for the TOE. The ITSEF is expected to perform actions mandated by the Common Evaluation Methodology (CEM) for the ASE and ALC SARs. The ITSEF also performs the Evaluation Activities contained within the SD, which are intended to be an interpretation of the other CEM assurance requirements as they apply to the specific technology instantiated in the TOE. The Evaluation Activities that are captured in the SD also provide clarification as to what the developer needs to provide to demonstrate the TOE is compliant with the cPP.

The TOE security assurance requirements are identified in [Table 3](#)~~Table 2~~.

Assurance Class	Assurance Components
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
Tests (ATE)	Independent testing – sample (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

*Table 32: Security Assurance Requirements*

### 6.17.1 ASE: Security Target

The ST is evaluated as per ASE activities defined in the CEM. In addition, there may be Evaluation Activities specified within the SD that call for necessary descriptions to be included in the TSS that are specific to the TOE technology type.

Appendix D provides a description of the information expected to be provided regarding the quality of entropy in the random bit generator.

**ASE\_TSS.1.1C Refinement:** The TOE summary specification shall describe how the TOE meets each SFR. **In the case of entropy analysis the TSS is used in conjunction with required supplementary information on Entropy.**

The requirements for exact conformance of the Security Target are described in section ~~2~~ and ~~in [SD, 3.1].2.~~

## **6.27.2ADV: Development**

The design information about the TOE is contained in the guidance documentation available to the end user as well as the TSS portion of the ST, and any required supplementary information required by this cPP that is not to be made public.

### **6.2.17.2.1 Basic Functional Specification (ADV\_FSP.1)**

The functional specification describes the TOE Security Functions Interfaces (TSFIs). It is not necessary to have a formal or complete specification of these interfaces. Additionally, because TOEs conforming to this cPP will necessarily have interfaces to the Operational Environment that are not directly invocable by TOE users, there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. For this cPP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation. No additional “functional specification” documentation is necessary to satisfy the Evaluation Activities specified in the SD.

The Evaluation Activities in the SD are associated with the applicable SFRs; since these are directly associated with the SFRs, the tracing in element ADV\_FSP.1.2D is implicitly already done and no additional documentation is necessary.

## **6.37.3AGD: Guidance Documentation**

The guidance documents will be provided with the ST. Guidance must include a description of how the IT personnel verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by the IT personnel.

Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes:

- instructions to successfully install the TSF in that environment; and
- instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and
- instructions to provide a protected administrative capability.

Guidance pertaining to particular security functionality must also be provided; requirements on such guidance are contained in the Evaluation Activities specified in the SD.

### **6.3.17.3.1 Operational User Guidance (AGD\_OPE.1)**

The operational user guidance does not have to be contained in a single document. Guidance to users, administrators and application developers can be spread among documents or web pages.

The developer should review the Evaluation Activities contained in the SD to ascertain the specifics of the guidance that the evaluator will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

### **6.3.27.3.2 Preparative Procedures (AGD\_PRE.1)**

As with the operational guidance, the developer should look to the Evaluation Activities to determine the required content with respect to preparative procedures. The refinement below describes some specific requirements for distributed TOEs.

#### **AGD\_PRE.1 Preparative Procedures**

##### **Refinement:**

The following specific topics must be addressed as part of the Preparative Procedures for the TOE:

1. If the TOE is a distributed TOE and relies on the environment to provide security for the registration process when joining components to the TOE then the Preparative Procedures shall:
  - clearly state the strength of the authentication and encryption provided by the registration channel itself and the specific requirements on the environment used for joining components to the TOE (e.g. where the environment is relied upon to prevent interception of sensitive messages, IP spoofing attempts, man-in-the-middle attacks, or race conditions)
  - identify what confidential values are transmitted over the enablement channel (e.g. any keys, their lengths, and their purposes), use of any non-confidential keys (e.g. where a developer uses the same key for more than one device or across all devices of a type or family), and use of any unauthenticated identification data (e.g. IP addresses, self-signed certificates)
  - highlight any situation in which a secret value/key may be transmitted over a channel that uses a key of lower comparable strength than the transmitted value/key. Comparable strength is defined as the amount of work required to compromise the algorithm or key and is typically expressed as ‘bits’ of security. The ST author should consult NIST 800-57 Table 2 for further guidance on comparable algorithm strength.

### **Application Note 39**

*A distributed TOE that relies on the environment in this way is one where the ST author uses the FTP TRP.1/Join channel type in the main selection for FCO CPC EXT.1.2, and the TOE relies on the operational environment to provide security for some aspects of the registration channel security (this is the case where the ST gives a reference to operational guidance in the assignment in FTP TRP.1.3/Join).*

**2. If the TOE is a distributed TOE and uses a registration channel for registering components to the TOE then the Preparative Procedures shall:**

- describe the security characteristics of the registration channel (e.g. the protocol, keys and authentication data on which it is based) and shall highlight any aspects which do not meet the requirements for a steady-state inter-component channel (as in FTP ITC.1 or FPT ITT.1)**
- identify any dependencies between the configuration of the registration channel and the security of the subsequent inter-component communications (e.g. where AES-256 inter-component communications depend on transmitting 256 bit keys between components and therefore rely on the registration channel being configured to use an equivalent key length)**
- identify any aspects of the channel can be modified by the operational environment in order to improve the channel security, and shall describe how this modification can be achieved (e.g. generating a new key pair, or replacing a default public key certificate).**

### **Application Note 40**

*This requirement deals with cases where the ST author uses the FTP ITC.1/FPT ITT.1 or FTP TRP.1/Join channel types in the main selection for FCO CPC EXT.1.2.*

*This requirement is intended to ensure that administrators can make an accurate judgement of any risks that arise from the default registration process. Examples would be the use of self-signed certificates (i.e. certificates that are not chained to an external or local Certification Authority), manufacturer-issued certificates (where control over aspects such as revocation, or which devices are issued with recognised certificates, is outside the control of the operational environment), use of generic/non-unique keys (e.g. where the same key is present on more than one instance of a device), or well-known keys (i.e. where the confidentiality of the keys is not intended to be strongly protected – note that this need not mean there is a positive action or intention to publicise the keys).*

*Where aspects of the channel are identified that would not meet FTP ITC.1 or FPT ITT.1, this implies that the ST will also have selected the FTP TRP.1/Join option in the main selection in FCO CPC EXT.1.2.*

#### **6.47.4 Class ALC: Life-cycle Support**

At the assurance level provided for TOEs conformant to this cPP, life-cycle support is limited to end-user-visible aspects of the life-cycle, rather than an examination of the TOE vendor's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it is a reflection on the information to be made available for evaluation at this assurance level.

##### **6.4.17.4.1 Labelling of the TOE (ALC\_CMC.1)**

This component is targeted at identifying the TOE such that it can be distinguished from other products or versions from the same vendor and can be easily specified when being procured by an end user. A label could consist of a "hard label" (e.g., stamped into the metal, paper label) or a "soft label" (e.g., electronically presented when queried).

The evaluator performs the CEM work units associated with ALC\_CMC.1.

##### **6.4.27.4.2 TOE CM Coverage (ALC\_CMS.1)**

Given the scope of the TOE and its associated evaluation evidence requirements, the evaluator performs the CEM work units associated with ALC\_CMS.1.

#### **6.57.5 Class ATE: Tests**

Testing is specified for functional aspects of the system as well as aspects that take advantage of design or implementation weaknesses. The former is done through the ATE\_IND family, while the latter is through the AVA\_VAN family. For this cPP, testing is based on advertised functionality and interfaces with dependency on the availability of design information. One of the primary outputs of the evaluation process is the test report as specified in the following requirements.

##### **6.5.17.5.1 Independent Testing – Conformance (ATE\_IND.1)**

Testing is performed to confirm the functionality described in the TSS as well as the guidance documentation (includes "evaluated configuration" instructions). The focus of the testing is to confirm that the requirements specified in Section 5.1.75 are being met. The Evaluation Activities in the SD identify the specific testing activities necessary to verify compliance with the SFRs. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this cPP.

#### **6.67.6 Class AVA: Vulnerability Assessment**

For the first generation of this cPP, the iTC is expected to survey open sources to discover what vulnerabilities have been discovered in these types of products and provide that content into the AVA\_VAN discussion. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. This information will be used in the development of future protection profiles.

**6.6.17.6.1 Vulnerability Survey (AVA\_VAN.1)**

[SD] Appendix A-~~in [SD]~~ provides a guide to the evaluator in performing a vulnerability analysis.

## A. Optional Requirements

As indicated in the introduction to this cPP, the baseline requirements (those that must be performed by the TOE) are contained in the body of this cPP. Additionally, there are two other types of requirements specified in Appendices AA and BB.

The first type (in this Appendix) comprises requirements that can be included in the ST, but are not mandatory in order for a TOE to claim conformance to this cPP. The second type (in Appendix BB) comprises requirements based on selections in other SFRs from the cPP: if certain selections are made, then additional requirements in that appendix will need to be included in the body of the ST (e.g., cryptographic protocols selected in a trusted channel requirement).

### A.1 Audit Events for Optional SFRs

Requirement	Auditable Events	Additional Audit Record Contents
FAU_STG.1	None.	None.
FAU_STG_EXT.2/ <u>LocSpace</u>	None.	None.
FAU_STG_ <del>EXT</del> .3/ <u>LocSpace</u>	Warning about low storage space for audit events.	None.
<u>FIA_X509_EXT.1/ITT</u>	<u>Unsuccessful attempt to validate a certificate</u>	<u>Reason for failure</u>
<u>FMT_MOF.1/Services</u>	<u>Starting and stopping of services.</u>	<u>None.</u>
FMT_MOF.1( <del>1</del> )/ <u>Audit/Functions</u>	Modification of the behaviour of the transmission of audit data to an external IT entity, <u>the handling of audit data, the TSF, the audit functionality when Local Audit Storage Space is full.</u>	None.
<u>FMT_MTD.1/CryptoKeys</u>	<u>Management of cryptographic keys.</u>	<u>None.</u>

<u>FPT_FLS.1/LocSpace</u>	<u>None.</u>	<u>None.</u>
<del>FMT_MOFFPT_ITT.1(2)/Audit</del>	<del>Modification</del> <u>Initiation</u> <del>of the behaviour</del> <u>trusted</u> <del>channel.</del>  <u>Termination</u> of the <del>handling</del> <u>trusted</u> <del>channel.</del>  <u>Failure</u> of <del>audit data</del> <u>the</u> <del>trusted</del> channel <u>functions.</u>	<del>None.</del> <u>Identification</u> <del>of the initiator and</del> <del>target of failed</del> <del>trusted channels</del> <u>establishment</u> <u>attempt.</u>
<del>FMT_MOF.1(1)/AdminAct</del> <u>FTP_TRP.1/Join</u>	<del>Modification</del> <u>Initiation</u> <del>of the behaviour</del> <u>trusted</u> <del>path.</del>  <u>Termination</u> of the <del>TSP</del> <u>trusted</u> path.  <u>Failure of the trusted</u> <u>path functions.</u>	<del>None.</del> <u>Identification</u> <del>of the claimed user</del> <u>identity.</u>
<del>FMT_MOF.1(2)/AdminAct</del>	<del>Starting and stopping</del> <del>of services.</del>	<del>None.</del>
<del>FMT_MOFFCO_CPC_EXT.1/LocSpace</del>	<del>Modification of the</del> <del>behaviour of the audit</del> <del>functionality when</del> <del>Local Audit Storage</del> <del>Space is full.</del> <u>Enabling</u> <u>communications</u> <u>between a pair of</u> <u>components.</u>  <u>Disabling</u> <u>communications</u> <u>between a pair of</u> <u>components.</u>	<del>None.</del> <u>Identities of</u> <u>the endpoints pairs</u> <u>enabled or disabled.</u>
<del>FMT_MTD.1/AdminAct</del>	<del>Modification, deletion,</del> <del>generation/import of</del> <del>cryptographic keys.</del>	<del>None.</del>
<del>FPT_FLS.1/LocSpace</del>	<del>None.</del>	<del>None.</del>

Table 43: TOE Optional SFRs and Auditable Events



#### **Application Note 41**

The audit event for FIA X509 EXT.1/ITT is based on the TOE not being able to complete the certificate validation by ensuring the following:

- the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- Verification of the digital signature of the trusted hierarchical CA
- read/access the CRL or access the OCSP server (according to selection in the ST).

If any of these checks fails, then an audit event with the failure should be written to the audit log.

## **A.2 Security Audit (FAU)**

### **A.2.1 Security audit event storage (FAU\_STG.1 & Extended – FAU\_STG\_EXT)**

Local storage space for audit data may be necessary on the TOE itself, and the TOE may then claim protection of the audit trail against unauthorised modification (including deletion) as described in FAU\_STG.1. The local storage space for audit data of a network device is also limited, and if the local storage space is exceeded then audit data might be lost. A security administrator might be interested in the number of dropped, overwritten, etc. audit records. This number might serve as an indication if a severe problem has occurred after the storage space was exceeded that continuously generated audit data. Therefore FAU\_STG\_EXT.2/LocSpace and FAU\_STG\_EXT.3/LocSpace are defined to express these optional capabilities of a network device.

#### **A.2.1.1 FAU\_STG.1 Protected audit trail storage**

<b>FAU_STG.1</b>	<b>Protected audit trail storage</b>
------------------	--------------------------------------

**FAU\_STG.1.1** The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

**FAU\_STG.1.2** The TSF shall be able to prevent unauthorised modifications to the stored audit records in the audit trail.

#### **A.2.1.2 FAU\_STG\_EXT.2/LocSpace Counting lost audit data**

<b>FAU_STG_EXT.2/<u>LocSpace</u></b>	<b>Counting lost audit data</b>
--------------------------------------	---------------------------------

**FAU\_STG\_EXT.2.1/LocSpace** The TSF shall provide information about the number of [selection: *dropped*, *overwritten*, *assignment: other information*] audit records in the case where the local storage has been filled and the TSF takes one of the actions defined in FAU\_STG\_EXT.1.3.

~~Application Note 41~~ Application Note 42

*This option should be chosen if the TOE supports this functionality.*

*In case the local storage for audit records is cleared by the administrator, the counters associated with the selection in the SFR should be reset to their initial value (most likely to 0). The guidance documentation should contain a warning for the administrator about the loss of audit data when he clears the local storage for audit records.*

*~~-For distributed TOEs each component that implements counting of lost audit data has to provide a mechanism for administrator access to, and management of, this information.~~*

*~~If FAU\_-STG\_EXT.2/LocSpace is added to the ST, the ST has to make clear any situations in which lost audit data is not counted.~~*

**A.2.1.3 FAU\_STG.3 /LocSpace Action in case of possible audit data loss**  
**Display warning for local storage space**

**FAU\_STG\_EXT.3 Display warning for local storage space**

**FAU\_STG\_EXT.3/LocSpace Action in case of possible audit data loss**

**FAU\_STG.3.1/LocSpace** The TSF shall generate a warning to inform the user if the audit trail exceeds the local audit trail storage capacity.

**Application Note 43**

*This option should be chosen if the TOE generates a warning to inform the user before the local storage space for audit data is used up. This might be useful if auditable events are stored on local storage space only.*

*It has to be ensured that the warning message required by FAU\_STG.3.1/LocSpace can be communicated to the user. The communication should be done via the audit log itself because it cannot be guaranteed that an administrative session is active at the time the event occurs.*

*~~before~~The warning should inform the user when the local space to store audit data is used up and/or the TOE will lose audit data due to insufficient local space.*

~~**Application Note 42**~~

*~~This option should be chosen if the TOE generates a warning to inform the user before the when local storage space for audit data is used up. This might be useful if auditable events are stored on local storage space only.~~*

*~~It is exhausted, it has to be ensured that the described which TOE components support this feature (not necessarily all TOE components have to support this feature if selected for the overall TOE). Each component that supports that feature shall either generate a warning itself or through another component.~~*

If FAU\_STG.3/LocSpace is added to the ST, the ST has to make clear any situations in which audit records might be 'invisibly lost'.

### **A.3 Identification and Authentication (FIA)**

#### **A.1.1 Authentication using X.509 certificates (Extended – FIA X509 EXT)**

##### **A.3.1.1 FIA X509 EXT.1 Certificate Validation**

<b><u>FIA X509 EXT.1/ITT X.509 Certificate Validation</u></b>
---

FIA X509 EXT.1.1/ITT The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280, Certificate Revocation List (CRL) as specified in RFC 5759, [assignment: list of sections which the TSF enforces], no revocation method*].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - [selection: *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field, none*].

#### **Application Note 44**

This SFR should be chosen if the TOE is distributed and FPT\_ITT.1 is chosen to protect internal TSF data between TOE entities and if ssh-rsa is not chosen in FIA\_SSHC\_EXT.1.5 or FIA\_SSHS\_EXT.1.5. If certificate revocation checking is supported, the ST author selects whether this is performed using OCSP or CRLs.

*The TSS shall describe when revocation checking is performed. It is expected that revocation checking is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device.*

*If revocation checking is not supported, the ST author should select no revocation method.*

*If 'none' is selected in relation to extendedKeyUsage then this word may be omitted in the ST to improve readability.*

**FIA X509 EXT.1.2/ITT** The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

#### Application Note 45

*This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.*

~~message required by FAU\_STG\_EXT.1.3 can be communicated to the user. The communication should be done via the audit log itself because it cannot be guaranteed that an administrative session is active at the time the event occurs.~~

### A.3A.4 Security Management (FMT)

#### A.3.1A.4.1 Management of functions in TSF (FMT\_MOF)

##### A.3.1.1A.4.1.1 FMT\_MOF.1/Services Management of security functions behaviour

<b>FMT_MOF.1(1)/<u>Audit/Services</u></b>	<b>Management of security functions behaviour</b>
---	---

**FMT\_MOF.1.1(1)/Audit/Services** The TSF shall restrict the ability to enable and disable the functions services to Security Administrators.

#### Application Note 46

*FMT\_MOF.1/Services should only be chosen if the Security Administrator has the ability to start and stop services.*

##### A.4.1.2 FMT MOF.1/Functions Management of security functions behaviour

<b>FMT MOF.1/Functions</b>	<b>Management of security functions behaviour</b>
----------------------------	---

**FMT MOF.1.1/Functions** The TSF shall restrict the ability to [selection: determine the behaviour of, modify the behaviour of] the functions [selection: transmission of audit data to

~~an external IT entity, handling of audit data, TOE Security Functions, audit functionality when Local Audit Storage Space is full]~~ to Security Administrators.

~~Application Note 43~~ **Application Note 47**

~~FMT\_MOF.1(1)/Audit/Functions~~ should ~~always~~ be chosen if one or more of the following scenarios apply:

- If the transmission protocol for transmission of audit data to an external IT entity as defined in FAU\_STG\_EXT.1.1 is configurable-, 'transmission of audit data to an external IT entity' shall be chosen.

~~FMT\_MOF.1(2)/Audit~~ **Management of security functions behaviour**

~~FMT\_MOF.1.1(2)/Audit~~ The TSF shall restrict the ability to determine the behaviour of, modify the behaviour of the functions handling of audit data to Security Administrators.

~~Application Note 44~~

- ~~FMT\_MOF.1(2)/Audit should only be chosen if~~ If the handling of audit data is configurable-, 'handling of audit data' shall be chosen. The term 'handling of audit data' refers to the different options for selection and assignments in SFRs FAU\_STG\_EXT.1.2, FAU\_STG\_EXT.1.3 and FAU\_STG\_EXT.2.-/LocSpace.

~~FMT\_MOF.1(1)/AdminAct~~ **Management of security functions behaviour**

~~FMT\_MOF.1.1(1)/AdminAct~~ The TSF shall restrict the ability to modify the behaviour of the functions TOE Security Functions to Security Administrators.

~~Application Note 45~~

- ~~FMT\_MOF.1(1)/AdminAct should only be chosen if~~ If the behaviour of the TOE Security Functions is configurable-, 'TOE Security Functions' shall be chosen.

~~Application Note 1~~ If the

~~FMT\_MOF.1(2)/AdminAct~~ **Management of security functions** behaviour

~~FMT\_MOF.1.1(2)/AdminAct~~ The TSF shall restrict the ability to enable, disable\_ of the functions services to Security Administrators.

~~Application Note 46~~

~~FMT\_MOF.1(2)/AdminAct should only be chosen if the Security Administrator has the ability to start and stop services.~~

**FMT\_MOF.1/LocSpace** Management of security functions behaviour

**FMT\_MOF.1.1/LocSpace** The TSF shall restrict the ability to ~~determine the behaviour of, modify the behaviour of the functions~~ audit functionality is configurable when Local Audit Storage Space is full ~~to Security Administrators~~.

**Application Note 47**

- ~~FMT\_MOF.1/LocSpace should only be chosen if the behaviour of the~~ 'audit functionality when Local Audit Storage Space is ~~full~~ is configurable. full' shall be chosen.

*The first selection for 'determine the behaviour of' and 'modify the behaviour of' should be done as appropriate. It might be necessary to have different selections for the first selection depending on the second selection (e.g. 'handling of audit data' might require 'determine the behaviour of' and 'modify the behaviour of' for the first selection on the one hand and 'TOE Security Functions' might require 'modify the behaviour of' only). In that case FMT\_MOF.1/Functions should be iterated with increasing number appended (i.e. FMT\_MOF.1/Functions1, FMT\_MOF.1/Functions2, etc.).*

**A.3.2A.4.2** Management of TSF data (FMT\_MTD)

**A.3.2.1A.4.2.1** FMT\_MTD.1/~~AdminAet~~CryptoKeys Management of TSF data

**FMT\_MTD.1/~~AdminAet~~CryptoKeys** Management of TSF data

**FMT\_MTD.1.1/~~AdminAet~~CryptoKeys** The TSF shall restrict the ability to ~~modify, delete, generate/import~~manage the cryptographic keys to Security Administrators.

**Application Note 48**

*FMT\_MTD.1.1/~~AdminAet~~CryptoKeys restricts management of cryptographic keys to Security administrators. It should only be chosen if cryptographic keys can be managed (e.g. modified, deleted or generated/imported) by the Security Administrator. The identifier 'CryptoKeys' has been added here to separate this iteration of FMT\_MTD.1 from the mandatory iteration of FMT\_MTD.1 defined in Chapter 6.6.2.1 (FMT\_MTD.1/CoreData).*

**A.4A.5** Protection of the TSF (FPT)

**A.4.1A.5.1** Fail Secure (FPT\_FLS)

**A.4.1.1A.5.1.1** FPT\_FLS.1/LocSpace Failure with preservation of secure state

**FPT\_FLS.1/LocSpace** Failure with preservation of secure state

**FPT\_FLS.1.1/LocSpace** The TSF shall preserve a secure state when the following types of failures occur: Local Storage Space for audit data is full.

#### **Application Note 49**

*This SFR shall be added if the TOE is configured to stop all security functions (i.e. preserving a secure state) if no more local storage space for audit data would be available. By this an attacker would not be able to hide his actions by generating additional audit events. This behaviour is expected to be modelled in FAU\_STG\_EXT.1.3 in the last assignment of the selection (i.e. 'other option').*

### **A.5.2 Internal TOE TSF data transfer (FPT ITT)**

#### **A.5.2.1 FPT ITT.1 Basic internal TSF data transfer protection (Refinement)**

<b>FPT ITT.1</b>	<b>Basic internal TSF data transfer protection</b>
------------------	--

**FPT ITT.1.1** The TSF shall protect TSF data from disclosure *and detect its modification* when it is transmitted between separate parts of the TOE **through the use of [selection: choose at least one of: IPsec, SSH, TLS, HTTPS].**

#### **Application Note 50**

*This requirement is only applicable to distributed TOEs, and ensures that all communications between components of the distributed TOE are protected through the use of an encrypted communications channel. The data passed in this trusted communication channel are encrypted as defined by the protocol chosen in the selection. The ST author should identify the channels and protocols used by each pair of communicating components in a distributed TOE, iterating this SFR as appropriate.*

*This channel may also be used as the registration channel for the registration process, as described in section 3.3 and FCO\_CPC\_EXT.1.2.*

*If TLS is selected, then the requirements to have the reference identifier established by the user (FCS\_TLSC\_EXT.1.2) are relaxed and the identifier may also be established through a "Gatekeeper" discovery process. The TSS should describe the discovery process and highlight how the reference identifier is supplied to the "joining" component.*

### **A.6 Trusted Path/Channels (FTP)**

#### **A.6.1 Trusted Path (FTP TRP)**

##### **A.6.1.1 FTP\_TRP.1/Join Trusted Path (Refinement)**

*This iteration of FTP\_TRP.1 is defined as one of the options selectable for distributed TOE component registration in FCO\_CPC\_EXT.1 (section A.7.1).*

<b>FTP_TRP.1/Join</b>	<b>Trusted Path</b>
-----------------------	---------------------

**FTP\_TRP.1.1/Join** The TSF shall provide a communication path between itself and a **joining component**~~{selection: remote, local}~~ users that is logically distinct from other communication paths and provides assured identification of **[selection: the TSF endpoint,**



both joining component and TSF endpoint] its end-points and protection of the communicated data from modification [selection: *and disclosure, none*].

**FTP TRP.1.2/Join** The TSF shall permit [selection: the TSF, **the joining component**  
~~local users, remote users~~] to initiate communication via the trusted path.

**FTP TRP.1.3/Join** The TSF shall require the use of the trusted path for *joining components to the TSF under environmental constraints identified in [assignment: reference to operational guidance]*.

### **Application Note 51**

*This SFR implements one of the types of channel identified in the main selection for FCO CPC EXT.1.2. The “joining component” in FTP TRP.1/Join is the IT entity that is attempting to join the distributed TOE by using the registration process.*

*The effect of this SFR is to require the ability for components to communicate in a secure manner while the distributed TSF is being created (or when adding components to an existing distributed TSF). When creating the TSF from the initial pair of components, either of these components may be identified as the TSF for the purposes of satisfying the meaning of “TSF” in this SFR.*

*The selection at the end of FTP TRP.1.1/Join recognises that in some cases confidentiality (i.e. protection of the data from disclosure) may not be provided by the channel. The ST author distinguishes in the TSS whether in this case the TOE relies on the environment to provide confidentiality (as part of the constraints referenced in FTP TRP.1.3/Join) or whether the registration data exchanged does not require confidentiality (in which case this assertion must be justified). If ‘none’ is selected then this word may be omitted in the ST to improve readability.*

*The assignment in FTP TRP.1.3/Join ensures that the ST highlights any specific details needed to protect the registration environment.*

*Note that when the ST uses FPT TRP.1/Join for the registration channel then this channel cannot be reused as the normal inter-component communication channel (the latter channel must meet FTP ITC.1 or FPT ITT.1).*

## **A.7 Communication (FCO)**

### **A.7.1 Communication Partner Control (FCO CPC EXT)**

The SFR in this section defines the top-level requirement for control over the way in which components are joined together under the control of a Security Administrator to create the distributed TOE (cf. section 3.3). The SFR makes use of references to other SFRs to define the lower-level characteristics of the types of channel that may be used in the registration process.



#### **A.7.1.1 FCO CPC EXT.1 Component Registration Channel Definition**

<b><u>FCO CPC EXT.1</u></b>	<b><u>Component Registration Channel Definition</u></b>
-----------------------------	---

**FCO CPC EXT.1.1** The TSF shall require a Security Administrator to enable communications between any pair of TOE components before such communication can take place.

**FCO CPC EXT.1.2** The TSF shall implement a registration process in which components establish and use a communications channel that uses [selection:

- A channel that meets the secure channel requirements in [selection: FTP ITC.1, FPT ITT.1].
- A channel that meets the secure registration channel requirements in FTP TRP.1/Join,
- No channel].

for at least TSF data.

**FCO CPC EXT.1.3** The TSF shall enable a Security Administrator to disable communications between any pair of TOE components.

#### **Application Note 52**

*This SFR is only applicable if the TOE is distributed and therefore has multiple components that need to communicate via an internal TSF channel. When creating the TSF from the initial pair of components, either of these components may be identified as the TSF for the purposes of satisfying the meaning of “TSF” in this SFR.*

*The intention of this requirement is to ensure that there is a registration process that includes a positive enablement step by an administrator before components joining a distributed TOE can communicate with the other components of the TOE and before the new component can act as part of the TSF. The registration process may itself involve communication with the joining component: many network devices use a bespoke process for this, and the security requirements for the ‘registration communication’ are then defined in FCO CPC EXT.1.2. Use of this ‘registration communication’ channel is not deemed inconsistent with the requirement of FCO CPC EXT.1.1 (i.e. the registration channel can be used before the enablement step, but only in order to complete the registration process).*

*The channel selection (for the registration channel) in FCO CPC EXT.1.2 is essentially a choice between the use of a normal secure channel that is equivalent to a channel used to communicate with external IT entities (FTP ITC.1) or existing TOE components (FPT ITT.1), or else a separate type of channel that is specific to registration (FTP TRP.1/Join). If the TOE does not require a communications channel for registration (e.g. because the registration is achieved entirely by configuration actions by an administrator at each of the components) then the main selection in FCO CPC EXT.1.2 is completed with the ‘No channel’ option.*

*If the ST author selects the FTP ITC.1/FPT ITT.1 channel type in the main selection in FCO CPC EXT.1.2 then the TSS identifies the relevant SFR iteration that specifies the channel used. If the ST author selects the FTP TRP.1/Join channel type then the TOE*

Summary Specification (possibly with support from the operational guidance) describes details of the channel and the mechanisms that it uses (and describes how the registration process ensures that the channel can only be used by the intended joiner and gatekeeper). Note that the FTP TRP.1/Join channel type may require support from security measures in the operational environment (see the definition of FTP TRP.1/Join for details).

If the ST author selects the FTP ITC.1/FPT ITT.1 channel type in the main selection in FCO CPC EXT.1.2 then the ST identifies the registration channel as a separate iteration of FTP ITC.1 or FPT ITT.1 and gives the iteration identifier (e.g. 'FPT ITT.1/Join') in an ST Application Note for FCO CPC EXT.1.

Note that the channel set up and used for registration may be adopted as a continuing internal communication channel (i.e. between different TOE components) provided that the channel meets the requirements of FTP ITC.1 or FPT ITT.1. Otherwise the registration channel is closed after use and a separate channel is used for the internal communications.

## B. Selection-Based Requirements

As indicated in the introduction to this **PPcPP**, the baseline requirements (those that must be performed by the TOE or its underlying platform) are contained in the body of this **PPcPP**. There are additional requirements based on selections in the body of the **PPcPP**: if certain selections are made, then additional requirements below will need to be included.

### B.1 Audit Events for Selection-Based SFRs

Requirement	Auditable Events	Additional Audit Record Contents
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
FCS_SSHC_EXT.1	Failure to establish an SSH session	Reason for failure
	Successful SSH rekey	Non-TOE endpoint of connection (IP Address)
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
	Successful SSH rekey	Non-TOE endpoint of connection (IP Address)
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSC_EXT.2	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.2	Failure to establish a TLS Session	Reason for failure
<u>FIA_X509_EXT.1/Rev</u>	<u>Unsuccessful attempt to validate a certificate</u>	<u>Reason for failure</u>
<u>FIA_X509_EXT.2</u>	<u>None</u>	<u>None</u>
<u>FIA_X509_EXT.3</u>	<u>None.</u>	<u>None.</u>

FPT_TST_EXT.2	Failure of self-test	Reason for failure (including identifier of invalid certificate)
FPT_TUD_EXT.2	Failure of update	Reason for failure (including identifier of invalid certificate)
FMT_MOF.1(2)/ <u>TrustedUpdate</u> <u>/AutoUpdate</u>	Enabling or Disabling automatic checking for updates or automatic updates.	None.

Table 54: Selection-Based SFRs and Auditable Events

### Application Note 53

The audit event for FIA X509 EXT.1/Rev is based on the TOE not being able to complete the certificate validation by ensuring the following:

- the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- Verification of the digital signature of the trusted hierarchical CA
- read/access the CRL or access the OCSP server (according to selection in the ST).

If any of these checks fails, then an audit event with the failure should be written to the audit log.

## **B.2 \_\_Cryptographic Support (FCS)**

**B.2.1 Cryptographic Protocols (Extended – FCS\_HTTPS\_EXT, FCS\_IPSEC\_EXT, FCS\_SSHC\_EXT, FCS\_SSHS\_EXT, FCS\_TLSC\_EXT, FCS\_TLSS\_EXT)**

### **B.2.1.1 FCS\_HTTPS\_EXT.1 HTTPS Protocol**

<b>FCS_HTTPS_EXT.1</b>	<b>HTTPS Protocol</b>
------------------------	-----------------------

**FCS\_HTTPS\_EXT.1.1** The TSF shall implement the HTTPS protocol that complies with RFC 2818.

### ~~Application Note 50~~Application Note 54

*The ST author must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done either by adding elements to this component, or by additional detail in the TSS.*

**FCS\_HTTPS\_EXT.1.2** The TSF shall implement HTTPS using TLS.

**FCS\_HTTPS\_EXT.1.3** The TSF shall [selection: not require client authentication,  
[selection: not establish the connection, request authorization to establish the connection;  
~~no~~, [assignment: other action]] if the peer certificate is deemed invalid.

~~Application Note 54~~Application Note 55

Validity~~If HTTPS is selected in FPT TRP.1/Admin or FTP ITC.1 then validity is determined~~  
~~by the identifier verification, certificate path, the expiration date, and the revocation status in~~  
~~accordance with RFC 5280. Certificate validity is tested in accordance with testing~~  
~~performed for FIA X509 EXT.1/Rev. If HTTPS is selected in FPT ITT.1 then certificate~~  
~~validity is tested in accordance with testing performed for FIA X509 EXT.1/ITT~~

### B.2.1.2 FCS\_IPSEC\_EXT.1 IPsec Protocol

The endpoints of network device communication can be geographically and logically distant and may pass through a variety of other potentially untrusted systems. The security functionality of the network device must be able to protect any critical network traffic (administration traffic, authentication traffic, audit traffic, etc.). One way to provide a mutually authenticated communication channel between the network device and an external IT entity is to implement IPsec.

IPsec is not a required component of this cPP. If a TOE implements IPsec, a corresponding selection in FTP\_ITC.1, FPT ITT.1 and/or FTP\_TRP.1/Admin should have been made that defines what the IPsec protocol is implemented to protect.

IPsec is a peer to peer protocol and as such does not need to be separated into client and server requirements.

<b>FCS_IPSEC_EXT.1</b>	<b>IPsec Protocol</b>
------------------------	-----------------------

**FCS\_IPSEC\_EXT.1.1** The TSF shall implement the IPsec architecture as specified in RFC 4301.

~~Application Note 52~~Application Note 56

*RFC 4301 calls for an IPsec implementation to protect IP traffic through the use of a Security Policy Database (SPD). The SPD is used to define how IP packets are to be handled: PROTECT the packet (e.g., encrypt the packet), BYPASS the IPsec services (e.g., no encryption), or DISCARD the packet (e.g., drop the packet). The SPD can be implemented in various ways, including router access control lists, firewall rulesets, a “traditional” SPD, etc. Regardless of the implementation details, there is a notion of a “rule” that a packet is “matched” against and a resulting action that takes place.*

*While there must be a means to order the rules, a general approach to ordering is not mandated, as long as the SPD can distinguish the IP packets and apply the rules accordingly. There may be multiple SPDs (one for each network interface), but this is not required.*

**FCS\_IPSEC\_EXT.1.2** The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

**FCS\_IPSEC\_EXT.1.3** The TSF shall implement [selection: *transport mode* ~~and [selection: *tunnel mode*, *no other mode*]~~].

**Application Note 57**

*The selection of supported modes shall be performed according to RFC 4301. The TSS shall provide details about the supported modes.*

**FCS\_IPSEC\_EXT.1.4** The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [selection: *AES-CBC-128, AES-CBC-256 (both specified by RFC 3602)* ~~and [selection: *AES-GCM-128 (specified in RFC 4106), AES-GCM-256 (specified in RFC 4106)*, *no other algorithms*]~~] together with a Secure Hash Algorithm (SHA)-based HMAC.

**FCS\_IPSEC\_EXT.1.5** The TSF shall implement the protocol: [selection:

- *IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [selection: *no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers*], and [selection: *no other RFCs for hash functions, RFC 4868 for hash functions*];*
- *IKEv2 as defined in RFC 5996 and [selection: *with no support for NAT traversal, with mandatory support for NAT traversal as specified in RFC 5996, section 2.23*]), and [selection: *no other RFCs for hash functions, RFC 4868 for hash functions*]*

].

~~Application Note 53~~ **Application Note 58**

*If the TOE implements SHA-2 hash algorithms for IKEv1 or IKEv2, the ST author selects RFC 4868. ~~If the ST author selects IKEv1, FCS\_IPSEC\_EXT.1.15 must also be included in the ST. IKEv2 will be required for those TOEs entering evaluation after Quarter 3, 2016.~~*

**FCS\_IPSEC\_EXT.1.6** The TSF shall ensure the encrypted payload in the [selection: *IKEv1, IKEv2*] protocol uses the cryptographic algorithms [selection: *AES-CBC-128, AES-CBC-256 as specified in RFC 3602* ~~and [selection: *AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm*]~~].

~~Application Note 54~~ **Application Note 59**

*AES-GCM-128 and AES-GCM-256 may only be selected if IKEv2 is also selected, as there is no RFC defining AES-GCM for IKEv1.*

**FCS\_IPSEC\_EXT.1.7** The TSF shall ensure that [selection:

- *IKEv1 Phase 1 SA lifetimes can be configured by an Security Administrator based on [selection:*
  - *number of bytes;*
  - *length of time, where the time values can **be** configured within [assignment: integer range including 24] hours;**]*

];

- *IKEv2 SA lifetimes can be configured by an Security Administrator based on*  
[selection:
  - *number of bytes;*
  - *length of time, where the time values can be configured within [assignment:*  
*integer range including 24] hours*

]

].

~~Application Note 55~~Application Note 60

*The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS\_IPSEC\_EXT.1.5). The ST author chooses either volume-based lifetimes or time-based lifetimes (or a combination). This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD\_OPE). Hardcoded limits do not meet this requirement. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the guidance documentation generated for AGD\_OPE.*

**FCS\_IPSEC\_EXT.1.8** The TSF shall ensure that [selection:

- *IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on*  
[selection:
  - *number of bytes;*
  - *length of time, where the time values can be configured within [assignment:*  
*integer range including 8] hours;*

];

- *IKEv2 Child SA lifetimes can be configured by a Security Administrator based on*  
[selection:
  - *number of bytes;*
  - *length of time, where the time values can be configured within [assignment:*  
*integer range including 8] hours;*

]

].

~~Application Note 56~~Application Note 61

*The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS\_IPSEC\_EXT.1.5). The ST author chooses either volume-based lifetimes or time-based lifetimes (or a combination). This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD\_OPE). Hardcoded limits do not meet this requirement. In general, instructions for setting the parameters of the implementation,*



including lifetime of the SAs, should be included in the guidance documentation generated for AGD\_OPE.

**FCS\_IPSEC\_EXT.1.9** The TSF shall generate the secret value  $x$  used in the IKE Diffie-Hellman key exchange (" $x$ " in  $g^x \bmod p$ ) using the random bit generator specified in FCS\_RBG\_EXT.1, and having a length of at least [assignment: (one or more) number(s) of bits that is at least twice the security strength of the negotiated Diffie-Hellman group] bits.

~~Application Note 57~~Application Note 62

For DH groups 19 and 20, the " $x$ " value is the point multiplier for the generator point  $G$ .

Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS\_IPSEC\_EXT.1.9 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 ~~is~~ 192.

**FCS\_IPSEC\_EXT.1.10** The TSF shall generate nonces used in [selection: IKEv1, IKEv2] exchanges of length [selection:

- [assignment: security strength associated with the negotiated Diffie-Hellman group];
  - at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash
- ].

~~Application Note 58~~Application Note 63

The ST author must select the second option for nonce lengths if IKEv2 is also selected (as this is mandated in RFC 5996). The ST author may select either option for IKEv1.

For the first option for nonce lengths, since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS\_IPSEC\_EXT.1.10 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.

Because nonces may be exchanged before the DH group is negotiated, the nonce used should be large enough to support all TOE-chosen proposals in the exchange.

**FCS\_IPSEC\_EXT.1.11** The TSF shall ensure that all IKE protocols implement DH Groups [selection: 14 (2048-bit MODP), ~~and [selection: 19 (256-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP),~~ [assignment: other DH groups that are implemented by the TOE], no other DH groups].



~~Application Note 59~~Application Note 64

The selection is used to specify additional DH groups supported. This applies to IKEv1 and IKEv2 exchanges. ~~For products entering into evaluation after Quarter 3, 2015, DH Group 19 (256-bit Random ECP) and DH Group 20 (384-bit Random ECP) will be required.~~ It should be noted that if any additional DH groups are specified, they must comply with the requirements (in terms of the ephemeral keys that are established) listed in FCS\_CKM.1.

**FCS\_IPSEC\_EXT.1.12** The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: IKEv1 Phase 1, IKEv2 IKE\_SA] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: IKEv1 Phase 2, IKEv2 CHILD\_SA] connection.

~~Application Note 60~~Application Note 65

The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. Obviously, the IKE version(s) chosen should be consistent not only in this element, but with other choices for other elements in this component. While it is acceptable for this capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the AGD documentation) must enable this functionality.

**FCS\_IPSEC\_EXT.1.13** The TSF shall ensure that all IKE protocols perform peer authentication using [selection: RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [selection: Pre-shared Keys, no other method].

~~Application Note 61~~Application Note 66

At least one public-key-based Peer Authentication method is required in order to conform to this **PPcPP**; one or more of the public key schemes is chosen by the ST author to reflect what is implemented. The ST author also ensures that appropriate FCS requirements reflecting the algorithms used (and key generation capabilities, if provided) are listed to support those methods. Note that the TSS will elaborate on the way in which these algorithms are to be used (for example, RFC 2409 specifies three authentication methods using public keys; each one supported will be described in the TSS). ~~Peer authentication using ECDSA X.509v3 certificates will be required for TOEs entering evaluation after Quarter 3, 2015.~~

**FCS\_IPSEC\_EXT.1.14** The TSF shall only establish a trusted channel to peers with valid certificates.

~~Application Note 62~~Application Note 67

Supported peer certificate algorithms are the same as FCS\_IPSEC\_EXT.1.~~4~~13

### B.2.1.3 FCS\_SSHC\_EXT.1 SSH Client Protocol

FCS_SSHC_EXT.1	SSH Client Protocol
----------------	---------------------

**FCS\_SSHC\_EXT.1.1** The TSF shall implement the SSH protocol that complies with ~~RFCs~~RFC(s) [selection: 4251, 4252, 4253, 4254, ~~and~~ ~~[selection~~: 5647, 5656, 6187, 6668, no other RFCs].

~~Application Note 63~~Application Note 68

The ST author selects which of the ~~additional~~ RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are “REQUIRED”. This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as “REQUIRED” but not listed in the later elements of this component are implemented is out of scope of the ~~assurance~~evaluation activity for this requirement.

**FCS\_SSHC\_EXT.1.2** The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [selection: password-based, ~~no other method~~].

**FCS\_SSHC\_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: number of bytes] bytes in an SSH transport connection are dropped.

~~Application Note 64~~Application Note 69

RFC 4253 provides for the acceptance of “large packets” with the caveat that the packets should be of “reasonable length” or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining “reasonable length” for the TOE.

**FCS\_SSHC\_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [selection: ~~aes128-cbc, aes256-cbc, [selection:—aes128-ctr, aes256-ctr, AEAD\_AES\_128\_GCM, AEAD\_AES\_256\_GCM, no other algorithms]~~].

~~Application Note 65~~Application Note 70

RFC 5647 specifies the use of the AEAD\_AES\_128\_GCM and AEAD\_AES\_256\_GCM algorithms in SSH. As described in RFC 5647, AEAD\_AES\_128\_GCM and AEAD\_AES\_256\_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. ~~In the assignment, the ST author can select the AES GCM algorithms, or “no other algorithms” if AES GCM is not supported. If AES GCM is selected, there should be corresponding~~Corresponding FCS\_COP entries are included in the ST for the algorithms selected here.

**FCS\_SSHC\_EXT.1.5** The TSF shall ensure that the SSH transport implementation uses [selection: ~~ssh-rsa, ecdsa-sha2-nistp256]~~ and [selection: ~~ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, no other public key algorithms]~~ as its public key algorithm(s) and rejects all other public key algorithms.

~~Application Note 66~~ Application Note 71

~~Implementations that select only ssh-rsa will not achieve the 112-bit security strength in the digital signature generation for SSH authentication as is recommended in NIST SP 800-131A. Future versions of this profile may remove ssh-rsa as a selection. If x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384 or x509v3-ecdsa-sha2-nistp521 are selected, then the list of trusted certification authorities must be selected in FCS\_SSHC\_EXT.1.9 and the FIA\_X509\_EXT SFRs in Appendix B are applicable.~~

**FCS\_SSHC\_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512] and [selection: AEAD\_AES\_128\_GCM, AEAD\_AES\_256\_GCM, no other MAC algorithms] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

~~Application Note 67~~ Application Note 72

RFC 5647 specifies the use of the AEAD\_AES\_128\_GCM and AEAD\_AES\_256\_GCM algorithms in SSH. As described in RFC 5647, AEAD\_AES\_128\_GCM and AEAD\_AES\_256\_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH.

**FCS\_SSHC\_EXT.1.7** The TSF shall ensure that [selection: diffie-hellman-group14-sha1, ecdh-sha2-nistp256] and [selection: ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.

**FCS\_SSHC\_EXT.1.8** The TSF shall ensure that within SSH connections the SSH connection be rekeyed after same session keys are used for a threshold of no longer than one hour, and no more than ~~2<sup>28</sup> packets have been~~ one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

[\*\*Note for public review: this requires that products check and act on both such thresholds.]

Application Note 73

This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold the total incoming and outgoing data needs to be counted. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.

It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.

For any configurable threshold related to this requirement the guidance documentation needs to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.

**FCS\_SSHC\_EXT.1.9** The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or [selection: *a list of trusted certification authorities, no other methods*] as described in RFC 4251 section 4.1.

~~Application Note 68~~Application Note 74

*The list of trusted certification authorities can only be selected if x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384 or x509v3-ecdsa-sha2-~~nistp384~~nistp521 are selected in FCS\_SSHC\_EXT.1.5.*

**B.2.1.4 FCS\_SSHS\_EXT.1 SSH Server Protocol**

FCS_SSHS_EXT.1	SSH Server Protocol
----------------	---------------------

**FCS\_SSHS\_EXT.1.1** The TSF shall implement the SSH protocol that complies with ~~RFCs~~RFC(s) [selection: 4251, 4252, 4253, 4254, ~~and~~ [selection: 5647, 5656, 6187, 6668, no other RFCs].

~~Application Note 69~~Application Note 75

*The ST author selects which of the **additional** RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are “REQUIRED”. This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as “REQUIRED” but not listed in the later elements of this component are implemented is out of scope of the assurance evaluation activity for this requirement.*

**FCS\_SSHS\_EXT.1.2** The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

**FCS\_SSHS\_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: *number of bytes*] bytes in an SSH transport connection are dropped.

~~Application Note 70~~Application Note 76

*RFC 4253 provides for the acceptance of “large packets” with the caveat that the packets should be of “reasonable length” or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining “reasonable length” for the TOE.*

**FCS\_SSHS\_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: ~~aes128-~~[selection: aes123-cbc, aes256-cbc, [selection: ~~aes128-ctr,~~ aes256-ctr, AEAD\_AES\_128\_GCM, AEAD\_AES\_256\_GCM, no other algorithms].

~~Application Note 71~~Application Note 77

RFC 5647 specifies the use of the AEAD\_AES\_128\_GCM and AEAD\_AES\_256\_GCM algorithms in SSH. As described in RFC 5647, AEAD\_AES\_128\_GCM and AEAD\_AES\_256\_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. ~~In the assignment, the ST author can select the AES-GCM algorithms, or "no other algorithms" if AES-GCM is not supported. If AES-GCM is selected, there should be corresponding FCS\_COP entries in the ST. Corresponding FCS\_COP entries are included in the ST for the algorithms selected here.~~

**FCS\_SSHS\_EXT.1.5** The TSF shall ensure that the SSH transport implementation uses [selection: ssh-rsa, ecdsa-sha2-nistp256] and [selection: ecdsa-sha2-nistp384, ~~x509v3-ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, no other public key algorithms~~] as its public key algorithm(s) and rejects all other public key algorithms.

~~Application Note 72~~Application Note 78

~~Implementations that select only ssh-rsa will not achieve the 112-bit security strength in the digital signature generation for SSH authentication as is recommended in NIST SP 800-131A. Future versions of this profile may remove ssh-rsa as a selection. If x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384 or x509v3-ecdsa-sha2-nistp521 are selected then the FIA\_X509\_EXT SFRs in Appendix B are applicable~~

**FCS\_SSHS\_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512] and [selection: AEAD\_AES\_128\_GCM, AEAD\_AES\_256\_GCM, no other MAC algorithms] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

~~Application Note 73~~Application Note 79

RFC 5647 specifies the use of the AEAD\_AES\_128\_GCM and AEAD\_AES\_256\_GCM algorithms in SSH. As described in RFC 5647, AEAD\_AES\_128\_GCM and AEAD\_AES\_256\_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH.

**FCS\_SSHS\_EXT.1.7** The TSF shall ensure that [selection: diffie-hellman-group14-sha1, ecdh-sha2-nistp256] and [selection: ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.

**FCS\_SSHS\_EXT.1.8** The TSF shall ensure that within SSH connections the ~~SSH connection~~ be rekeyed after same session keys are used for a threshold of no longer than one hour, and no more than ~~2<sup>28</sup> packets have been~~ one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

Application Note 80

This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data



*threshold the total incoming and outgoing data needs to be counted. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.*

*It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.*

*For any configurable threshold related to this requirement the guidance documentation needs to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.*

#### B.2.1.5 FCS\_TLSC\_EXT.1 TLS Client Protocol

TLS is not a required component of this cPP. If a TOE implements TLS, a corresponding selection in FTP\_ITC.1, [FPT\\_ITT.1](#) or FTP\_TRP.1/[Admin](#) should be made to define what the TLS protocol is implemented to protect.

A TOE may act as the client, the server, or both in TLS sessions. The requirement has been separated into TLS Client (FCS\_TLSC\_EXT) and TLS Server (FCS\_TLSS\_EXT) requirements to allow for these differences. If the TOE acts as the client during the claimed TLS sessions, the ST author should claim one of the FCS\_TLSC\_EXT requirements.

Additionally, TLS may or may not be performed with client authentication. The ST author shall claim FCS\_TLSC\_EXT.1 and FCS\_TLSS\_EXT.1 if the TOE does not support client authentication. The ST author should claim FCS\_TLSC\_EXT.2 and FCS\_TLSS\_EXT.2 if client authentication is performed by the TOE. If TLS is selected as a means to provide a trusted communication channel for an external IT entity in FTP\_ITC.1, then FCS\_TLSC\_EXT.2 is required.

<b>FCS_TLSC_EXT.1</b>	<b>TLS Client Protocol</b>
-----------------------	----------------------------

**FCS\_TLSC\_EXT.1.1** The TSF shall implement [selection: *TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] supporting the following ciphersuites:

- ~~Mandatory~~*Optional* Ciphersuites: [selection:
  - *TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 3268*
- [selection: ~~Optional~~ Ciphersuites:
  - *TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 3268*
  - *TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 3268*
  - *TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 3268*
  - *TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 4492*
  - *TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 4492*
  - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 4492*
  - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 4492*

- *TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5246*
- *TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256 as defined in RFC 5246*
- *TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5246*
- *TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256 as defined in RFC 5246*
- *TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5288*
- *TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5288*
- *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5289*
- *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384 as defined in RFC 5289*
- *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289*
- *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289*
- *TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289*
- *TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289*
- *TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5289*
- *TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384 as defined in RFC 5289*
- *No other ciphersuite*

L

#### Application Note 81

~~**Application Note 74**—~~*The ciphersuites to be tested in the evaluated configuration are limited by this requirement.*

~~*The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then “None” should be selected. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA is required in order to ensure compliance with RFC 5246.*~~

*These requirements will be revisited as new TLS versions are standardized by the IETF.*

*In a future version of this cPP TLS v1.2 will be required for all TOEs.*

**FCS\_TLSC\_EXT.1.2** The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

#### ~~Application Note 75~~Application Note 82

*The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.*

*The preferred method for verification is the Subject Alternative Name using DNS names, URI names, or Service Names. Verification using the Common Name is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name or Subject Alternative name is discouraged as against best practices but may be implemented. Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the assurance evaluation activity.*

**FCS\_TLSC\_EXT.1.3** The TSF shall only establish a trusted channel if the peer certificate is valid.

~~Application Note 76~~Application Note 83

~~Validity~~*If TLS is selected in FPT\_TRP.1/Admin or FTP\_ITC then validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity is tested in accordance with testing performed for FIA\_X509\_EXT.1-/Rev. If TLS is selected in FPT\_ITT then certificate validity is tested in accordance with testing performed for FIA\_X509\_EXT.1/ITT*

**FCS\_TLSC\_EXT.1.4** The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1, ~~or~~ none] and no other curves.

~~Application Note 77~~Application Note 84

*If ciphersuites with elliptic curves were selected in FCS\_TLSC\_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS\_TLS\_EXT.1.1, then 'none' should be selected.*

*This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS\_COP.1(2)/SigGen and FCS\_CKM.1 and FCS\_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.*

## **B.2.1.6 FCS\_TLSC\_EXT.2 TLS Client Protocol with authentication**

(See introductory text in section B.2.1.5~~B.2.1.5~~)

<b>FCS_TLSC_EXT.2</b>	<b>TLS Client Protocol with authentication</b>
-----------------------	--

**FCS\_TLSC\_EXT.2.1** The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

- ~~Mandatory~~Optional Ciphersuites: [selection:
  - TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 3268
- ~~[selection: Optional Ciphersuites:~~
  - TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 3268
  - TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 3268
  - TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 3268
  - TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 4492



- *TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 4492*
  - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 4492*
  - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 4492*
  - *TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5246*
  - *TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256 as defined in RFC 5246*
  - *TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5246*
  - *TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256 as defined in RFC 5246*
  - ~~○ *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5289*~~
  - ~~○ *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384 as defined in RFC 5289*~~
  - ~~○ *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289*~~
  - ~~○ *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289*~~
  - ~~*TLS\_ECDHE*~~*TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289*~~5288~~
  - ~~*TLS\_ECDHE*~~*TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289*~~5288~~
  - ~~*TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5289*~~
  - ~~*TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384 as defined in RFC 5289*~~
  - ~~*TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289*~~
  - ~~*TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289*~~
  - ~~*TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289*~~
  - ~~*TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289*~~
  - ~~*no TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5289*~~
  - ~~*TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384 as defined in RFC 5289*~~
  - ~~*No other ciphersuite*~~
- L

---

#### Application Note 85

~~Application Note 78~~Application Note 1—The ciphersuites to be tested in the evaluated configuration are limited by this requirement.

~~The ciphersuites to be tested in the evaluated configuration are limited by this requirement.~~  
~~The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then “None” should be selected.~~ It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. The Suite B algorithms listed above (RFC 6460) are the preferred algorithms for implementation. *TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA* is required in order to ensure compliance with RFC 5246.

*These requirements will be revisited as new TLS versions are standardized by the IETF.*

*In a future version of this cPP TLS v1.2 will be required for all TOEs.*

**FCS\_TLSC\_EXT.2.2** The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

~~Application Note 79~~Application Note 86

*The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.*

*The preferred method for verification is the Subject Alternative Name using DNS names, URI names, or Service Names. Verification using the Common Name is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name or Subject Alternative name is discouraged as against best practices but may be implemented. Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the assurance evaluation activity.*

**FCS\_TLSC\_EXT.2.3** The TSF shall only establish a trusted channel if the peer certificate is valid.

~~Application Note 80~~Application Note 87

~~Validity~~If TLS is selected in FPT TRP.1/Admin or FTP ITC then validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity shall be tested in accordance with testing performed for FIA\_X509\_EXT.1.~~/Rev.~~If TLS is selected in FPT\_ITT then certificate validity is tested in accordance with testing performed for FIA\_X509\_EXT.1/ITT

**FCS\_TLSC\_EXT.2.4** -The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [selection: *secp256r1*, *secp384r1*, *secp521r1*, ~~or~~ *none*] and no other curves.

~~Application Note 81~~Application Note 88

*If ciphersuites with elliptic curves were selected in FCS\_TLSC\_EXT.2.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS\_TLS\_EXT.2.1, then 'none' should be selected.*

*This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS\_COP.1~~(2)~~SigGen and FCS\_CKM.1 and FCS\_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.*

**FCS\_TLSC\_EXT.2.5** The TSF shall support mutual authentication using X.509v3 certificates.

~~Application Note 82~~ Application Note 89

The use of X.509v3 certificates for TLS is addressed in FIA\_X509\_EXT.2.1. This requirement adds that the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.

### B.2.1.7 FCS\_TLSS\_EXT.1 TLS Server Protocol

As discussed in section B.2.1.5~~B.2.1.5~~, the TOE may act as the client, the server, or both in TLS sessions. If the TOE acts as the server during the claimed TLS sessions (FTP\_ITC.1, FPT\_ITT.1 or FTP\_TRP.1/Admin), the ST author should claim one of the FCS\_TLSS\_EXT claims.

TLS may or may not be performed with mutual authentication. The ST author shall claim FCS\_TLSS\_EXT.1 if the TOE does not support mutual authentication. The ST author should claim FCS\_TLSS\_EXT.2 if mutual authentication is supported by the TOE.

<b>FCS_TLSS_EXT.1</b>	<b>TLS Server Protocol</b>
-----------------------	----------------------------

**FCS\_TLSS\_EXT.1.1** The TSF shall implement [selection: *TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346)*] supporting the following ciphersuites:

- ~~Mandatory~~Optional Ciphersuites: [selection:
  - *TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 3268*
- ~~[selection: Optional Ciphersuites:~~
  - *TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 3268*
  - *TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 3268*
  - *TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 3268*
  - *TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 4492*
  - *TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 4492*
  - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 4492*
  - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 4492*
  - *TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5246*
  - *TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256 as defined in RFC 5246*
  - *TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5246*
  - *TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256 as defined in RFC 5246*
  - ~~◦ *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5289*~~
  - ~~◦ *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384 as defined in RFC 5289*~~
  - ~~◦ *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289*~~
  - ~~◦ *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289*~~
  - ~~◦ *TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289*~~
  - ~~◦ *TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289*~~
  - *TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5288*
  - *TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5288*
  - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256 as defined in RFC 5289*

- TLS ECDHE ECDSA WITH AES 256 CBC SHA384 as defined in RFC 5289
- TLS ECDHE ECDSA WITH AES 128 GCM SHA256 as defined in RFC 5289
- TLS ECDHE ECDSA WITH AES 256 GCM SHA384 as defined in RFC 5289
- TLS ECDHE RSA WITH AES 128 GCM SHA256 as defined in RFC 5289
- TLS ECDHE RSA WITH AES 256 GCM SHA384 as defined in RFC 5289
- TLS ECDHE RSA WITH AES 128 CBC SHA256 as defined in RFC 5289
- TLS ECDHE RSA WITH AES 256 CBC SHA384 as defined in RFC 5289

~~○ No other ciphersuite.~~

#### ~~Application Note 83~~ Application Note 1

- ~~The ciphersuites to be tested in the evaluated configuration are limited by this requirement.~~

L

#### Application Note 90

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; ~~if there are no ciphersuites supported other than the mandatory suites, then “None” should be selected.~~ It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA is required in order to ensure compliance with RFC 5246.

These requirements will be revisited as new TLS versions are standardized by the IETF.

In a future version of this cPP TLS v1.2 will be required for all TOEs.

**FCS\_TLSS\_EXT.1.2** The TSF shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0; and [selection: TLS 1.1, TLS 1.2, none].

#### ~~Application Note 84~~ Application Note 91

All SSL versions and TLS v1.0 are denied. Any TLS versions not selected in FCS\_TLSS\_EXT.1.1 should be selected here. (If “none” is the selection for this element then the ST author may omit the words “and none”).

**FCS\_TLSS\_EXT.1.3** The TSF shall generate key establishment parameters using RSA with key size ~~2048 bits and~~ [selection: 2048 bits, 3072 bits, 4096 bits, no other size] and [selection: over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; Diffie-Hellman parameters of size 2048 bits and [selection: 3072 bits, no other size]; no other].

#### ~~Application Note 85~~ Application Note 92

If the ST lists a DHE or ECDHE ciphersuite in FCS\_TLSS\_EXT.1.1, the ST must include the Diffie-Hellman or NIST curves selection in the requirement. FMT\_SMF.1 requires the configuration of the key agreement parameters in order to establish the security strength of the TLS connection.

### B.2.1.8 FCS\_TLSS\_EXT.2 TLS Server Protocol with mutual authentication

(See introductory text in section B.2.1.7~~B.2.1.7~~)

FCS_TLSS_EXT.2	TLS Server Protocol with mutual authentication
----------------	--

**FCS\_TLSS\_EXT.2.1** The TSF shall implement [selection: *TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346)*] supporting the following ciphersuites:

- ~~Mandatory~~Optional Ciphersuites: [selection:
    - *TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA* as defined in RFC 3268
  - ~~[selection: Optional Ciphersuites:~~
    - *TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA* as defined in RFC 3268
    - *TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA* as defined in RFC 3268
    - *TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA* as defined in RFC 3268
    - *TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA* as defined in RFC 4492
    - *TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA* as defined in RFC 4492
    - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA* as defined in RFC 4492
    - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA* as defined in RFC 4492
    - *TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA256* as defined in RFC 5246
    - *TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256* as defined in RFC 5246
    - *TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256* as defined in RFC 5246
    - *TLS\_DHE\_RSA\_WITH\_AES\_256\_CBC\_SHA256* as defined in RFC 5246
    - *TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256* as defined in RFC 5288
    - *TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384* as defined in RFC 5288
    - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA256* as defined in RFC 5289
    - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA384* as defined in RFC 5289
    - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256* as defined in RFC 5289
    - *TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384* as defined in RFC 5289
    - *TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256* as defined in RFC 5289
    - *TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384* as defined in RFC 5289
    - ~~*TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA256* as defined in RFC 5289~~
    - *TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384* as defined in RFC 5289
    - No other ciphersuite
- ].

#### ~~Application Note 86~~Application Note 93

*The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; ~~if there are no ciphersuites supported other than the mandatory suites, then “None” should be selected.~~ It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. The Suite B algorithms listed above (RFC 6460) are the preferred algorithms for implementation. *TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA* is required in order to ensure compliance with RFC 5246.*



*These requirements will be revisited as new TLS versions are standardized by the IETF.*

*In a future version of this cPP TLS v1.2 will be required for all TOEs.*

**FCS\_TLSS\_EXT.2.2** The TSF shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0, ~~TLS 1.0~~, and [selection: TLS 1.1, TLS 1.2, none].

~~Application Note 87~~Application Note 94

*All SSL versions and TLS v1.0 shall be denied. Any TLS versions not selected in FCS\_TLSS\_EXT.2.1 should be selected here.*

**FCS\_TLSS\_EXT.2.3** The TSF shall generate key establishment parameters using RSA with key size ~~2048 bits and~~ [selection: 2048 bits, 3072 bits, 4096 bits, no other size] and [selection: over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; Diffie-Hellman parameters of size 2048 bits and [selection: 3072 bits, no other size]; no other].

~~Application Note 88~~Application Note 95

*If the ST lists a DHE or ECDHE ciphersuite in FCS\_TLSS\_EXT.2.1, the ST must include the Diffie-Hellman or NIST curves selection in the requirement. FMT\_SMF.1 requires the configuration of the key agreement parameters in order to establish the security strength of the TLS connection.*

**FCS\_TLSS\_EXT.2.4** The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

**FCS\_TLSS\_EXT.2.5** The TSF shall not establish a trusted channel if the peer certificate is invalid.

~~Application Note 89~~Application Note 96

*The use of X.509v3 certificates for TLS is addressed in FIA\_X509\_EXT.2.1. This requirement adds that this use must include support for client-side certificates for TLS mutual authentication.*

~~Validity~~If TLS is selected for FPT\_TRP or FPT\_ITC then validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity shall be tested in accordance with testing performed for FIA\_X509\_EXT.1-/Rev. If TLS is selected for FPT\_ITT then certificate validity is tested in accordance with testing performed for FIA\_X509\_EXT.1/ITT

**FCS\_TLSS\_EXT.2.6** The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

~~Application Note 90~~Application Note 97

*The peer identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the Domain Name, IP address, username, or email address used by the peer, or may be passed to a directory server for comparison. Matching should be performed by a bit-wise comparison.*

## **B.3 Identification and Authentication (FIA)**

### **B.3.1 Authentication using X.509 certificates (Extended – FIA X509 EXT)**

X.509 certificate-based authentication is required if IPsec or TLS communications are claimed for FTP ITC.1 or FPT ITT. These SFRs are also required if FPT TUD EXT.2 or FPT TST EXT.2 are claimed. If SSH client communications are claimed and any x509 algorithms are claimed in FCS SSHC EXT.1.5 or FCS SSHS EXT.1.5, these SFRs are required. In the case of the TOE only acting as the SSH server or acting as the client, but not claiming any x509 algorithms in FCS SSHC EXT.1.5 or FCS SSHS EXT.1.5, these SFRs are optional.

#### **B.3.1.1 FIA X509 EXT.1 X.509 Certificate Validation**

<b><u>FIA X509 EXT.1/Rev X.509 Certificate Validation</u></b>
---

**FIA X509 EXT.1.1/Rev** The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280, Certificate Revocation List (CRL) as specified in RFC 5759, [assignment: list of sections which TSF enforces]* ].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

**FIA X509 EXT.1.2/Rev** The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

### Application Note 98

*FIA X509 EXT.1.1/Rev lists the rules for validating certificates. The ST author selects whether revocation status is verified using OCSP or CRLs. The trusted channel/path protocols may require that certificates are used; this use requires that the extendedKeyUsage rules are verified.*

*The validation is expected to end in a trusted root CA certificate in a root store managed by the platform.*

*The TSS shall describe when revocation checking is performed. It is expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device.*

*It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).*

*FIA X509 EXT.1.2/Rev applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.*

*The ST author must include FIA X509 EXT.1/Rev in all instances except when only SSH is selected within FTP ITC.1 or FPT ITT.1 and ssh-rsa authentication is also selected. Additionally, FIA X509 EXT.1/Rev must also be included if either FPT TUD EXT or FPT TST EXT have selected to use X509 certificates.*

### B.3.1.2 FIA X509 EXT.2 X.509 Certificate Authentication

<u>FIA X509 EXT.2</u>	<u>X.509 Certificate Authentication</u>
-----------------------	---

FIA X509 EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection: IPsec, TLS, HTTPS, SSH], and [selection: code signing for system software updates, code signing for integrity verification, [assignment: other uses], no additional uses].

FIA X509 EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate].

### Application Note 99

*In FIA X509 EXT.2.1, the ST author's selection includes IPsec, TLS, or HTTPS if these protocols are included in FTP ITC.1.1 or FPT ITT.1. SSH should be included if authentication other than ssh-rsa is selected in FCS SSHC EXT.1.5 or FCS SSHS EXT.1.5. Certificates may optionally be used for trusted updates of system software (FPT TUD EXT.2) and for integrity verification (FPT TST EXT.2).*

*Often a connection must be established to check the revocation status of a certificate - either to download a CRL or to perform a lookup using OCSP. In FIA X509 EXT.2.2 the selection*



*is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA\_X509\_EXT.1, the behavior indicated in the selection determines the validity. The TOE must not accept the certificate if it fails any of the other validation rules in FIA\_X509\_EXT.1. If the administrator-configured option is selected by the ST Author, the ST Author also selects the corresponding function in FMT\_SMF.1. The selection should be consistent with the validation requirements in FCS\_IPSEC\_EXT.1.14, FCS\_TLSC\_EXT.1.3 and FCS\_TLSC\_EXT.2.3. If a connection is not required to determine validity, that is that no revocation method is selected then this SFR is trivially satisfied.*

*The ST author must include FIA\_X509\_EXT.2 in all instances except when only SSH is selected within FTP\_ITC.1 or FPT\_ITT.1 and ssh-rsa authentication is also selected. Additionally, FIA\_X509\_EXT.2 must also be included if either FPT\_TUD\_EXT or FPT\_TST\_EXT have selected X509 certificates.*

### **B.3.1.3 FIA\_X509\_EXT.3 X.509 Certificate Requests**

#### **FIA\_X509\_EXT.3 X.509 Certificate Requests**

**FIA\_X509\_EXT.3.1** The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: device-specific information, Common Name, Organization, Organizational Unit, Country].

#### **Application Note 100**

*The public key is the public key portion of the public-private key pair generated by the TOE as specified in FCS\_CKM.1.*

**FIA\_X509\_EXT.3.2** The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

### **B.3B.4 Protection of the TSF (FPT)**

#### **B.3.1B.4.1 TSF self test (Extended)**

##### **B.3.1.1B.4.1.1 FPT\_TST\_EXT.2 Self tests based on certificates**

#### **FPT\_TST\_EXT.2 Self tests based on certificates**

**FPT\_TST\_EXT.2.1** The TSF shall fail self-testing if a certificate is used for self tests and the corresponding certificate is deemed invalid.

~~Application Note 91~~ Application Note 101

*Certificates may optionally be used for self-tests (FPT\_TST\_EXT.1.1). This element must be included in the ST if certificates are used for self-tests. If “code signing for integrity verification” is selected in FIA\_X509\_EXT.2.1, FPT\_TST\_EXT.2 must be included in the ST.*

*Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with FIA\_X509\_EXT.1/Rev.*

~~B.3.2~~ B.4.2 **Trusted Update (FPT\_TUD\_EXT)**

~~B.3.2.1~~ B.4.2.1 **FPT\_TUD\_EXT.2 Trusted Update based on certificates**

<b>FPT_TUD_EXT.2</b>	<b>Trusted Update based on certificates</b>
----------------------	---

**FPT\_TUD\_EXT.2.1** The TSF shall not install an update if the code signing certificate is deemed invalid.

**FPT\_TUD\_EXT.2.2** When the certificate is deemed invalid because the certificate has expired, the TSF shall [selection: *allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

~~Application Note 92~~ Application Note 102

*Certificates may optionally be used for code signing of system software updates (FPT\_TUD\_EXT.1.3). This element must be included in the ST if certificates are used for validating updates. If “code signing for system software updates” is selected in FIA\_X509\_EXT.2.1, FPT\_TUD\_EXT.2 must be included in the ST. The use of X.509 certificates is not applicable if only published hashes are supported for trusted updates.*

*Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with FIA\_X509\_EXT.1/Rev. For expired certificates the author of the ST selects whether the certificate shall be accepted, rejected or the choice is left to the administrator to accept or reject the certificate.*

~~B.4~~ B.5 **Security Management (FMT)**

~~B.4.1~~ B.5.1 **Management of functions in TSF (FMT\_MOF)**

~~B.4.1.1~~ B.5.1.1 **FMT\_MOF.1(2)/TrustedUpdate/AutoUpdate Management of security functions behaviour**

<b>FMT_MOF.1(2)/TrustedUpdate/AutoUpdate</b>	<b>Management of security functions behaviour</b>
--	---

**FMT\_MOF.1.1(2)/TrustedUpdate/AutoUpdate** The TSF shall restrict the ability to enable, disable the functions [selection: *automatic checking for updates, automatic update*] to Security Administrators.

~~Application Note 93~~Application Note 103

*FMT\_MOF.1(2)~~TrustedUpdate~~/AutoUpdate* is only applicable if the TOE supports automatic checking for updates and/or automatic updates and allows ~~to enable and disable them; to be enabled and disabled.~~ Enable and disable of automatic checking for updates and/or automatic updates is restricted to Security Administrators. The option “automatic update” may only be selected if digital signatures are used to validate the trusted update.

## C. Extended Component Definitions

This appendix contains the definitions for the extended requirements that are used in the cPP, including those used in Appendices A and B.

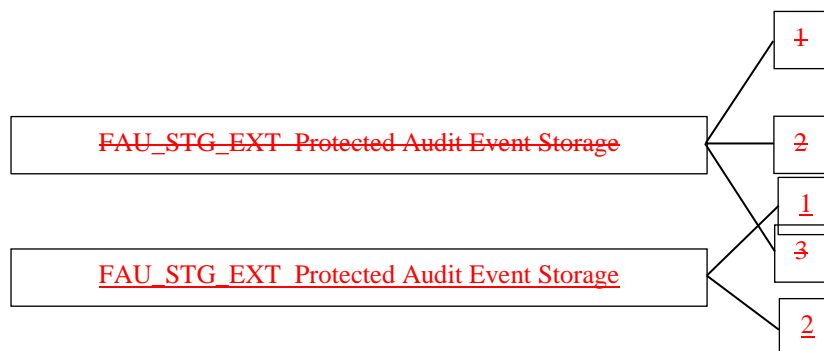
### C.1 Security Audit (FAU)

#### C.1.1 Protected audit event storage (FAU\_STG\_EXT)

##### Family Behaviour

This component defines the requirements for the TSF to be able to securely transmit audit data between the TOE and an external IT entity.

##### Component leveling



FAU\_STG\_EXT.1 Protected audit event storage requires the TSF to use a trusted channel implementing a secure protocol.

FAU\_STG\_EXT.2 Counting lost audit data requires the TSF to provide information about audit records affected when the audit log becomes full.

~~FAU\_STG\_EXT.3 Display warning for local storage space requires the TSF to generate a warning before the audit log becomes full.~~

**Management:** FAU\_STG\_EXT.1, FAU\_STG\_EXT.2, ~~FAU\_STG\_EXT.3~~

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

- a) The TSF shall have the ability to configure the cryptographic functionality.

**Audit:** FAU\_STG\_EXT.1, FAU\_STG\_EXT.2, ~~FAU\_STG\_EXT.3~~

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

- a) No audit necessary.

#### C.1.1.1 FAU\_STG\_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1	Protected Audit Event Storage
---------------	-------------------------------

Hierarchical to: No other components.

Dependencies: FAU\_GEN.1 Audit data generation  
FTP\_ITC.1 Inter-TSF Trusted Channel

**FAU\_STG\_EXT.1.1** The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP\_ITC.

~~Application Note 94~~Application Note 104

*For selecting the option of transmission of generated audit data to an external IT entity the TOE relies on a non-TOE audit server for storage and review of audit records. The storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment in that case. Since the external audit server is not part of the TOE, there are no requirements on it except the capabilities for ITC transport for audit data. No requirements are placed upon the format or underlying protocol of the audit data being transferred. The TOE must be capable of being configured to transfer audit data to an external IT entity without administrator intervention. Manual transfer would not meet the requirements. Transmission could be done in real-time or periodically. If the transmission is not done in real-time then the TSS describes what event stimulates the transmission to be made and what range of frequencies the TOE supports for making transfers of audit data to the audit server; the TSS also suggests typical acceptable frequencies for the transfer.*

*For distributed TOEs each component must be able to export audit data across a protected channel external (FTP ITC.1) or intercomponent (FTP ITT.1 or FTP ITC.1) as appropriate. At least one component of the TOE must be able to export audit records via FTP\_ITC.1 such that all TOE audit records can be exported to an external IT entity.*

**FAU\_STG\_EXT.1.2** The TSF shall be able to store generated audit data on the TOE itself.

**FAU\_STG\_EXT.1.3** The TSF shall [selection: drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]] when the local storage space for audit data is full.

~~Application Note 95~~Application Note 105

*The external log server might be used as alternative storage space in case the local storage space is full. The 'other action' could in this case be defined as 'send the new audit data to an external IT entity'.*

*For distributed TOEs each component must provide some amount of local storage to ensure that audit records are preserved in case of network connectivity issues. The behaviour when local storage is exhausted must be described for each component.*

### C.1.1.2 FAU\_STG\_EXT.2 Counting lost audit data

FAU_STG_EXT.2	Counting lost audit data
---------------	--------------------------

Hierarchical to: No other components.

Dependencies: FAU\_GEN.1 Audit data generation  
FAU\_STG\_EXT.1 External Audit Trail Storage

**FAU\_STG\_EXT.2.1** The TSF shall provide information about the number of [selection: *dropped, overwritten, assignment: other information*] audit records in the case where the local storage has been filled and the TSF takes one of the actions defined in FAU\_STG\_EXT.1.3.

~~Application Note 96~~Application Note 106

*This option should be chosen if the TOE supports this functionality.*

*In case the local storage for audit records is cleared by the administrator, the counters associated with the selection in the SFR should be reset to their initial value (most likely to 0). The guidance documentation should contain a warning for the administrator about the loss of audit data when he clears the local storage for audit records.*

~~*-For distributed TOEs each component that implements counting of lost audit data has to provide a mechanism for administrator access to, and management of, this information.*~~

### ~~C.1.1.3~~FAU\_STG\_EXT.3 ~~Display warning for local storage space~~

<del>FAU_STG_EXT.3</del>	<del>Display warning for local storage space</del>
--------------------------	--

~~**FAU\_STG\_EXT.3.1** The TSF shall generate a warning to inform 2 is added to the user before ST, the local space to store audit data is used up and/or the TOE will lose ST has to make clear any situations in which lost audit data due to insufficient local space is not counted.~~

~~Application Note 97~~Application Note 1

~~*This option should be chosen if the TOE generates a warning to inform the user before the local storage space for audit data is used up. This might be useful if auditable events are stored on local storage space only.*~~

~~*It has to be ensured that the warning message required by FAU\_STG\_EXT.1.3 can be communicated to the user. The communication should be done via the audit log itself because it cannot be guaranteed that an administrative session is active at the time the event occurs.*~~

## C.2 Cryptographic Support (FCS)

### C.2.1 Random Bit Generation (FCS\_RBG\_EXT)

#### C.2.1.1 FCS\_RBG\_EXT.1 Random Bit Generation

##### Family Behaviour

Components in this family address the requirements for random bit/number generation. This is a new family ~~define~~ defined for the FCS class.

##### Component ~~leveling~~ levelling



FCS\_RBG\_EXT.1 Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source.

##### Management: FCS\_RBG\_EXT.1

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

- a) There are no management activities foreseen

##### Audit: FCS\_RBG\_EXT.1

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

- a) Minimal: failure of the randomization process

FCS_RBG_EXT.1	Random Bit Generation
---------------	-----------------------

Hierarchical to: No other components

Dependencies: No other components

**FCS\_RBG\_EXT.1.1** The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [selection: *Hash\_DRBG (any)*, *HMAC\_DRBG (any)*, *CTR\_DRBG (AES)*].

**FCS\_RBG\_EXT.1.2** The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [selection: [assignment: *number of software-based sources*] *software-based noise source*, [assignment: *number of hardware-based sources*] *hardware-based noise source*] with minimum of [selection: *128 bits*, *192 bits*, *256 bits*] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

~~Application Note 98~~Application Note 107

*For the first selection in FCS\_RBG\_EXT.1.2, the ST selects at least one of the types of noise sources. If the TOE contains multiple noise sources of the same type, the ST author fills the assignment with the appropriate number for each type of source (e.g., 2 software-based noise sources, 1 hardware-based noise source). The documentation and tests required in the Evaluation Activity for this element necessarily describes each source indicated in the ST.*

*ISO/IEC 18031:2011 contains three different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used, and include the specific underlying cryptographic primitives used in the requirement. While any of the identified hash functions (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash\_DRBG or HMAC\_DRBG, only AES-based implementations for CTR\_DRBG are allowed.*

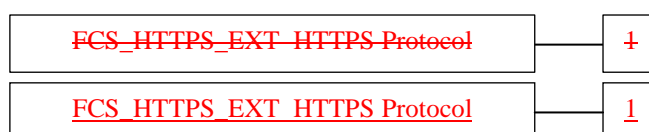
## C.2.2 Cryptographic Protocols (Extended – FCS\_HTTPS\_EXT, FCS\_IPSEC\_EXT, FCS\_SSHC\_EXT, FCS\_SSHS\_EXT, FCS\_TLSC\_EXT, FCS\_TLSS\_EXT)

### C.2.2.1 FCS\_HTTPS\_EXT.1 HTTPS Protocol

#### Family Behaviour

Components in this family define the requirements for protecting remote management sessions between the TOE and a Security Administrator. This family describes how HTTPS will be implemented. This is a new family defined for the FCS Class.

#### Component leveling



FCS\_HTTPS\_EXT.1 HTTPS requires that HTTPS be implemented according to RFC 2818 and supports TLS.

#### Management: FCS\_HTTPS\_EXT.1

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

- a) There are no management activities foreseen.

#### Audit: FCS\_HTTPS\_EXT.1

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

- a) There are no auditable events foreseen.



<b>FCS_HTTPS_EXT.1</b>	<b>HTTPS Protocol</b>
------------------------	-----------------------

Hierarchical to: No other components

Dependencies: [FCS\_TLSC\_EXT.1 TLS Client Protocol, or FCS\_TLSS\_EXT.1 TLS Server Protocol]

**FCS\_HTTPS\_EXT.1.1** The TSF shall implement the HTTPS protocol that complies with RFC 2818.

**FCS\_HTTPS\_EXT.1.2** The TSF shall implement the HTTPS protocol using TLS.

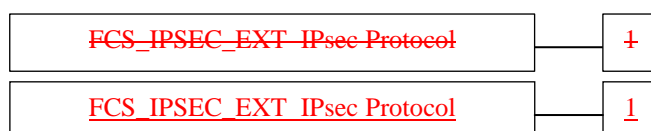
**FCS\_HTTPS\_EXT.1.3** The TSF shall [selection: *not establish the connection, request authorization to establish the connection*, ~~+~~ assignment: other action] if the peer certificate is deemed invalid.

### C.2.2.2 FCS\_IPSEC\_EXT.1 IPsec Protocol

#### Family Behaviour

Components in this family address the requirements for protecting communications using IPsec.

#### Component ~~leveling~~levelling



**FCS\_IPSEC\_EXT.1** IPsec requires that IPsec be implemented as specified.

#### Management: FCS\_IPSEC\_EXT.1

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

- a) Maintenance of SA lifetime configuration

#### Audit: FCS\_IPSEC\_EXT.1

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

- a) Decisions to DISCARD, BYPASS, PROTECT network packets processed by the TOE.
- b) Failure to establish an IPsec SA
- c) IPsec SA establishment
- d) IPsec SA termination

- e) Negotiation “down” from an IKEv2 to IKEv1 exchange.

<b>FCS_IPSEC_EXT.1</b>	<b>Internet Protocol Security (IPsec) Communications</b>
Hierarchical to:	No other components
Dependencies:	<p>FCS_CKM.1 Cryptographic Key Generation</p> <p>FCS_CKM.2 Cryptographic Key Establishment</p> <p>FCS_COP.1<del>(4)</del>/<u>DataEncryption</u> Cryptographic operation (AES Data encryption/decryption)</p> <p>FCS_COP.1<del>(2)</del>/<u>SigGen</u> Cryptographic operation (Signature <u>Generation and Verification</u>)</p> <p>FCS_COP.1<del>(3)</del>/<u>Hash</u> Cryptographic <del>Operation</del><u>operation</u> (Hash Algorithm)</p> <p><u>FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)</u></p> <p>FCS_RBG_EXT.1 Random Bit Generation</p>

**FCS\_IPSEC\_EXT.1.1** The TSF shall implement the IPsec architecture as specified in RFC 4301.

~~*Application Note 99*~~ *Application Note 108*

*RFC 4301 calls for an IPsec implementation to protect IP traffic through the use of a Security Policy Database (SPD). The SPD is used to define how IP packets are to be handled: PROTECT the packet (e.g., encrypt the packet), BYPASS the IPsec services (e.g., no encryption), or DISCARD the packet (e.g., drop the packet). The SPD can be implemented in various ways, including router access control lists, firewall rulesets, a “traditional” SPD, etc. Regardless of the implementation details, there is a notion of a “rule” that a packet is “matched” against and a resulting action that takes place.*

*While there must be a means to order the rules, a general approach to ordering is not mandated, as long as the SPD can distinguish the IP packets and apply the rules accordingly. There may be multiple SPDs (one for each network interface), but this is not required.*

**FCS\_IPSEC\_EXT.1.2** The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

**FCS\_IPSEC\_EXT.1.3** The TSF shall implement ~~transport mode and~~ [selection: *tunnel mode*, ~~no other~~transport mode].

**FCS\_IPSEC\_EXT.1.4** The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [selection: *AES-CBC-128, AES-CBC-256 (both specified by RFC 3602)* ~~and [selection:]~~ *AES-GCM-128 (specified in RFC 4106), AES-GCM-256 (specified in RFC 4106), no other algorithms*] together with a Secure Hash Algorithm (SHA)-based HMAC.

**FCS\_IPSEC\_EXT.1.5** The TSF shall implement the protocol: [selection:

- *IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [selection: no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions];*
- *IKEv2 as defined in RFCs 5996 [selection: with no support for NAT traversal, with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]].*

**FCS\_IPSEC\_EXT.1.6** The TSF shall ensure the encrypted payload in the [selection: *IKEv1, IKEv2*] protocol uses the cryptographic algorithms [selection: AES-CBC-128, AES-CBC-256 as specified in RFC 3602~~and [selection: AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm]~~.

~~*Application Note 100*~~*Application Note 109*

*AES-GCM-128 and AES-GCM-256 may only be selected if IKEv2 is also selected, as there is no RFC defining AES-GCM for IKEv1.*

**FCS\_IPSEC\_EXT.1.7** The TSF shall ensure that [selection:

- *IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on [selection:*
  - *number of bytes;*
  - *length of time, where the time values can configured within [assignment: integer range including 24] hours;**];*
- *IKEv2 SA lifetimes can be configured by a Security Administrator based on [selection:*
  - *number of bytes;*
  - *length of time, where the time values can configured within [assignment: integer range including 24] hours*

*]*

*].*

~~*Application Note 101*~~*Application Note 110*

*The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS\_IPSEC\_EXT.1.5). The ST author chooses either volume-based lifetimes or time-based lifetimes (or a combination). This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD\_OPE). Hardcoded limits do not meet this requirement. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the guidance documentation generated for AGD\_OPE.*

**FCS\_IPSEC\_EXT.1.8** The TSF shall ensure that [selection:

- *IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [selection:*
  - *number of bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 8] hours;**];*
- *IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [selection:*
  - *number of bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 8] hours;**]*

].

~~Application Note 102~~Application Note 111

*The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS\_IPSEC\_EXT.1.5). The ST author chooses either volume-based lifetimes or time-based lifetimes (or a combination). This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD\_OPE). Hardcoded limits do not meet this requirement. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the guidance documentation generated for AGD\_OPE.*

**FCS\_IPSEC\_EXT.1.9** The TSF shall generate the secret value  $x$  used in the IKE Diffie-Hellman key exchange (" $x$ " in  $g^x \bmod p$ ) using the random bit generator specified in FCS\_RBG\_EXT.1, and having a length of at least [assignment: (one or more) number(s) of bits that is at least twice the security strength of the negotiated Diffie-Hellman group] bits.

~~Application Note 103~~Application Note 112

*For DH groups 19 and 20, the "x" value is the point multiplier for the generator point G.*

*Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS\_IPSEC\_EXT.1.9 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.*

**FCS\_IPSEC\_EXT.1.10** The TSF shall generate nonces used in [selection: IKEv1, IKEv2] exchanges of length [selection:

- *[assignment: security strength associated with the negotiated Diffie-Hellman group];*

- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash  
].

~~Application Note 104~~Application Note 113

*The ST author must select the second option for nonce lengths if IKEv2 is also selected (as this is mandated in RFC 5996). The ST author may select either option for IKEv1.*

*For the first option for nonce lengths, since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS\_IPSEC\_EXT.1.10 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 “Recommendation for Key Management –Part 1: General” to determine the security strength (“bits of security”) associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.*

*Because nonces may be exchanged before the DH group is negotiated, the nonce used should be large enough to support all TOE-chosen proposals in the exchange.*

**FCS\_IPSEC\_EXT.1.11** The TSF shall ensure that all IKE protocols implement DH Groups [selection: 14 (2048-bit MODP), ~~and [selection: 19 (256-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP),~~ [assignment: other DH groups that are implemented by the TOE], no other DH groups].

**FCS\_IPSEC\_EXT.1.12** The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: IKEv1 Phase 1, IKEv2 IKE\_SA] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: IKEv1 Phase 2, IKEv2 CHILD\_SA] connection.

~~Application Note 105~~Application Note 114

*The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. While it is acceptable for this capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the AGD documentation) must enable this functionality.*

**FCS\_IPSEC\_EXT.1.13** The TSF shall ensure that all IKE protocols perform peer authentication using [selection: RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [selection: Pre-shared Keys, no other method].

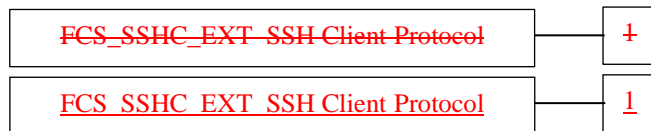
**FCS\_IPSEC\_EXT.1.14** The TSF shall only establish a trusted channel to peers with valid certificates.

### C.2.2.3 FCS\_SSHC\_EXT.1 SSH Client

#### Family Behaviour

The component in this family addresses the ability for a client to use SSH to protect data between the client and a server using the SSH protocol.

### Component ~~leveling~~levelling



FCS\_SSHC\_EXT.1 SSH Client requires that the client side of SSH be implemented as specified.

### Management: FCS\_SSHC\_EXT.1

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

- a) There are no management activities foreseen.

### Audit: FCS\_SSHC\_EXT.1

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

- a) Failure of SSH session establishment.
- b) SSH session establishment
- c) SSH session termination

<b>FCS_SSHC_EXT.1</b>	<b>SSH Client Protocol</b>
-----------------------	----------------------------

Hierarchical to: No other components

Dependencies: FCS\_CKM.1 Cryptographic Key Generation  
FCS\_CKM.2 Cryptographic Key Establishment  
FCS\_COP.1 ~~(+)~~ DataEncryption Cryptographic operation (AES Data encryption/decryption)  
FCS\_COP.1 ~~(2)~~ SigGen Cryptographic operation (Signature Generation and Verification)  
FCS\_COP.1 ~~(3)~~ Hash Cryptographic ~~Operation~~ operation (Hash Algorithm)  
FCS\_COP.1 KeyedHash Cryptographic operation (Keyed Hash Algorithm)  
FCS\_RBG\_EXT.1 Random Bit Generation

**FCS\_SSHC\_EXT.1.1** The TSF shall implement the SSH protocol that complies with RFCs [selection: 4251, 4252, 4253, 4254, and [selection: 5647, 5656, 6187, 6668, no other RFCs].



~~Application Note 106~~Application Note 115

The ST author selects which of the ~~additional~~ RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are “REQUIRED”. This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as “REQUIRED” but not listed in the later elements of this component are implemented is out of scope of the ~~assurance~~evaluation activity for this requirement.

**FCS\_SSHC\_EXT.1.2** The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [selection: password-based-, no other method].

**FCS\_SSHC\_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: *number of bytes*] bytes in an SSH transport connection are dropped.

~~Application Note 107~~Application Note 116

RFC 4253 provides for the acceptance of “large packets” with the caveat that the packets should be of “reasonable length” or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining “reasonable length” for the TOE.

**FCS\_SSHC\_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [assignment: *List of encryption algorithms*].

**FCS\_SSHC\_EXT.1.5** The TSF shall ensure that the SSH transport implementation uses [assignment: *List of public key algorithms*] as its public key algorithm(s) and rejects all other public key algorithms.

**FCS\_SSHC\_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [assignment: *List of data integrity MAC algorithms*] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS\_SSHC\_EXT.1.7** The TSF shall ensure that [assignment: *List of key exchange methods*] are the only allowed key exchange methods used for the SSH protocol.

**FCS\_SSHC\_EXT.1.8** The TSF shall ensure that within SSH connections the ~~SSH connection be rekeyed after~~ same session keys are used for a threshold of no longer than one hour, and no more than ~~2^28 packets have been~~ one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

Application Note 117

This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold the total incoming and outgoing data needs to be counted. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.

*It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.*

*For any configurable threshold related to this requirement the guidance documentation needs to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.*

**FCS\_SSHC\_EXT.1.9** The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding ~~public key or~~ [selection: public key, a list of trusted certification authorities, no other methods] as described in RFC 4251 section 4.1.

~~Application Note 108~~Application Note 118

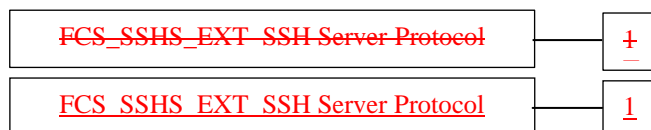
*The list of trusted certification authorities can only be selected if x509v3-ecdsa-sha2-nistp256 or x509v3-ecdsa-sha2-nistp384 are specified in FCS\_SSHC\_EXT.1.5.*

## **C.2.2.4 FCS\_SSHS\_EXT.1 SSH Server Protocol**

### **Family Behaviour**

The component in this family addresses the ability for a server to offer SSH to protect data between a client and the server using the SSH protocol.

### **Component ~~leveling~~levelling**



FCS\_SSHS\_EXT.1 SSH Server requires that the server side of SSH be implemented as specified.

### **Management: FCS\_SSHS\_EXT.1**

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

- a) There are no management activities foreseen.

### **Audit: FCS\_SSHS\_EXT.1**

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

- a) Failure of SSH session establishment.
- b) SSH session establishment
- c) SSH session termination



<b>FCS_SSHS_EXT.1</b>	<b>SSH Server Protocol</b>
-----------------------	----------------------------

Hierarchical to: No other components

Dependencies: [FCS\\_CKM.1 Cryptographic Key Generation](#)  
[FCS\\_CKM.2 Cryptographic Key Establishment](#)  
[FCS\\_COP.1\(4\)/DataEncryption](#) Cryptographic operation (AES Data encryption/decryption)  
[FCS\\_COP.1\(2\)/SigGen](#) Cryptographic operation (Signature Generation and Verification)  
[FCS\\_COP.1\(3\)/Hash](#) Cryptographic ~~Operation~~operation (Hash Algorithm)  
[FCS\\_COP.1/KeyedHash](#) Cryptographic operation (Keyed Hash Algorithm)  
[FCS\\_RBG\\_EXT.1 Random Bit Generation](#)

**FCS\_SSHS\_EXT.1.1** The TSF shall implement the SSH protocol that complies with RFCs [\[selection: 4251, 4252, 4253, 4254, and \[selection: 5647, 5656, 6187, 6668, no other RFCs\]](#).

~~Application Note 109~~ Application Note 119

*The ST author selects which of the ~~additional~~ RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are “REQUIRED”. This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as “REQUIRED” but not listed in the later elements of this component are implemented is out of scope of the ~~assurance evaluation~~ activity for this requirement.*

**FCS\_SSHS\_EXT.1.2** The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

**FCS\_SSHS\_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [\[assignment: number of bytes\]](#) bytes in an SSH transport connection are dropped.

~~Application Note 110~~ Application Note 120

*RFC 4253 provides for the acceptance of “large packets” with the caveat that the packets should be of “reasonable length” or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining “reasonable length” for the TOE.*

**FCS\_SSHS\_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [\[assignment: encryption algorithms\]](#).

**FCS\_SSHS\_EXT.1.5** The TSF shall ensure that the SSH transport implementation uses [assignment: *List of public key algorithms*] as its public key algorithm(s) and rejects all other public key algorithms.

**FCS\_SSHS\_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [assignment: *List of MAC algorithms*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS\_SSHS\_EXT.1.7** The TSF shall ensure that [assignment: *List of key exchange methods*] are the only allowed key exchange methods used for the SSH protocol.

**FCS\_SSHS\_EXT.1.8** The TSF shall ensure that within SSH connections the ~~SSH connection~~ ~~be rekeyed after~~ same session keys are used for a threshold of no longer than one hour, and no more than ~~2<sup>28</sup> packets have been~~ one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

#### Application Note 121

This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using ~~that~~ the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold the total incoming and outgoing data needs to be counted. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.

It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.

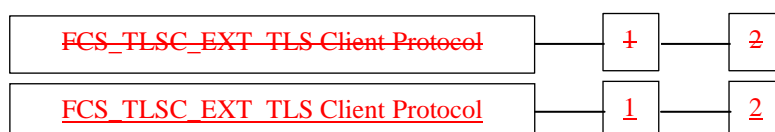
For any configurable threshold related to this requirement the guidance documentation needs to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.

### C.2.2.5 FCS\_TLSC\_EXT TLS Client Protocol

#### Family Behaviour

The component in this family addresses the ability for a client to use TLS to protect data between the client and a server using the TLS protocol.

#### Component ~~level~~levelling



FCS\_TLSC\_EXT.1 TLS Client requires that the client side of TLS be implemented as specified.

FCS\_TLSC\_EXT.2 TLS Client requires that the client side of the TLS implementation include mutual authentication.

### Management: FCS\_TLSC\_EXT.1, FCS\_TLSC\_EXT.2

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

- a) There are no management activities foreseen.

### Audit: FCS\_TLSC\_EXT.1, FCS\_TLSC\_EXT.2

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

- a) Failure of TLS session establishment.
- b) TLS session establishment
- c) TLS session termination

FCS_TLSC_EXT.1	TLS Client Protocol
----------------	---------------------

Hierarchical to: No other components

Dependencies:

- FCS\_CKM.1 Cryptographic Key Generation
- FCS\_CKM.2 Cryptographic Key Establishment
- FCS\_COP.1~~(4)~~/DataEncryption Cryptographic operation (AES Data encryption/decryption)
- FCS\_COP.1~~(2)~~/SigGenSigGen Cryptographic operation (Signature Generation and Verification)
- FCS\_COP.1~~(3)~~/Hash Cryptographic ~~Operation~~operation (Hash Algorithm)
- FCS\_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
- FCS\_RBG\_EXT.1 Random Bit Generation

**FCS\_TLSC\_EXT.1.1** The TSF shall implement [selection: *TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346)*] supporting the following ciphersuites: [selection:

- ~~• Mandatory Ciphersuites:~~
  - ~~◦ [assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]~~
- ~~• [selection: Optional Ciphersuites:~~
  - ~~◦ [assignment: List of optional ciphersuites and reference to RFC in which each is defined]~~ Mandatory Ciphersuites:
    - [assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]

- [selection: Optional Ciphersuites:

- ⊖ ~~—~~

- [assignment: List of optional ciphersuites and reference to RFC in which each is defined];

- ⊖ no other ciphersuite~~—~~111.

~~Application Note 111~~Application Note 122

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA is required in order to ensure compliance with RFC 5246.

**FCS\_TLSC\_EXT.1.2** The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

~~Application Note 112~~Application Note 123

The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

**FCS\_TLSC\_EXT.1.3** The TSF shall only establish a trusted channel if the peer certificate is valid.

~~Application Note 113~~Application Note 124

Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

**FCS\_TLSC\_EXT.1.4** The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [assignment: List of supported curves including an option for 'none'].

~~Application Note 114~~Application Note 125

If ciphersuites with elliptic curves were selected in FCS\_TLSC\_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS\_TLS\_EXT.1.1, then 'none' should be selected.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS\_COP.1~~(2)~~SigGen and FCS\_CKM.1 and FCS\_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

## FCS\_TLSC\_EXT.2

## TLS Client Protocol with Authentication

Hierarchical to: FCS\_TLSC\_EXT.1 TLS Client Protocol

Dependencies:

- FCS\_CKM.1 Cryptographic Key Generation
- FCS\_CKM.2 Cryptographic Key Establishment
- FCS\_COP.1(4) DataEncryption Cryptographic operation (AES Data encryption/decryption)
- FCS\_COP.1(2) SigGen Cryptographic operation (Signature Generation and Verification)
- FCS\_COP.1(3) Hash Cryptographic ~~Operation~~operation (Hash Algorithm)
- FCS\_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
- FCS\_RBG\_EXT.1 Random Bit Generation
- ~~FCS\_RBG\_EXT.1 Random Bit Generation~~

**FCS\_TLSC\_EXT.2.1** The TSF shall implement [selection: *TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346)*] supporting the following ciphersuites: [selection:

- Mandatory Ciphersuites:
  - [assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]
- [selection: Optional Ciphersuites:
- [assignment: List of optional ciphersuites and reference to RFC in which each is defined] ~~Mandatory Ciphersuites:~~
  - ~~[assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]~~
- ~~[selection: Optional Ciphersuites:~~
  - ~~[assignment: List of optional ciphersuites and reference to RFC in which each is defined]~~
  - 1;
  - ~~no other ciphersuite~~ 1.1.1.

### ~~Application Note 115~~ Application Note 126

*The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA is required in order to ensure compliance with RFC 5246.*

**FCS\_TLSC\_EXT.2.2** The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

### ~~Application Note 116~~ Application Note 127

*The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server),*

or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

**FCS\_TLSC\_EXT.2.3** The TSF shall only establish a trusted channel if the peer certificate is valid.

~~Application Note 117~~Application Note 128

Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

**FCS\_TLSC\_EXT.2.4** The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [assignment: List of supported curves including an option for 'none'].

If ciphersuites with elliptic curves were selected in FCS\_TLSC\_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS\_TLS\_EXT.1.1, then 'none' should be selected.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS COP.1/SigGen and FCS CKM.1 and FCS CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

**FCS\_TLSC\_EXT.2.5** The TSF shall support mutual authentication using X.509v3 certificates.

~~Application Note 118~~Application Note 129

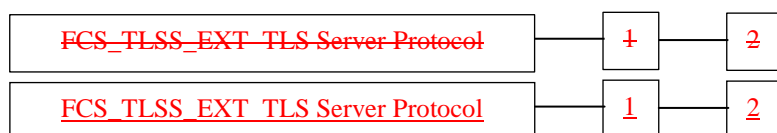
The use of X.509v3 certificates for TLS is addressed in FIA\_X509\_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.

## C.2.2.6 FCS\_TLSS\_EXT TLS Server Protocol

### Family Behaviour

The component in this family addresses the ability for a server to use TLS to protect data between a client and the server using the TLS protocol.

### Component leveling



FCS\_TLSS\_EXT.1 TLS Server requires that the server side of TLS be implemented as specified.

FCS\_TLSS\_EXT.2: TLS Server requires the mutual authentication be included in the TLS implementation.

### Management: FCS\_TLSS\_EXT.1, FCS\_TLSS\_EXT.2

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

- a) There are no management activities foreseen.

### Audit: FCS\_TLSS\_EXT.1, FCS\_TLSS\_EXT.2

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

- a) Failure of TLS session establishment.
- b) TLS session establishment
- c) TLS session termination

FCS_TLSS_EXT.1	TLS Server Protocol
----------------	---------------------

Hierarchical to: No other components

Dependencies:

- FCS\_CKM.1 Cryptographic Key Generation
- FCS\_CKM.2 Cryptographic Key Establishment
- FCS\_COP.1~~(4)~~/DataEncryption Cryptographic operation (AES Data encryption/decryption)
- FCS\_COP.1~~(2)~~/SigGen Cryptographic operation (Signature Generation and Verification)
- FCS\_COP.1~~(3)~~/Hash Cryptographic ~~Operation~~operation (Hash Algorithm)
- FCS\_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
- FCS\_RBG\_EXT.1 Random Bit Generation
- ~~FCS\_RBG\_EXT.1 Random Bit Generation~~

**FCS\_TLSS\_EXT.1.1** The TSF shall implement [selection: *TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346)*] supporting the following ciphersuites: [selection:

- *Mandatory Ciphersuites:*
  - *[assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]*
- *[selection: Optional Ciphersuites:*



- ~~[assignment: List of optional ciphersuites and reference to RFC in which each is defined]~~ Mandatory Ciphersuites:
  - ~~[assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]~~
- ~~[selection: Optional Ciphersuites:~~
  - ~~[assignment: List of optional ciphersuites and reference to RFC in which each is defined]~~
  - 1:
  - ~~no other ciphersuite~~ H.111.

~~Application Note 119~~ Application Note 130

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA is required in order to ensure compliance with RFC 5246.

**FCS\_TLSS\_EXT.1.2** The TSF shall deny connections from clients requesting ~~SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and~~ ~~[selection: TLS 1.1, TLS 1.2, none]~~ [assignment: list of protocol versions to deny].

~~Application Note 120~~ Application Note 131

Any TLS versions not selected in FCS\_TLSS\_EXT.1.1 should be selected here.

**FCS\_TLSS\_EXT.1.3** The TSF shall generate key establishment parameters using RSA with key size ~~2048 bits and~~ ~~[selection: 2048 bits, 3072 bits, 4096 bits, no other size]~~ and ~~[selection: [assignment: List of elliptic curves]; [assignment: List of diffie-hellman~~ Diffie-Hellman ~~parameter sizes]]~~.

~~Application Note 121~~ Application Note 132

The assignments will be filled in based on the assignments performed in FCS\_TLSS\_EXT.1.1.

FCS_TLSS_EXT.2	TLS Server Protocol with mutual authentication
Hierarchical to:	FCS_TLSS_EXT.1 TLS Server Protocol
Dependencies:	FCS_CKM.1 Cryptographic Key Generation <u>FCS_CKM.2 Cryptographic Key Establishment</u> FCS_COP.1 <del>(4)/DataEncryption</del> Cryptographic operation (AES Data encryption/decryption) FCS_COP.1 <del>(2)/SigGen</del> Cryptographic operation (Signature <u>Generation and Verification</u> ) FCS_COP.1 <del>(3)/Hash</del> Cryptographic <del>Operation</del> <u>operation</u> (Hash Algorithm) <del>FCS_RBG_EXT.1 Random Bit Generation</del> <u>FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)</u> <u>FCS_RBG_EXT.1 Random Bit Generation</u>

**FCS\_TLSS\_EXT.2.1** The TSF shall implement [selection: *TLS 1.2 (RFC 5246)*, *TLS 1.1 (RFC 4346)*] supporting the following ciphersuites: [selection:

- *Mandatory Ciphersuites:*
  - *[assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]*
- *[selection: Optional Ciphersuites:*
  - *[assignment: List of optional ciphersuites and reference to RFC in which each is defined]*~~*Mandatory Ciphersuites:*~~
    - ~~*[assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]*~~
  - *[selection: Optional Ciphersuites:*
    - ~~*[assignment: List of optional ciphersuites and reference to RFC in which each is defined]*~~
    - *1:*
    - ~~*no other ciphersuite*~~*1.1.1.*

~~*Application Note 122*~~*Application Note 133*

*The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA is required in order to ensure compliance with RFC 5246.*

**FCS\_TLSS\_EXT.2.2** The TSF shall deny connections from clients requesting ~~SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and~~ [selection: *TLS 1.1*, *TLS 1.2*, ~~none~~] [assignment: list of protocol versions to deny].

~~*Application Note 123*~~*Application Note 134*

*Any TLS versions not selected in FCS\_TLSS\_EXT.2.1 should be selected here.*

**FCS\_TLSS\_EXT.2.3** The TSF shall generate key establishment parameters using RSA with key size ~~2048 bits and~~ [selection: *2048 bits*, *3072 bits*, *4096 bits*, *no other size*] and [selection: *[assignment: List of elliptic curves]*; *[assignment: List of* ~~*diffie-hellman*~~*Diffie-Hellman* *parameter sizes]*].

~~*Application Note 124*~~*Application Note 135*

*The assignments will be filled in based on the assignments performed in FCS\_TLSS\_EXT.2.1.*

**FCS\_TLSS\_EXT.2.4** The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

~~*Application Note 125*~~*Application Note 136*

*The use of X.509v3 certificates for TLS is addressed in FIA\_X509\_EXT.2.1. This requirement adds that this use must include support for client-side certificates for TLS mutual authentication.*

**FCS\_TLSS\_EXT.2.5** The TSF shall not establish a trusted channel if the peer certificate is invalid.

~~Application Note 126~~Application Note 137

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

**FCS\_TLSS\_EXT.2.6** The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

~~Application Note 127~~Application Note 138

*This requirement only applies to those TOEs performing mutually-authenticated TLS (FCS\_TLSS\_EXT.2.4). The peer identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the Domain Name, IP address, username, or email address used by the peer, or may be passed to a directory server for comparison.*

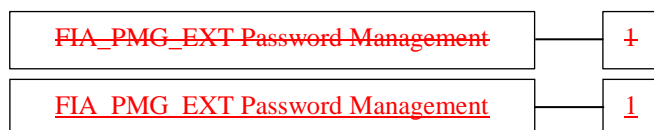
## C.3 Identification and Authentication (FIA)

### C.3.1 Password Management (FIA\_PMG\_EXT)

#### Family Behaviour

The TOE defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

#### Component leveling



**FIA\_PMG\_EXT.1** Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

#### Management: FIA\_PMG\_EXT.1

No management functions.

#### Audit: FIA\_PMG\_EXT.1

No specific audit requirements.

#### C.3.1.1 FIA\_PMG\_EXT.1 Password Management

FIA_PMG_EXT.1	Password Management
---------------	---------------------

Hierarchical to: No other components.

Dependencies: No other components.

**FIA\_PMG\_EXT.1.1** The TSF shall provide the following password management capabilities for administrative passwords:

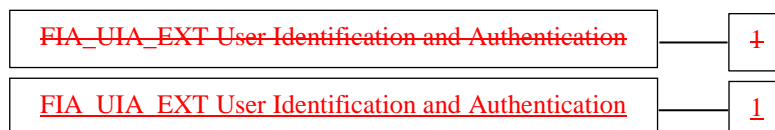
- a) *Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [selection: “!”, “@”, “#”, “\$”, “%”, “^”, “&”, “\*”, “(“, “)”, [assignment: other characters]];*
- b) *Minimum password length shall be settable by the Security Administrator, and support passwords of 15 characters or greater.*

### C.3.2 User Identification and Authentication (FIA\_UIA\_EXT)

#### Family Behaviour

The TSF allows certain specified actions before the non-TOE entity goes through the identification and authentication process.

#### Component leveling



FIA\_UIA\_EXT.1 User Identification and Authentication requires administrators (including remote administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions

#### Management: FIA\_UIA\_EXT.1

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

- a) Ability to configure the list of TOE services available before an entity is identified and authenticated

#### Audit: FIA\_UIA\_EXT.N

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

- a) All use of the identification and authentication mechanism
- b) Provided user identity, origin of the attempt (e.g. IP address)

### C.3.2.1 FIA\_UIA\_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1	User Identification and Authentication
---------------	--

Hierarchical to: No other components.

Dependencies: FTA\_TAB.1 Default TOE Access Banners

**FIA\_UIA\_EXT.1.1** The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA\_TAB.1;
- [selection: *no other actions*, [assignment: *list of services, actions performed by the TSF in response to non-TOE requests.*]]

**FIA\_UIA\_EXT.1.2** The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

#### Application Note 139

This requirement applies to users (administrators and external IT entities) of services available from the TOE directly, and not services available by connecting through the TOE. While it should be the case that few or no services are available to external entities prior to identification and authentication, if there are some available (perhaps ICMP echo) these should be listed in the assignment statement; otherwise “no other actions” should be selected.

Authentication can be password-based through the local console or through a protocol that supports passwords (such as SSH), or be certificate based (such as SSH, TLS).

For communications with external IT entities (e.g., an audit server or NTP server, for instance), such connections must be performed in accordance with FTP ITC.1, whose protocols perform identification and authentication. This means that such communications (e.g., establishing the IPsec connection to the authentication server) would not have to be specified in the assignment, since establishing the connection “counts” as initiating the identification and authentication process.

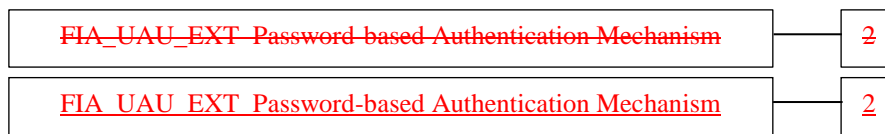
According to the application note for FMT SMR.2, for distributed TOEs at least one TOE component has to support the authentication of Security Administrators according to FIA\_UIA\_EXT.1 and FIA\_UAU\_EXT.2 but not necessarily all TOE components. In case not all TOE components support this way of authentication for Security Administrators the TSS shall describe how Security Administrators are authenticated and identified.

### C.3.3 User authentication (FIA\_UAU) (FIA\_UAU\_EXT)

#### Family Behaviour

Provides for a locally based administrative user authentication mechanism

## Component leveling



FIA\_UAU\_EXT.2 The password-based authentication mechanism provides administrative users a locally based authentication mechanism..

### Management: FIA\_UAU\_EXT.2

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

- a) None

### Audit: FIA\_UAU\_EXT.2

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

- a) Minimal: All use of the authentication mechanism

### C.3.3.1 FIA\_UAU\_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2	Password-based Authentication Mechanism
Hierarchical to:	No other components.
Dependencies:	None

**FIA\_UAU\_EXT.2.1** The TSF shall provide a local password-based authentication mechanism, [selection: *[assignment: other authentication mechanism(s)], none*] to perform administrative user authentication.

#### Application Note 140

The assignment should be used to identify any additional local authentication mechanisms supported. Local authentication mechanisms are defined as those that occur through the local console; remote administrative sessions (and their associated authentication mechanisms) are specified in FTP\_TRP.1/Admin.

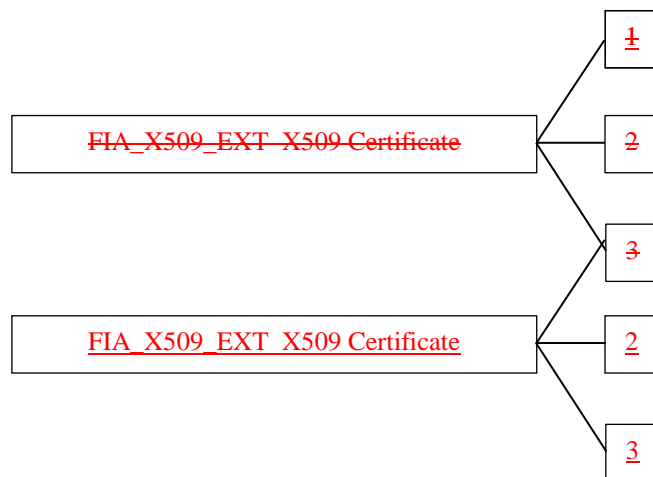
According to the application note for FMT\_SMR.2, for distributed TOEs at least one TOE component has to support the authentication of Security Administrators according to FIA\_UIA\_EXT.1 and FIA\_UAU\_EXT.2 but not necessarily all TOE components. In case not all TOE components support this way of authentication for Security Administrators the TSS shall describe how Security Administrators are authenticated and identified.

### C.3.4 Authentication using X.509 certificates (Extended – FIA\_X509\_EXT)

#### Family Behaviour

This family defines the behavior, management, and use of X.509 certificates for functions to be performed by the TSF. Components in this family require validation of certificates according to a specified set of rules, use of certificates for authentication for protocols and integrity verification, and the generation of certificate requests.

#### Component leveling



FIA\_X509\_EXT.1 X509 Certificate Validation, requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component.

FIA\_X509\_EXT.2 X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates.

FIA\_X509\_EXT.3 X509 Certificate Requests, requires the TSF to be able to generate Certificate Request Messages and validate responses.

#### Management: FIA\_X509\_EXT.1, FIA\_X509\_EXT.2, FIA\_X509\_EXT.3

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

- a) Remove imported X.509v3 certificates
- b) Approve import and removal of X.509v3 certificates
- c) Initiate certificate requests

#### Audit: FIA\_X509\_EXT.1, FIA\_X509\_EXT.2, FIA\_X509\_EXT.3

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

- a) Minimal: No specific audit requirements are specified.



#### C.3.4.1 FIA\_X509\_EXT.1 X.509 Certificate Validation

<b>FIA_X509_EXT.1</b>	<b>X.509 Certificate Validation</b>
-----------------------	-------------------------------------

Hierarchical to: No other components

Dependencies: No other components

**FIA\_X509\_EXT.1.1** The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC ~~2560~~6960, a Certificate Revocation List (CRL) as specified in RFC 5280, Certificate Revocation List (CRL) as specified in RFC 5759, [assignment: list of sections which TSF enforces], no revocation method*].
- The TSF shall validate the extendedKeyUsage field according to the following rules: [assignment: *rules that govern contents of the extendedKeyUsage field that need to be verified*].

~~Application Note 128~~Application Note 141

*FIA\_X509\_EXT.1.1 lists the rules for validating certificates. The ST author selects whether revocation status is verified using OCSP or CRLs. If revocation is not supported the ST author selects no revocation method. The ST author fills in the assignment with rules that may apply to other requirements in the ST. For instance, if a protocol such as TLS that uses certificates is specified in the ST, then certain values for the extendedKeyUsage field (e.g., "Server Authentication Purpose") could be specified.*

**FIA\_X509\_EXT.1.2** The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

~~Application Note 129~~Application Note 142

*This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.*

#### C.3.4.2 FIA\_X509\_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2	X.509 Certificate Authentication
----------------	----------------------------------

Hierarchical to: No other components

Dependencies: ~~No other components~~ [FIA\\_X509\\_EXT.1 X.509 Certificate Validation](#)

**FIA\_X509\_EXT.2.1** The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection: *IPsec, TLS, HTTPS, SSH, [assignment: other protocols], no protocols*], and [selection: *code signing for system software updates, code signing for integrity verification, [assignment: other uses], no additional uses*].

~~Application Note 130~~ [Application Note 143](#)

*If the TOE specifies the implementation of communications protocols that perform peer authentication using certificates, the ST author either selects or assigns the protocols that are specified; otherwise, they select “no protocols”. The TOE may also use certificates for other purposes; the second selection and assignment are used to specify these cases.*

**FIA\_X509\_EXT.2.2** When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: *allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

~~Application Note 131~~ [Application Note 144](#)

*Often a connection must be established to check the revocation status of a certificate - either to download a CRL or to perform a lookup using OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA\_X509\_EXT.1, the behavior indicated in the selection determines the validity. [The TOE must not accept the certificate if it fails any of the other validation rules in FIA\\_X509\\_EXT.1. If the administrator-configured option is selected by the ST Author, the ST Author also selects the corresponding function in FMT\\_SMF.1. If a connection is not required to determine validity, that is that no revocation method is selected then this SFR is trivially satisfied.](#)*

⋮

#### C.3.4.3 FIA\_X509\_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3	X.509 Certificate Requests
----------------	----------------------------

Hierarchical to: No other components

Dependencies: ~~No other components~~ [FCS\\_CKM.1 Cryptographic Key Generation](#)  
[FIA\\_X509\\_EXT.1 X.509 Certificate Validation](#)

**FIA\_X509\_EXT.3.1** The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: *device-specific information, Common Name, Organization, Organizational Unit, Country, [assignment: other information]*].

**FIA\_X509\_EXT.3.2** The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

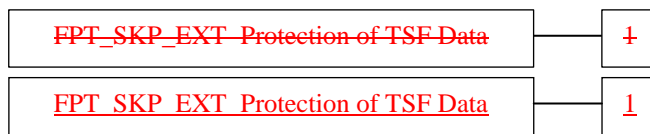
## **C.4 Protection of the TSF (FPT)**

### **C.4.1 Protection of TSF Data (FPT\_SKP\_EXT)**

#### **Family Behaviour**

Components in this family address the requirements for managing and protecting TSF data, such as cryptographic keys. This is a new family modelled after the FPT\_PTD Class.

## Component leveling



FPT\_SKP\_EXT.1 Protection of TSF Data (for reading all symmetric keys), requires preventing symmetric keys from being read by any user or subject. It is the only component of this family.

### Management: FPT\_SKP\_EXT.1

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

- a) There are no management activities foreseen.

### Audit: FPT\_SKP\_EXT.1

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

- a) There are no auditable events foreseen.

#### C.4.1.1 FPT\_SKP\_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)
---------------	--

Hierarchical to: No other components.

Dependencies: No other components.

**FPT\_SKP\_EXT.1.1** The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

#### ~~Application Note 132~~ Application Note 145

*The intent of this requirement is for the device to protect keys, key material, and authentication credentials from unauthorized disclosure. This data should only be accessed for the purposes of their assigned security functionality, and there is no need for them to be displayed/accessed at any other time. This requirement does not prevent the device from providing indication that these exist, are in use, or are still valid. It does, however, restrict the reading of the values outright.*

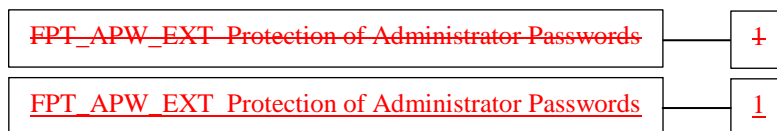
## C.4.2 Protection of Administrator Passwords (FPT\_APW\_EXT)

### C.4.2.1 FPT\_APW\_EXT.1 Protection of Administrator Passwords

#### Family Behaviour

Components in this family ensure that the TSF will protect plaintext credential data such as passwords from unauthorized disclosure.

#### Component leveling



FPT\_APW\_EXT.1 Protection of administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.

#### Management: FPT\_APW\_EXT.1

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

- a) No management functions.

#### Audit: FPT\_APW\_EXT.1

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

- a) No audit necessary.

<b>FPT_APW_EXT.1</b>	<b>Protection of Administrator Passwords</b>
----------------------	--

Hierarchical to: No other components

Dependencies: No other components.

**FPT\_APW\_EXT.1.1** The TSF shall store passwords in non-plaintext form.

**FPT\_APW\_EXT.1.2** The TSF shall prevent the reading of plaintext passwords.

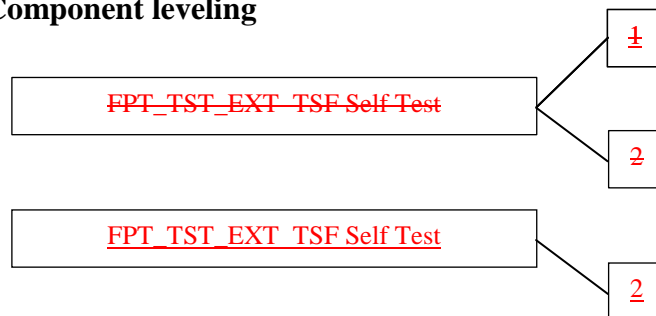
## C.4.3 TSF self test

### C.4.3.1 FPT\_TST\_EXT.1 TSF Testing

#### Family Behaviour

Components in this family address the requirements for self-testing the TSF for selected correct operation.

### Component leveling



FPT\_TST\_EXT.1 TSF Self Test requires a suite of self tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

FPT\_TST\_EXT.2 Self tests based on certificates applies when using certificates as part of self test, and requires that the self test fails if a certificate is invalid.

### Management: FPT\_TST\_EXT.1, FPT\_TST\_EXT.2

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

- a) No management functions.

### Audit: FPT\_TST\_EXT.1, FPT\_TST\_EXT.2

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

- a) Indication that TSF self test was completed

FPT_TST_EXT.1	TSF testing
---------------	-------------

Hierarchical to: No other components.

Dependencies: None

**FPT\_TST\_EXT.1.1** The TSF shall run a suite of the following self-tests [selection: *during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]*] to demonstrate the correct operation of the TSF: [assignment: *list of self-tests run by the TSF*].

### Application Note 133~~Application Note 146~~

*It is expected that self-tests are carried out during initial start-up (on power on). Other options should only be used if the developer can justify why they are not carried out during initial start-up. It is expected that at least self-tests for verification of the integrity of the firmware and software as well as for the correct operation of cryptographic functions*

necessary to fulfil the SFRs will be performed. If not all self-test are performed during start-up multiple iterations of this SFR are used with the appropriate options selected. In future versions of this cPP the suite of self-tests will be required to contain at least mechanisms for measured boot including self-tests of the components which perform the measurement.

For distributed TOEs all TOE components have to perform self-tests. This does not necessarily mean that each TOE component has to carry out the same self-tests: the ST describes the applicability of the selection (i.e. when self-tests are run) and the final assignment (i.e. which self-tests are carried out) to each TOE component.

#### ~~Application Note 134~~Application Note 147

If certificates are used by the self-test mechanism (e.g. for verification of signatures for integrity verification), certificates are validated in accordance with FIA\_X509\_EXT.1 and should be selected in FIA\_X509\_EXT.2.1. Additionally, FPT\_TST\_EXT.2 must be included in the ST.

FPT_TST_EXT.2	Self tests based on certificates
---------------	----------------------------------

Hierarchical to: No other components.

Dependencies: None

**FPT\_TST\_EXT.2.1** The TSF shall fail self-testing if a certificate is used for self tests and the corresponding certificate is deemed invalid.

#### ~~Application Note 135~~Application Note 148

Certificates may optionally be used for self-tests (FPT\_TST\_EXT.1.1). This element must be included in the ST if certificates are used for self-tests. If “code signing for integrity verification” is selected in FIA\_X509\_EXT.2.1, FPT\_TST\_EXT.2 must be included in the ST.

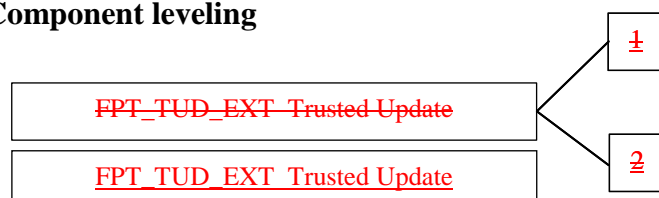
Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with FIA\_X509\_EXT.1.

## C.4.4 Trusted Update (FPT\_TUD\_EXT)

### Family Behaviour

Components in this family address the requirements for updating the TOE firmware and/or software.

#### Component leveling





FPT\_TUD\_EXT.1 Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

FPT\_TUD\_EXT.2 Trusted update based on certificates applies when using certificates as part of trusted update, and requires that the update does not install if a certificate is invalid.

#### **Management: FPT\_TUD\_EXT.1**

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

- a) Ability to update the TOE and to verify the updates
- b) Ability to update the TOE and to verify the updates using the digital signature capability (FCS\_COP.1~~(2)~~/SigGen) and [selection: no other functions, [assignment: other cryptographic functions (or other functions) used to support the update capability]]
- c) Ability to update the TOE, and to verify the updates using [selection: digital signature, published hash, no other mechanism] capability prior to installing those updates

#### **Audit: FPT\_TUD\_EXT.1**

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

- a) Initiation of the update process.
- b) Any failure to verify the integrity of the update

#### **C.4.4.1 FPT\_TUD\_EXT.1 Trusted Update**

<b>FPT_TUD_EXT.1</b>	<b>Trusted update</b>
Hierarchical to:	No other components
Dependencies:	FCS_COP.1 <del>(1)</del> / <u>SigGen</u> Cryptographic <del>operation</del> <u>operation</u> (for <del>cryptographic</del> <u>signature</u> <del>Cryptographic</del> <u>Signature</u> and <u>Verification</u> ), or FCS_COP.1 <del>(3)</del> / <u>Hash</u> Cryptographic operation (for cryptographic hashing)

**FPT\_TUD\_EXT.1.1** The TSF shall provide [assignment: *authorised users*] the ability to query the currently executing version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

#### ~~*Application Note 136*~~*Application Note 149*

*The version currently running (being executed) may not be the version most recently installed. For instance, maybe the update was installed but the system requires a reboot before this update will run. Therefore, it needs to be clear that the query should indicate both the most recently executed version as well as the most recently installed update.*

**FPT\_TUD\_EXT.1.2** The TSF shall provide [assignment: *authorised users*] the ability to manually initiate updates to TOE firmware/software and [selection: *support automatic checking for updates, support automatic updates, no other update mechanism*].

~~Application Note 137~~Application Note 150

*The selection in FPT\_TUD\_EXT.1.2 distinguishes the support of automatic checking for updates and support of automatic updates. The first option refers to a TOE that checks whether a new update is available, communicates this to the administrator (e.g. through a message during an administrator session, through log files) but requires some action by the administrator to actually perform the update. The second option refers to a TOE that checks for updates and automatically installs them upon availability.*

*The TSS explains what actions are involved in the TOE support when using the 'support automatic checking for updates' or 'support automatic updates' selections.*

*When published hash values (see FPT\_TUD\_EXT.1.3) are used to protect the trusted update mechanism, the TOE must not automatically download the update file(s) together with the hash value (either integrated in the update file(s) or separately) and automatically install the update without any active authorization by the Security Administrator, even when the calculated hash value matches the published hash value. When using published hash values to protect the trusted update mechanism, the option 'support of automatic updates' must not be used (automated checking for updates is permitted, though). The TOE may automatically download the update file(s) themselves but not to the hash value. For the published hash approach, it is intended that a Security Administrator is always required to give active authorisation for installation of an update (as described in more detail under FPT\_TUD\_EXT.1.3) below. Due to this, the type of update mechanism is regarded as 'manually initiated update', even if the update file(s) may be downloaded automatically. A fully automated approach (without Security Administrator intervention) can only be used when 'digital signature mechanism' is selected in FPT\_TUD\_EXT.1.3 below.*

**FPT\_TUD\_EXT.1.3** The TSF shall provide means to authenticate firmware/software updates to the TOE using a [selection: *digital signature mechanism, published hash*] prior to installing those updates.

~~Application Note 138~~Application Note 151

*The digital signature mechanism referenced in the selection of FPT\_TUD\_EXT.1.3 is one of the algorithms specified in FCS\_COP.1(2)-/SigGen. The published hash referenced in FPT\_TUD\_EXT.1.3 is generated by one of the functions specified in FCS\_COP.1(3)-/Hash. The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms.*

*When published hash values are used to secure the trusted update mechanism, an active authorization of the update process by the Security Administrator is always required. The secure transmission of an authentic hash value from the developer to the Security Administrator is one of the key factors to protect the trusted update mechanism when using published hashes and the guidance documentation needs to describe how this transfer has to be performed. For the verification of the trusted hash value by the Security Administrator*

*different use cases are possible. The Security Administrator could obtain the published hash value as well as the update file(s) and perform the verification outside the TOE while the hashing of the update file(s) could be done by the TOE or by other means. Authentication as Security Administrator and initiation of the trusted update would in this case be regarded as 'active authorization' of the trusted update. Alternatively, the Administrator could provide the TOE with the published hash value together with the update file(s) and the hashing and hash comparison is performed by the TOE. In case of successful hash verification the TOE can perform the update without any additional step by the Security Administrator. Authentication as Security Administrator and sending the hash value to the TOE is regarded as 'active authorization' of the trusted update (in case of successful hash verification), because the administrator is expected to load the hash value only to the TOE when intending to perform the update. As long as the transfer of the hash value to the TOE is performed by the Security Administrator, loading of the update file(s) can be performed by the Security Administrator or can be automatically downloaded by the TOE from a repository.*

*If the digital signature mechanism is selected, the verification of the signature shall be performed by the TOE itself. For the published hash option, the verification can be done by the TOE itself as well as by the Security Administrator. In the latter case use of TOE functionality for the verification is not mandated, so verification could be done using non-TOE functionality of the device containing the TOE or without using the device containing the TOE.*

*For distributed TOEs all TOE components shall support Trusted Update. The verification of the signature or hash on the update shall either be done by each TOE component itself (signature verification) or for each component (hash verification).*

*Updating a distributed TOE might lead to the situation where different TOE components are running different software versions. Depending on the differences between the different software versions the impact of a mixture of different software versions might be no problem at all or critical to the proper functioning of the TOE. The TSS shall detail the mechanisms that support the continuous proper functioning of the TOE during trusted update of distributed TOEs.*

#### ~~Application Note 139~~Application Note 152

*Future versions of this cPP will mandate the use of a digital signature mechanism for trusted updates.*

#### ~~Application Note 140~~Application Note 153

*If certificates are used by the update verification mechanism, certificates are validated in accordance with FIA\_X509\_EXT.1 and should be selected in FIA\_X509\_EXT.2.1. Additionally, FPT\_TUD\_EXT.2 must be included in the ST.*

#### ~~Application Note 141~~Application Note 154

*“Update” in the context of this SFR refers to the process of replacing a non-volatile, system resident software component with another. The former is referred to as the NV image, and the latter is the update image. While the update image is typically newer than the NV image, this is not a requirement. There are legitimate cases where the system owner may want to rollback a component to an older version (e.g. when the component manufacturer releases a faulty update, or when the system relies on an undocumented feature no longer present in the*

update). Likewise, the owner may want to update with the same version as the NV image to recover from faulty storage.

All discrete software components (e.g. applications, drivers, kernel, firmware) of the TSF, need to be protected, i.e. they should *either* be digitally signed by the corresponding manufacturer and subsequently verified by the mechanism performing the update, *or a hash should be published for them which needs to be verified before the update.* Since it is recognized that components may be signed by different manufacturers, (in case signatures are used to protect updates), it is essential that the update process verify that both the update and NV images were produced by the same manufacturer (e.g. by comparing public keys) or signed by legitimate signing keys (e.g. successful verification of certificates when using X.509 certificates).

#### C.4.4.2 FPT\_TUD\_EXT.2 Trusted Update based on certificates

<b>FPT_TUD_EXT.2</b>	<b>Trusted update based on certificates</b>
----------------------	---

Hierarchical to: No other components

Dependencies: FPT\_TUD\_EXT.1

**FPT\_TUD\_EXT.2.1** The TSF shall not install an update if the code signing certificate is deemed invalid.

**FPT\_TUD\_EXT.2.2** When the certificate is deemed invalid because the certificate has expired, the TSF shall [selection: *allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

#### ~~Application Note 142~~Application Note 155

*Certificates may optionally be used for code signing of system software updates (FPT\_TUD\_EXT.1.3). This element must be included in the ST if certificates are used for validating updates. If “code signing for system software updates” is selected in FIA\_X509\_EXT.2.1, FPT\_TUD\_EXT.2 must be included in the ST.*

*Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with FIA\_X509\_EXT.1. For expired certificates the author of the ST selects whether the certificate shall be accepted, rejected or the choice is left to the administrator to accept or reject the certificate.*

## C.5 TOE Access (FTA)

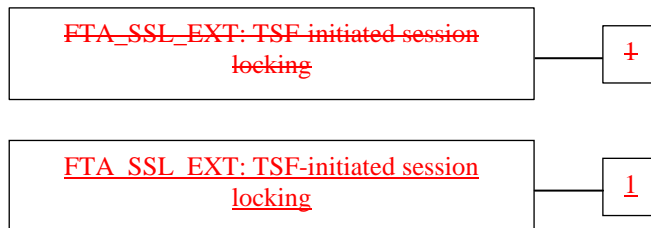
### C.5.1 FTA\_SSL\_EXT.1 TSF-initiated Session Locking

#### Family Behaviour

Components in this family address the requirements for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

The extended FTA\_SSL\_EXT family is based on the FTA\_SSL family.

### Component leveling



FTA\_SSL\_EXT.1 TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.

### Management: FTA\_SSL\_EXT.1

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

- c) Specification of the time of user inactivity after which lock-out occurs for an individual user.

### Audit: FTA\_SSL\_EXT.1

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

- a) Any attempts at unlocking an interactive session.

<b>FTA_SSL_EXT.1</b>	<b>TSF-initiated Session Locking</b>
----------------------	--------------------------------------

Hierarchical to: No other components

Dependencies: FIA\_UAU.1 Timing of authentication

**FTA\_SSL\_EXT.1.1** The TSF shall, for local interactive sessions, [selection:

- *lock the session - disable any activity of the user's data access/display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session;*
- *terminate the session]*

after a Security Administrator-specified time period of inactivity.

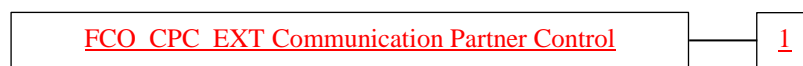
## **C.6 Communication (FCO)**

### **C.6.1 Communication Partner Control (FCO CPC EXT)**

#### **Family Behaviour**

This family is used to define high-level constraints on the ways that partner IT entities communicate. For example, there may be constraints on when communication channels can be used, how they are established, and links to SFRs expressing lower-level security properties of the channels.

#### **Component leveling**



FCO CPC EXT.1 Component Registration Channel Definition, requires the TSF to support a registration channel for joining together components of a distributed TOE, and to ensure that the availability of this channel is under the control of an Administrator. It also requires statement of the type of channel used (allowing specification of further lower-level security requirements by reference to other SFRs).

#### **Management: FCO CPC EXT.1**

No separate management functions are required. Note that elements of the SFR already specify certain constraints on communication in order to ensure that the process of forming a distributed TOE is a controlled activity.

#### **Audit: FCO CPC EXT.1**

The following actions should be auditable if FCO CPC EXT.1 is included in the PP/ST:

- a) Enabling communications between a pair of components as in FCO CPC EXT.1.1 (including identities of the endpoints).
- b) Disabling communications between a pair of components as in FCO CPC EXT.1.3 (including identity of the endpoint that is disabled).

If the required types of channel in FCO CPC EXT.1.2 are specified by using other SFRs then the use of the registration channel may be sufficiently covered by the audit requirements on those SFRs: otherwise a separate audit requirement to audit the use of the channel should be identified for FCO CPC EXT.1.

#### **A.1.1.1.1 FCO CPC EXT.1 Component Registration Channel Definition**

Hierarchical to: No other components.

Dependencies: No other components.

<b>FCO CPC EXT.1</b>	<b>Component Registration Channel Definition</b>
----------------------	--

**FCO CPC EXT.1.1** The TSF shall require a Security Administrator to enable communications between any pair of TOE components before such communication can take place.

**FCO CPC EXT.1.2** The TSF shall implement a registration process in which components establish and use a communications channel that uses [assignment: *list of different types of channel given in the form of a selection*] for at least [assignment: *type of data for which the channel must be used*].

**FCO CPC EXT.1.3** The TSF shall enable a Security Administrator to disable communications between any pair of TOE components.

**Application Note 156**

*This SFR is generally applied to a distributed TOE in order to control the process of creating the distributed TOE from its components by means of a registration process in which a component joins the distributed TOE by registering with an existing component of the distributed TOE. When creating the TSF from the initial pair of components, either of these components may be identified as the TSF for the purposes of satisfying the meaning of “TSF” in this SFR.*

*The intention of this requirement is to ensure that there is a registration process that includes a positive enablement step by an administrator before components joining a distributed TOE can communicate with the other components of the TOE and before the new component can act as part of the TSF. The registration process may itself involve communication with the joining component: many network devices use a bespoke process for this, and the security requirements for the ‘registration communication’ are then defined in FCO CPC EXT.1.2. Use of this ‘registration communication’ channel is not deemed inconsistent with the requirement of FCO CPC EXT.1.1 (i.e. the registration channel can be used before the enablement step, but only in order to complete the registration process).*



## **D. Entropy Documentation And Assessment**

This appendix describes the required supplementary information for each entropy source used by the TOE.

The documentation of the entropy source(s) should be detailed enough that, after reading, the evaluator will thoroughly understand the entropy source and why it can be relied upon to provide sufficient entropy. This documentation should include multiple detailed sections: design description, entropy justification, operating conditions, and health testing. This documentation is not required to be part of the TSS.

### **D.1 Design Description**

Documentation shall include the design of each entropy source as a whole, including the interaction of all entropy source components. Any information that can be shared regarding the design should also be included for any third-party entropy sources that are included in the product.

The documentation will describe the operation of the entropy source to include how entropy is produced, and how unprocessed (raw) data can be obtained from within the entropy source for testing purposes. The documentation should walk through the entropy source design indicating where the entropy comes from, where the entropy output is passed next, any post-processing of the raw outputs (hash, XOR, etc.), if/where it is stored, and finally, how it is output from the entropy source. Any conditions placed on the process (e.g., blocking) should also be described in the entropy source design. Diagrams and examples are encouraged.

This design must also include a description of the content of the security boundary of the entropy source and a description of how the security boundary ensures that an adversary outside the boundary cannot affect the entropy rate.

If implemented, the design description shall include a description of how third-party applications can add entropy to the RBG. A description of any RBG state saving between power-off and power-on shall be included.

### **D.2 Entropy Justification**

There should be a technical argument for where the unpredictability in the source comes from and why there is confidence in the entropy source delivering sufficient entropy for the uses made of the RBG output (by this particular TOE). This argument will include a description of the expected min-entropy rate (i.e. the minimum entropy (in bits) per bit or byte of source data) and explain that sufficient entropy is going into the TOE randomizer seeding process. This discussion will be part of a justification for why the entropy source can be relied upon to produce bits with entropy.

The amount of information necessary to justify the expected min-entropy rate depends on the type of entropy source included in the product.

For developer-provided entropy sources, in order to justify the min-entropy rate, it is expected that a large number of raw source bits will be collected, statistical tests will be

performed, and the min-entropy rate determined from the statistical tests. While no particular statistical tests are required at this time, it is expected that some testing is necessary in order to determine the amount of min-entropy in each output.

For third-party provided entropy sources, in which the TOE vendor has limited access to the design and raw entropy data of the source, the documentation will indicate an estimate of the amount of min-entropy obtained from this third-party source. It is acceptable for the vendor to “assume” an amount of min-entropy, however, this assumption must be clearly stated in the documentation provided. In particular, the min-entropy estimate must be specified and the assumption included in the ST.

Regardless of the type of entropy source, the justification will also include how the DRBG is initialized with the entropy stated in the ST, for example by verifying that the min-entropy rate is multiplied by the amount of source data used to seed the DRBG or that the rate of entropy expected based on the amount of source data is explicitly stated and compared to the statistical rate. If the amount of source data used to seed the DRBG is not clear or the calculated rate is not explicitly related to the seed, the documentation will not be considered complete.

The entropy justification shall not include any data added from any third-party application or from any state saving between restarts.

### **D.3 Operating Conditions**

The entropy rate may be affected by conditions outside the control of the entropy source itself. For example, voltage, frequency, temperature, and elapsed time after power-on are just a few of the factors that may affect the operation of the entropy source. As such, documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. Similarly, documentation shall describe the conditions under which the entropy source is no longer guaranteed to provide sufficient entropy. Methods used to detect failure or degradation of the source shall be included.

### **D.4 Health Testing**

More specifically, all entropy source health tests and their rationale will be documented. This will include a description of the health tests, the rate and conditions under which each health test is performed (e.g., at startup, continuously, or on-demand), the expected results for each health test, TOE behavior upon entropy source failure, and rationale indicating why each test is believed to be appropriate for detecting one or more failures in the entropy source.

## **E. Rationales**

### **E.1 SFR Dependencies Analysis**

The dependencies between SFRs implemented by the TOE are addressed as follows.

<b><u>SFR</u></b>	<b><u>Dependencies</u></b>	<b><u>Rationale Statement</u></b>
<u>FAU_GEN.1</u>	<u>FPT_STM.1</u>	<u>FPT_STM.1 included</u>
<u>FAU_GEN.2</u>	<u>FAU_GEN.1</u> <u>FIA_UID.1</u>	<u>FAU_GEN.1 included</u>  Satisfied by <u>FIA_UA_EXT.1</u> , which specifies the relevant <u>administrator identification timing</u>
<u>FAU_STG_EXT.1</u>	<u>FAU_GEN.1</u> <u>FTP_ITC.1</u>	<u>FAU_GEN.1 included</u>  <u>FTP_ITC.1 included</u>
<u>FCS_CKM.1</u>	<u>FCS_CKM.2 or FCS_COP.1</u>  <u>FCS_CKM.4</u>	<u>FCS_CKM.2 included</u>  <u>FCS_CKM.4 included</u>
<u>FCS_CKM.2</u>	<u>FTP_ITC.1 or FTP_ITC.2 or FCS_CKM.1</u>  <u>FCS_CKM.4</u>	<u>FCS_CKM.1 included (also FTP_ITC.1 as a secure channel that could be used for import)</u>  <u>FCS_CKM.4 included</u>
<u>FCS_CKM.4</u>	<u>FTP_ITC.1 or FTP_ITC.2 or FCS_CKM.1</u>	<u>FCS_CKM.1 included (also FTP_ITC.1 as a secure channel that could be used for</u>

		<u>import)</u>
<u>FCS_COP.1/DataEncryption</u>	<u>FTP_ITC.1 or</u> <u>FTP_ITC.2 or</u> <u>FCS_CKM.1</u>  <u>FCS_CKM.4</u>	<u>FCS_CKM.1</u> <u>included (also</u> <u>FTP_ITC.1 as a</u> <u>secure channel that</u> <u>could be used for</u> <u>import)</u>  <u>FCS_CKM.4</u> <u>included</u>
<u>FCS_COP.1/SigGen</u>	<u>FTP_ITC.1 or</u> <u>FTP_ITC.2 or</u> <u>FCS_CKM.1</u>  <u>FCS_CKM.4</u>	<u>FCS_CKM.1</u> <u>included (also</u> <u>FTP_ITC.1 as a</u> <u>secure channel that</u> <u>could be used for</u> <u>import)</u>  <u>FCS_CKM.4</u> <u>included</u>
<u>FCS_COP.1/Hash</u>	<u>FTP_ITC.1 or</u> <u>FTP_ITC.2 or</u> <u>FCS_CKM.1</u>  <u>FCS_CKM.4</u>	<u>This SFR specifies</u> <u>keyless hashing</u> <u>operations, so</u> <u>initialisation and</u> <u>destruction of keys</u> <u>are not relevant</u>
<u>FCS_COP.1/KeyedHash</u>	<u>FTP_ITC.1 or</u> <u>FTP_ITC.2 or</u> <u>FCS_CKM.1</u>  <u>FCS_CKM.4</u>	<u>FCS_CKM.1</u> <u>included (also</u> <u>FTP_ITC.1 as a</u> <u>secure channel that</u> <u>could be used for</u> <u>import)</u>  <u>FCS_CKM.4</u> <u>included</u>
<u>FCS_RBG_EXT.1</u>	<u>None</u>	
<u>FIA_AFL.1</u>	<u>FIA_UAU.1</u>	<u>Satisfied by</u> <u>FIA_UIA_EXT.1,</u> <u>which specifies the</u> <u>relevant</u> <u>administrator</u> <u>authentication</u>

<u>FIA_PMG_EXT.1</u>	<u>None</u>	
<u>FIA_UIA_EXT.1</u>	<u>FTA_TAB.1</u>	<u>FTA_TAB.1</u> <u>included</u>
<u>FIA_UAU_EXT.2</u>	<u>None</u>	
<u>FIA_UAU.7</u>	<u>FIA_UAU.1</u>	<u>Satisfied by</u> <u>FIA_UIA_EXT.1,</u> <u>which specifies the</u> <u>relevant</u> <u>administrator</u> <u>authentication</u>
<u>FMT_MOF.1/ManualUpdate</u>	<u>FMT_SMR.1</u>  <u>FMT_SMF.1</u>	<u>FMT_SMR.2</u> <u>included</u>  <u>FMT_SMF.1</u> <u>included</u>
<u>FMT_MTD.1/CoreData</u>	<u>FMT_SMR.1</u>  <u>FMT_SMF.1</u>	<u>FMT_SMR.2</u> <u>included</u>  <u>FMT_SMF.1</u> <u>included</u>
<u>FMT_SMF.1</u>	<u>None</u>	
<u>FMT_SMR.2</u>	<u>FIA_UID.1</u>	<u>Satisfied by</u> <u>FIA_UIA_EXT.1,</u> <u>which specifies the</u> <u>relevant</u> <u>administrator</u> <u>identification</u>
<u>FPT_SKP_EXT.1</u>	<u>None</u>	
<u>FPT_APW_EXT.1</u>	<u>None</u>	
<u>FPT_TST_EXT.1</u>	<u>None</u>	
<u>FPT_TUD_EXT.1</u>	<u>FCS_COP.1/SigGen</u> <u>or FCS_COP.1/Hash</u>	<u>FCS_COP.1/SigGen</u> <u>and</u> <u>FCS_COP.1/Hash</u> <u>included</u>
<u>FPT_STM.1</u>	<u>None</u>	
<u>FTA_SSL_EXT.1</u>	<u>FIA_UAU.1</u>	<u>Satisfied by</u> <u>FIA_UIA_EXT.1,</u>

		<u>which specifies the relevant administrator authentication</u>
<u>FTA_SSL.3</u>	<u>None</u>	
<u>FTA_SSL.4</u>	<u>None</u>	
<u>FTA_TAB.1</u>	<u>None</u>	
<u>FTP_ITC.1</u>	<u>None</u>	
<u>FTP_TRP.1/Admin</u>	<u>None</u>	

*Table 6: SFR Dependencies Rationale for Mandatory SFRs*

<u>SFR</u>	<u>Dependencies</u>	<u>Rationale Statement</u>
<u>FAU_STG.1</u>	<u>FAU_STG.3</u>	<u>FAU_STG.3/LocSpace</u> <u>included as optional SFRs</u>
<u>FAU_STG_EXT.2/LocSpace</u>	<u>FAU_GEN.1</u> <u>FAU_STG_EXT.1</u>	<u>FAU_GEN.1</u> & <u>FAU_STG_EXT.1</u> <u>included</u>
<u>FAU_STG.3/LocSpace</u>	<u>FAU_STG.1</u>	<u>FAU_STG.1</u> included as <u>optional SFR</u>
<u>FIA_X509_EXT.1/ITT</u>	<u>None</u>	
<u>FMT_MOF.1/Service</u>	<u>FMT_SMR.1</u> <u>FMT_SMF.1</u>	<u>FMT_SMR.2</u> included <u>FMT_SMF.1</u> included
<u>FMT_MOF.1/Functions</u>	<u>FMT_SMR.1</u> <u>FMT_SMF.1</u>	<u>FMT_SMR.2</u> included <u>FMT_SMF.1</u> included
<u>FMT_MTD.1/CryptoKeys</u>	<u>FMT_SMR.1</u> <u>FMT_SMF.1</u>	<u>FMT_SMR.2</u> included <u>FMT_SMF.1</u> included
<u>FPT_FLS.1/LocSpace</u>	<u>None</u>	
<u>FPT_ITT.1</u>	<u>None</u>	
<u>FTP_TRP.1/Join</u>	<u>None</u>	
<u>FCO_CPC_EXT.1</u>	<u>None</u>	

*Table 7: SFR Dependencies Rationale for Optional SFRs*

<u>SFR</u>	<u>Dependencies</u>	<u>Rationale Statement</u>
<u>FIA_X509_EXT.1/Rev</u>	<u>None</u>	
<u>FIA_X509_EXT.2</u>	<u>None</u>	
<u>FIA_X509_EXT.3</u>	<u>FCS_CKM.1</u>	
<u>FCS_HTTPS_EXT.1</u>	<u>FCS_TLSC_EXT.1</u> or <u>FCS_TLSS_EXT.1</u>	<u>FCS_TLSC_EXT.1</u> and <u>FCS_TLSS_EXT.1</u> <u>included as selection-</u>



		<u>based SFRs</u>
<u>FCS_IPSEC_EXT.1</u>	<u>FCS_CKM.1</u> <u>FCS_CKM.2</u> <u>FCS_COP.1/DataEncryption</u> <u>FCS_COP.1/SigGen</u> <u>FCS_COP.1/Hash</u> <u>FCS_COP.1/KeyedHash</u> <u>FCS_RBG_EXT.1</u>	<u>FCS_CKM.1 included</u> <u>FCS_CKM.2 included</u> <u>FCS_COP.1/DataEncryption</u> <u>on, FCS_COP.1/SigGen,</u> <u>FCS_COP.1/Hash,</u> <u>FCS_COP.1/KeyedHash</u> <u>included</u> <u>FCS_RBG_EXT.1</u> <u>included</u>
<u>FCS_SSHC_EXT.1</u>	<u>FCS_CKM.1</u> <u>FCS_CKM.2</u> <u>FCS_COP.1/DataEncryption</u> <u>on</u> <u>FCS_COP.1/SigGen</u> <u>FCS_COP.1/Hash</u> <u>FCS_COP.1/KeyedHash</u> <u>FCS_RBG_EXT.1</u>	<u>FCS_CKM.1 included</u> <u>FCS_CKM.2 included</u> <u>FCS_COP.1/DataEncryption</u> <u>on, FCS_COP.1/SigGen,</u> <u>FCS_COP.1/Hash,</u> <u>FCS_COP.1/KeyedHash</u> <u>included</u> <u>FCS_RBG_EXT.1</u> <u>included</u>
<u>FCS_SSHS_EXT.1</u>	<u>FCS_CKM.1</u> <u>FCS_CKM.2</u> <u>FCS_COP.1/DataEncryption</u> <u>on</u> <u>FCS_COP.1/SigGen</u> <u>FCS_COP.1/Hash</u> <u>FCS_COP.1/KeyedHash</u> <u>FCS_RBG_EXT.1</u>	<u>FCS_CKM.1 included</u> <u>FCS_CKM.2 included</u> <u>FCS_COP.1/DataEncryption</u> <u>on, FCS_COP.1/SigGen,</u> <u>FCS_COP.1/Hash,</u> <u>FCS_COP.1/KeyedHash</u> <u>included</u> <u>FCS_RBG_EXT.1</u> <u>included</u>
<u>FCS_TLSC_EXT.1</u>	<u>FCS_CKM.1</u> <u>FCS_CKM.2</u>	<u>FCS_CKM.1 included</u> <u>FCS_CKM.2 included</u>

	<u>FCS_COP.1/DataEncrypti</u> <u>on</u> <u>FCS_COP.1/SigGen</u> <u>FCS_COP.1/Hash</u> <u>FCS_COP.1/KeyedHash</u> <u>FCS_RBG_EXT.1</u>	<u>FCS_COP.1/DataEncrypti</u> <u>on, FCS_COP.1/SigGen,</u> <u>FCS_COP.1/Hash,</u> <u>FCS_COP.1/KeyedHash</u> <u>included</u> <u>FCS_RBG_EXT.1</u> <u>included</u>
<u>FCS_TLSC_EXT.2</u>	<u>FCS_CKM.1</u> <u>FCS_CKM.2</u> <u>FCS_COP.1/DataEncrypti</u> <u>on</u> <u>FCS_COP.1/SigGen</u> <u>FCS_COP.1/Hash</u> <u>FCS_COP.1/KeyedHash</u> <u>FCS_RBG_EXT.1</u>	<u>FCS_CKM.1 included</u> <u>FCS_CKM.2 included</u> <u>FCS_COP.1/DataEncrypti</u> <u>on, FCS_COP.1/SigGen,</u> <u>FCS_COP.1/Hash,</u> <u>FCS_COP.1/KeyedHash</u> <u>included</u> <u>FCS_RBG_EXT.1</u> <u>included</u>
<u>FCS_TLSS_EXT.1</u>	<u>FCS_CKM.1</u> <u>FCS_CKM.2</u> <u>FCS_COP.1/DataEncrypti</u> <u>on</u> <u>FCS_COP.1/SigGen</u> <u>FCS_COP.1/Hash</u> <u>FCS_COP.1/KeyedHash</u> <u>FCS_RBG_EXT.1</u>	<u>FCS_CKM.1 included</u> <u>FCS_CKM.2 included</u> <u>FCS_COP.1/DataEncrypti</u> <u>on, FCS_COP.1/SigGen,</u> <u>FCS_COP.1/Hash,</u> <u>FCS_COP.1/KeyedHash</u> <u>included</u> <u>FCS_RBG_EXT.1</u> <u>included</u>
<u>FCS_TLSS_EXT.2</u>	<u>FCS_CKM.1</u> <u>FCS_CKM.2</u> <u>FCS_COP.1/DataEncrypti</u> <u>on</u> <u>FCS_COP.1/SigGen</u>	<u>FCS_CKM.1 included</u> <u>FCS_CKM.2 included</u> <u>FCS_COP.1/DataEncrypti</u> <u>on, FCS_COP.1/SigGen,</u> <u>FCS_COP.1/Hash,</u> <u>FCS_COP.1/KeyedHash</u> <u>included</u>

	<u>FCS_COP.1/Hash</u> <u>FCS_COP.1/KeyedHash</u> <u>FCS_RBG_EXT.1</u>	<u>FCS_RBG_EXT.1</u> <u>included</u>
<u>FPT_TST_EXT.2</u>	<u>None</u>	
<u>FPT_TUD_EXT.2</u>	<u>FPT_TUD_EXT.1</u>	<u>FPT_TUD_EXT.1</u> <u>included</u>
<u>FMT_MOF.1/AutoUpdate</u>	<u>FMT_SMR.1</u> <u>FMT_SMF.1</u>	<u>FMT_SMR.2 included</u> <u>FMT_SMF.1 included</u>

*Table 8: SFR Dependencies Rationale for Selection-Based SFRs*

## Glossary

Term	Meaning
<b>Administrator</b>	See Security Administrator.
<b>Assurance</b>	Grounds for confidence that a TOE meets the SFRs [CC1].
<b>Key Chaining</b>	The method of using multiple layers of encryption keys to protect data. A top layer key encrypts a lower layer key which encrypts the data; this method can have any number of layers.
<b>Security Administrator</b>	The terms “Administrator” and “Security Administrator” are used interchangeably in this document at present.
<b>Target of Evaluation</b>	A set of software, firmware and/or hardware possibly accompanied by guidance. [CC1]
<b>TOE Security Functionality (TSF)</b>	A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the SFRs. [CC1]
<b>TSF Data</b>	Data for the operation of the TSF upon which the enforcement of the requirements relies.

See [CC1] for other Common Criteria abbreviations and terminology.

## Acronyms

Acronym	Meaning
<b>AEAD</b>	Authenticated Encryption with Associated Data
<b>AES</b>	Advanced Encryption Standard
<b>CA</b>	Certificate Authority
<b>CBC</b>	Cipher Block Chaining
<b>CRL</b>	Certificate Revocation List
<b>DH</b>	Diffie-Hellman
<b>DSA</b>	Digital Signature Algorithm
<b>ECDH</b>	Elliptic Curve Diffie Hellman
<b>ECDSA</b>	Elliptic Curve Digital Signature Algorithm
<b>EEPROM</b>	Electrically Erasable Programmable Read-Only Memory
<b>FIPS</b>	Federal Information Processing Standards
<b>GCM</b>	Galois Counter Mode
<b>HMAC</b>	Keyed-Hash Message Authentication Code
<b>HTTPS</b>	HyperText Transfer Protocol Secure
<b>IP</b>	Internet Protocol
<b>IPsec</b>	Internet Protocol Security
<b>NIST</b>	National Institute of Standards and Technology
<b>OCSP</b>	Online Certificate Status Protocol
<b>PP</b>	Protection Profile
<b>RBG</b>	Random Bit Generator
<b>RSA</b>	Rivest Shamir Adleman Algorithm
<b>SD</b>	Supporting Document
<b>SHA</b>	Secure Hash Algorithm
<b>SSH</b>	Secure Shell
<b>ST</b>	Security Target
<b>TLS</b>	Transport Layer Security
<b>TOE</b>	Target of Evaluation
<b>TSF</b>	TOE Security Functionality
<b>TSS</b>	TOE Summary Specification
<b>VPN</b>	Virtual Private Network