Commercial Solutions for Classified (CSfC) Selections for Transport Layer Security (TLS) Protected Servers

Overview

Transport Layer Security (TLS) Protected Servers products (as defined in the Commercial Solutions for Classified (CSfC) Capability Packages (CPs)) used in CSfC solutions must be validated by National Information Assurance Partnership (NIAP)/Common Criteria Evaluation and Validation Scheme (CCEVS) or Common Criteria Recognition Arrangement (CCRA) partnering schemes as complying with the current requirements of NIAP's:

- collaborative Protection Profile for Network Devices Version 4.0; and
- Functional Package for Transport Layer Security (TLS) Version 2.1; and
- Functional Package for X.509 Version 1.0; and
- If the TOE uses SSH: Functional Package for Secure Shell (SSH) Version 2.0

This validated compliance must include the selectable requirements contained in this document.

TLS Protected Servers can be used/implemented in many different ways, threats and technology continuously progress, and TLS Protected Servers continues to evolve, which may cause the below selections to change or become obsolete. Vendors are encouraged to review the CSfC CPs to ensure the product functions appropriately in CSfC solutions. The objective of the below selections is to provide information to enable the use of the Commercial National Security Algorithm (CNSA) Suite and facilitate the use of TLS Protected Servers in CSfC solutions.

Please provide questions, comments on usability, applicability, and/or shortcomings to the CSfC Program (csfc@nsa.gov).

Notes

Note 1: The following selections apply to CSfC TLS Protected Servers functionality. If needed, functionality and/or configurations outside the scope of a CSfC TLS Protected Servers that conflict with the CSfC selections could be NIAP validated without using a separate iteration of the Security Functional Requirement (SFR) (this is a change to previous guidance in Note 1). The Security Target (ST) author should document a specific CSfC TLS Protected Servers configuration in the product's Administrative Guide with a note that the configuration should be considered the NIAP-certified evaluated configuration for CSfC TLS Protected Servers Use Cases. The CSfC TLS Protected Servers configuration should be used to validate compliance with CSfC selections.

Note 2: The below SFRs/Selections contain some mandatory SFRs without Selections or modifications. The exclusion of other mandatory SFRs in the below Selections does not indicate that mandatory PP SFRs are not required (i.e., Compliance with the requirements as prescribed by the PP, Functional Packages, and outlined in the Overview Section above are required). Some mandatory SFRs are included in the below Selections to highlight some SFRs relevant to CSfC TLS Protected Servers.

Document Conventions

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

- Assignment completed within a selection in the PP: the completed assignment text is indicated with *italicized and underlined text*
- Assignment partially completed in the PP: indicated with *italicized text*
- Refinement text is indicated with ~~strikethroughs~~
- CSfC specific selections, refinements (e.g., underline, strikethrough) are highlighted in light blue text (i.e., CSfC mandatory completed assignments/selections unless otherwise indicated by the `light blue Courier New Text` "at least one of the following underlined selections").
- Additional clarifying text or CSfC specific language is indicated with `light blue Courier New Text`
- Links to sources, additional information, and email addresses are indicated with blue underlined text.

Network Devices Collaborative Protection Profile Version 4.0 Selections

FAU_STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1.6 The TSF shall provide the following mechanisms for administrative access to locally stored audit records [selection: ~~none~~, manual export, ability to view locally].

FCS_CKM.1/AKG Cryptographic Key Generation – Asymmetric Key

FCS_CKM.1.1/AKG The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [selection: *cryptographic key generation algorithm*] and specified cryptographic **algorithm parameters** ~~key sizes~~ [selection: *cryptographic algorithm parameters*] that meet the following: [selection: *list of standards*].

Identifier	Cryptographic Key Generation Algorithm	Cryptographic Algorithm Parameters	List of Standards
RSA	RSA	Modulus of size [selection: ~~*2048,*~~ *3072*, *4096*, *6144*, *8192*] bits	NIST FIPS PUB 186-5 (Section A.1.1)
ECC-ERB	ECC-ERB - Extra Random Bits	Elliptic Curve [selection: ~~*P-256,*~~ *P-384*, ~~*P-521*~~]	NIST FIPS PUB 186-5 (Section A.2.1), NIST SP 800-186 (Section 3) [NIST Curves]
ECC-RS	ECC-RS - Rejection Sampling	Elliptic Curve [selection: ~~*P-256,*~~ *P-384*, ~~*P-521*~~]	NIST FIPS PUB 186-5 (Section A.2.2), NIST SP

			800-186 (Section 3) [NIST Curves]
FFC-ERB	FFC-ERB - Extra Random Bits	Static domain parameters approved for [selection: • *IKE Groups [selection: ~~MODP-2048,~~ MODP-3072, MODP-4096, MODP-6144, MODP-8192*], • *TLS Groups [selection: ~~ffdhe-2048,~~ ffdhe-3072, ffdhe-4096, ffdhe-6144, ffdhe-8192*]]	NIST SP 800-56A Revision 3 (Section 5.6.1.1.3), [selection: *RFC 3526* [IKE groups], *RFC 7919* [TLS groups]]
FFC-RS	FFC-RS - Extra Random Bits	Static domain parameters approved for [selection: • *IKE Groups [selection: ~~MODP-2048,~~ MODP-3072, MODP-4096, MODP-6144, MODP-8192*], • *TLS Groups [selection: ~~ffdhe-2048,~~ ffdhe-3072, ffdhe-4096, ffdhe-6144, ffdhe-8192*]]	NIST SP 800-56A Revision 3 (Section 5.6.1.1.3), [selection: *RFC 3526* [IKE groups], *RFC 7919* [TLS groups]]
LMS	LMS	private key size [selection: • *192 bits with [selection: SHA-256/192, SHAKE256/192*], • *256 bits with [selection: SHA-256, SHAKE256*]] Winternitz parameter = [selection: *1, 2, 4, 8*], Tree height = [selection: *5, 10, 15, 20, 25*]	RFC 8554 [LMS], NIST SP 800-208 [parameters]
XMSS	XMSS	private key size [selection: • *192 bits with [selection: SHA-256/192, SHAKE256/192*] • *256 bits with [selection: SHA-256, SHAKE256*]] Tree height = [selection: *10, 16, 20*]	RFC 8391 [XMSS], NIST SP 800-208 [parameters]

ML-KEM	ML-KEM	Parameter set = ML-KEM-1024	NIST FIPS PUB 203
ML-DSA	ML-DSA	Parameter set = ML-DSA-87	NIST FIPS PUB 204

FCS_CKM.2 Cryptographic Key Distribution

FCS_CKM.2.1 The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [selection: *key encapsulation*, *key wrapping*, *encrypted channels*] that meets the following: none.

FCS_CKM_EXT.7 Cryptographic Key Agreement

FCS_CKM_EXT.7.1 The TSF shall derive shared cryptographic keys with input from multiple parties in accordance with specified cryptographic key agreement algorithms [selection: cryptographic algorithm] and specified cryptographic parameters [selection: cryptographic parameters] that meet the following: [selection: list of standards]

The following table provides the allowed choices for completion of the selection operations of FCS_CKM_EXT.7.1.

Identifier	**Cryptographic Key Generation Algorithm**	**Cryptographic Algorithm Parameters**	**List of Standards**
DH	Finite Field Cryptography Diffie-Hellman	Static domain parameters approved for [selection: • I*KE Groups [selection: ~~MODP-2048,~~ MODP-3072, MODP-4096, MODP-6144, MODP-8192]*, • *TLS Groups [selection: ~~ffdhe2048~~, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192]]*	NIST SP 800-56A Revision 3 (Section 5.7.1.1), [selection: RFC 3526 [IKE groups], RFC 7919 [TLS groups]]
ECDH	Elliptic Curve Diffie-Hellman	Elliptic Curve [selection: ~~P-256~~, P-384, ~~P-521~~]	NIST SP 800-56A Revision 3 (Section 5.7.1.2) [ECDH], NIST SP 800-186 (Section 3.2.1) [NIST Curves]

FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/ Decryption)

FCS_COP.1.1/DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES operating in [selection:

- *CBC mode as defined in FCS_COP.1/SKC,*

- *CTR mode as defined in FCS_COP.1/SKC,*
- *XTS mode as defined in FCS_COP.1/SKC,*
- *CCM mode as defined in FCS_COP.1/AEAD,*
- *GCM mode as defined in FCS_COP.1/AEAD*

].

FCS_COP.1/SigGen Cryptographic Operation - Signature Generation

FCS_COP.1.1/SigGen The TSF shall perform digital signature generation in accordance with a specified cryptographic algorithm [selection: *cryptographic algorithm*] and cryptographic key sizes [selection: *cryptographic key sizes*] that meet the following: [selection: *list of standards*].

The following table provides the allowed choices for completion of the selection operations of FCS_COP.1.1/SigGen.

Cryptographic Algorithm	**Cryptographic Algorithm Parameters**	**List of Standards**
RSASSA-PKCS1- v1_5	Modulus of size [selection: ~~*2048*~~, *3072*, *4096*, *6144*, *8192*] bits and hash [selection: ~~*SHA-256,*~~ *SHA-384*, *SHA-512*]	RFC 8017 (Section 8.2) [PKCS #1 v2.2], FIPS PUB 186-5 (Section 5.4) [RSASSA-PKCS1-v1_5]
RSASSA-PSS	Modulus of size [selection: ~~*2048*~~, *3072*, *4096*, *6144*, *8192*] bits and hash [selection: ~~*SHA-256,*~~ *SHA-384*, *SHA-512*], Salt Length (sLen) such that [assignment: $0 \leq sLen \leq hLen$ (Hash Output Length)] and Mask Generation Function = MGF1]	RFC 8017 (Section 8.1) [PKCS#1 v2.2], FIPS PUB 186-5 (Section 5.4) [RSASSA-PSS]
ECDSA	Elliptic Curve [selection: ~~*P-256*~~, *P384*, ~~*P-521*~~], per-message secret number generation [selection: *extra random bits*, *rejection sampling*, *deterministic*] and hash function using [selection: ~~*SHA256,*~~ *SHA-384*, *SHA-512*]	[selection: ISO/IEC 14888-3:2018 (Subclause 6.6), FIPS PUB 186-5 (Sections 6.3.1, 6.4.1]][ECDSA], NIST SP-800 186 (Section 4) [NIST Curves]
Module-Lattice Based Digital Signature Algorithm	ML-DSA-87	NIST FIPS PUB 204 (Section 5.2)

FCS_COP.1/SigVer Cryptographic Operation - Signature Verification

FCS_COP.1.1/SigVer The TSF shall perform digital signature verification in accordance with a specified cryptographic algorithm [selection: *cryptographic algorithm*] and cryptographic key sizes [selection: *cryptographic key sizes*] that meet the following: [selection: *list of standards*].

The following table provides the allowed choices for completion of the selection operations of FCS_COP.1.1/SigVer.

Cryptographic Algorithm	**Cryptographic Algorithm Parameters**	**List of Standards**
RSASSA-PKCS1- v1_5	Modulus of size [selection: ~~*2048,*~~ *3072*, *4096*, *6144*, *8192*] bits and hash [selection: ~~*SHA-256,*~~ *SHA-384*, *SHA-512*]	RFC 8017 (Section 8.2) [PKCS #1 v2.2], FIPS PUB 186-5 (Section 5.4) [RSASSA-PKCS1-v1_5]
RSASSA-PSS	Modulus of size [selection: ~~*2048,*~~ *3072*, *4096*, *6144*, *8192*] bits and hash [selection: ~~*SHA-256,*~~ *SHA-384*, *SHA-512*]	RFC 8017 (Section 8.1) [PKCS#1 v2.2], FIPS PUB 186-5 (Section 5.4) [RSASSA-PSS]
ECDSA	Elliptic Curve [selection: ~~*P-256,*~~ *P-384*, ~~*P-521*~~] using hash [selection: ~~*SHA-256,*~~ *SHA-384*, *SHA-512*]	[selection: ISO/IEC 14888-3:2018 (Subclause 6.6), FIPS PUB 186-5 (Section 6.4.2)][ECDSA] NIST SP-800 186 (Section 4) [NIST Curves]
LMS	private key size [selection: • *192 bits with [selection: SHA-256/192, SHAKE256/192]* • *256 bits with [selection: SHA-256, SHAKE256]*] Winternitz parameter = [selection: 1, 2, 4, 8]	RFC 8554 [LMS], NIST SP 800-208 [parameters]
XMSS	Tree height = [selection: *5*, *10*, *15*, *20*, *25*] XMSS private key size [selection: • *192 bits with [selection:* *SHA-256/192, SHAKE256/192]* • *256 bits with [selection: SHA-256, SHAKE256]*] Tree height = [selection: *10*, *16*, *20*]	RFC 8391 [XMSS], NIST SP 800-208 [parameters]

ML-DSA	ML-DSA-87	NIST FIPS PUB 204 (Section 5.3)

FCS_COP.1/Hash Cryptographic Operation – Hashing

FCS_COP.1.1/Hash The TSF shall perform cryptographic hashing in accordance with a specified cryptographic algorithm [selection: *SHA-256, SHA-384, SHA-512, SHA3-256, SHA3-384, SHA3-512*] that meets the following: [selection: ISO/IEC 10118-3:2018 [SHA, SHA3], FIPS PUB 180-4 [SHA], FIPS PUB 202 [SHA3]].

Application Note: In accordance with CNSA 1.0 and 2.0:

- SHA-1 hash is no longer permitted to be used as a hash function,
- SHA3 hashes may be used only for internal hardware functionality such as boot integrity checks, and
- SHA-256 is permitted only as part of LMS or XMSS.

The hash selection should be consistent with the overall strength of the algorithm used for signature generation.

FCS_COP.1/KeyedHash Cryptographic Operation - Keyed Hash

FCS_COP.1.1/KeyedHash The TSF shall perform keyed hash message authentication in accordance with a specified cryptographic algorithm [selection: *keyed hash algorithm, implicit*] and cryptographic key sizes [selection: *cryptographic key sizes*] that meet the following: [selection: *list of standards*].

The following table provides the allowed choices for completion of the selection operations of FCS_COP.1.1/KeyedHash.

Keyed Hash Algorithm	**Cryptographic Key Sizes**	**List of Standards**
~~HMAC-SHA256~~	~~256 bits~~	~~[selection: *ISO/IEC 9797-2:2021 (Section 7 "MAC Algorithm 2"), FIPS PUB 198-1*]~~
HMAC-SHA384	[selection: *384 (ISO, FIPS), 256 (FIPS)*] bits	[selection: *ISO/IEC 9797-2:2021 (Section 7 "MAC Algorithm 2"), FIPS PUB 198-1*]
HMAC-SHA512	[selection: *512 (ISO, FIPS), 384 (FIPS), 256 (FIPS)*] bits	[selection: *ISO/IEC 9797-2:2021 (Section 7 "MAC Algorithm 2"), FIPS PUB 198-1*]

Application Note: The HMAC minimum key sizes in the table are specified in ISO/IEC 9797-2:2021, which requires that the minimum key size be equal to the digest size. The FIPS standard specifies no

minimum or maximum key sizes, so if FIPS PUB 198-1 is selected, larger or smaller key sizes may be used. This is indicated by the parenthesized annotations in the Cryptographic Key Sizes column.

FCS_COP.1/AEAD Cryptographic Operation – Authenticated Encryption with Associated Data

This is a selection-based SFR, to be included in the ST if CCM mode or GCM mode are selected in FCS_COP.1/DataEncryption.

FCS_COP.1.1/AEAD The TSF shall perform authenticated encryption with associated data in accordance with a specified cryptographic algorithm [selection: *cryptographic algorithm*] and cryptographic key sizes [selection: *cryptographic key sizes*] that meet the following: [selection: *list of standards*]

The following table provides the allowed choices for completion of the selection operations of FCS_COP.1/AEAD.

Identifier	Cryptographic Algorithm	Cryptographic Algorithm Parameters	List of Standards
AES-CCM	AES in CCM mode with unpredictable, nonrepeating nonce, minimum size of 64 bits	[selection: ~~*128,*~~ *256]* bits	[selection: *ISO/IEC 18033-3:2010 (Subclause 5.2), FIPS PUB 197*] [AES] [selection: *ISO/IEC 19772:2020 (Clause 7), NIST SP 800-38C*] [CCM]
AES-GCM	AES in GCM mode with non-repeating IVs using [selection: *deterministic, RBG-based*], IV construction; the tag must be of length [selection: *96, 104, 112, 120, 128*] bits.	[selection: ~~*128*~~, *256*] bits	[selection: *ISO/IEC 18033-3:2010 (Subclause 5.2), FIPS PUB 197*] [AES] [selection: *ISO/IEC 19772:2020 (Clause 10), NIST SP 800- 38D*] [GCM]

FCS_COP.1/KeyWrap Cryptographic Operation - Key Wrapping

This is a selection-based SFR, to be included in the ST if "key wrapping" is selected in FCS_CKM.2.1.

FCS_COP.1.1/KeyWrap The TSF shall perform key wrapping in accordance with a specified cryptographic algorithm [selection: cryptographic algorithm] and cryptographic key sizes [selection: cryptographic key sizes] that meet the following: [selection: list of standards]

The following table provides the allowed choices for completion of the selection operations of FCS_COP.1/KeyWrap.

Identifier	Cryptographic Algorithm	Cryptographic Algorithm Parameters	List of Standards
AES-KW	AES in KW mode	256 bits	[selection: ISO/IEC 18033-3:2010 (Subclause 5.2), FIPS PUB 197] [AES] [selection: ISO/IEC 19772:2020 (clause 6), NIST SP 800-38F (Section 6.2)] [KW mode]
AES-KWP	AES in KWP mode	256 bits	[selection: ISO/IEC 18033-3:2010 (Subclause 5.2), FIPS PUB 197] [AES] NIST SP 800- 38F (Section 6.3) [KWP mode]
~~AES-CCM~~	~~AES in CCM mode with unpredictable, nonrepeating nonce, minimum size of 64 bits~~	~~256 bits~~	~~[selection: ISO/IEC 18033-3:2010 (Subclause 5.2), FIPS PUB 197] [AES] [selection: ISO/IEC 19772:2020 (clause 7), NIST SP 800- 38C CCM]~~

FCS_COP.1/SKC Cryptographic Operation - Symmetric Key Cryptography

This is a selection-based SFR, to be included in the ST if CBC mode, CTR mode, or XTS mode are selected in FCS_COP.1/DataEncryption.

FCS_COP.1.1/SKC The TSF shall perform symmetric-key encryption/decryption in accordance with a specified cryptographic algorithm [selection: *cryptographic algorithm*] and cryptographic key sizes [selection: *cryptographic key sizes*] that meet the following: [selection: *list of standards*] The following table provides the allowed choices for completion of the selection operations of FCS_COP.1/SKC.

Identifier	Cryptographic Algorithm	Cryptographic Algorithm Parameters	List of Standards
AES-CBC	AES in CBC mode with non-repeating and unpredictable IVs	[selection: ~~*128,*~~ *256]* bits	[selection: *ISO/IEC 18033-3:2010 (Subclause 5.2), FIPS PUB 197*] [AES] [selection: *ISO/IEC 10116:2017 (Clause 7), NIST SP 800-38A*] [CBC]

AES-CTR	AES in CTR mode with a non-repeating initial counter and with no repeated use of counter values across multiple messages with the same secret key	[selection: ~~*128,*~~ *256]* bits	[selection: *ISO/IEC 18033-3:2010 (Subclause 5.2), FIPS PUB 197*] [AES] [selection: *ISO/IEC 10116:2017 (Clause 10), NIST SP 800- 38A*] [CTR]
XTS-AES	AES in XTS mode with unique tweak values that are consecutive nonnegative integers starting at an arbitrary nonnegative integer	[selection: ~~*256,*~~ *512]* bits	[selection: *ISO/IEC 18033-3:2010 (Subclause 5.2), FIPS PUB 197*] [AES] [selection: *IEEE Std. 1619-2018, NIST SP 800- 38E*] [XTS]

FCS_RBG.1 Random Bit Generation

FCS_RBG.1.1 The TSF shall perform deterministic random bit generation services using [selection: DRBG algorithm] in accordance with [selection: list of standards] after initialization.

The following table provides the allowed choices for completion of the selection operations of FCS_RBG.1.1.

Identifier	**DRBG Algorithm**	**List of Standards**
HASH_DRBG	Hash_DRBG with [selection: ~~SHA-256,~~ SHA-384, SHA-512, ~~SHA3-256, SHA3-384, SHA3- 512~~]	[selection: ISO/IEC 18031: 2025 (Section C.2.2), NIST SP 800-90A Revision 1 Section 10.1.1]
HMAC_DRBG	HMAC_DRBG with [selection: ~~SHA-256,~~ SHA-384, SHA-512, ~~SHA3-256, SHA3-384, SHA3- 512~~]	[selection: ISO/IEC 18031: 2025 (Section C.2.3), NIST SP800-90A Revision 1 Section 10.1.2]
CTR_DRBG	CTR_DRBG with [selection: ~~AES-128, AES-192,~~ AES-256]	[selection: ISO/IEC 18031: 2025 (Section C.3.2), NIST SP800-90A Revision 1 Section 10.2.1]

FCS_RBG.1.2 The TSF shall use a [selection: TSF entropy source [assignment: *name of entropy source*], **multiple TSF entropy sources [assignment: *name of entropy sources*],** TSF interface for seeding] for initialized seeding.

FCS_RBG.2 Random Bit Generation (External Seeding - VS platform)

This component is included if the TOE uses a VS for DRBG seeding and "TSF interface for seeding" is selected in FCS_RBG.1.2

FCS_RBG.2.1 The TSF shall be able to accept a minimum input of [assignment: *minimum input length of 256 bits*] from a TSF interface for obtaining entropy.

FCS_RBG.3 Random Bit Generation (Internal Seeding - Single Source)

This component is included if "TSF entropy source" is selected in FCS_RBG.1.2

FCS_RBG.3.1 The TSF shall be able to seed the DRBG using a [selection, choose one of: *TSF software-based entropy source*, *TSF hardware-based entropy source*] [assignment: *name of entropy source*] with [assignment: *256*] bits of min-entropy.

FCS_RBG.4 Random Bit Generation (Internal Seeding - Multiple Sources)
This component is included if "multiple TSF entropy sources" is selected in FCS_RBG.1.2

FCS_RBG.4.1 The TSF shall be able to seed the DRBG using [selection: [assignment: number] TSF software-based entropy source(s), [assignment: *at least one*] TSF hardware-based entropy source(s)].

FCS_RBG.5 Random Bit Generation (Combining Entropy Sources)
This component is included if "multiple TSF entropy sources" is selected in FCS_RBG.1.2

FCS_RBG.5.1 The TSF shall [**selection**: *hash, concatenate and hash, XOR, input into a linear feedback shift register, [**assignment**: combining operation]*] [**selection**: *output from TSF entropy source(s), input from TSF interface(s) for obtaining entropy*] resulting in a minimum of [assignment: *256*] bits of min-entropy to create the entropy input into the derivation function as defined in [**selection**: *ISO/IEC 18031:2011, NIST SP 800-90A Revision 1*]

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 configurable to between [assignment: minimum number of characters supported by the TOE] and [assignment: number of characters greater than or equal to 15] characters.

FMT_SMF.1 Specification of management functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- *Ability to administer the TOE remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the remote session inactivity time before session termination;*
- *Ability to update the TOE, and to verify the updates using* digital signature *capability prior to installing those updates;*
- [at least the following underlined selections:
    - *Ability to start and stop services;*
    - *Ability to configure local audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full; changes to local audit storage size);*
    - *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*
    - *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 manage the cryptographic keys;*
    - *Ability to configure the cryptographic functionality;*
    - *Ability to configure thresholds for SSH rekeying;*
    - *Ability to configure the lifetime for IPsec SAs;*
    - *Ability to configure the list of supported (D)TLS ciphers;*
    - *Ability to configure the interaction between TOE components;*
    - *Ability to enable or disable automatic checking for updates or automatic updates;*
    - *Ability to re-enable an Administrator account;*
    - *Ability to set the time which is used for time-stamps;*
    - *Ability to configure NTP;*
    - *Ability to configure the reference identifier for the peer;*
    - *Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;*
    - *Ability to generate Certificate Signing Request (CSR) and process CA certificate response;*
    - *Ability to administer the TOE locally;*
    - *Ability to configure the local session inactivity time before session termination or locking;*
    - *Ability to configure the authentication failure parameters for FIA_AFL.1;*
    - *Ability to manage the trusted public keys database;*
    - *Ability to manage the public key or certificate used to validate the digital update;*
    - ~~*No other capabilities*~~].

FPT_STM_EXT.1 Reliable time stamps

FPT_STM_EXT.1.2 The TSF shall be capable of at least one of the following underlined selections [selection: *allow the Security Administrator to set the time, synchronise time with an NTP server, obtain time from the underlying virtualization system*].

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF **shall be capable of using** at least the following underlined **[selection:** ***IPsec, SSH as defined in the Functional Package for SSH, TLS as defined in the Functional Package for TLS, DTLS as defined in the Functional Package for TLS, HTTPS*****]** **to** provide a **trusted** communication channel between itself and ~~another trusted IT product~~ **authorized IT entities supporting the following capabilities: audit server, [selection:** ***authentication server, [assignment:*** *TLS Software Applications: HTTPS/TLS Clients on authorized End User Devices (EUDs),* ***other capabilities], no 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 ~~modification or~~ disclosure **and detection of modification of the channel data.**

Application Note: TLS Servers in CSfC Solutions must support HTTPS and/or TLS to/from the TLS Software Application. SSH, TLS, and/or IPsec are all acceptable selections for audit server connections in CSfC Solutions.

FTP_TRP.1/Admin Trusted path

FTP_TRP.1.1/Admin The TSF shall **be capable of using** at least one of the following **[selection:** ***IPsec, SSH as defined in the Functional Package for SSH, TLS as defined in the Functional Package for TLS, DTLS as defined in the Functional Package for TLS, HTTPS*****]** to provide a communication path between itself and authorized remote **Administrators** ~~users~~ that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from disclosure **and provides detection of modification of the channel data**.

FAU_STG.2 Protected audit data storage

FAU_STG.2.1 The TSF shall protect the stored audit data in the audit trail from unauthorized deletion.

FAU_STG.2.2 The TSF shall be able to prevent unauthorized modifications to the stored audit data in the audit trail.

FPT_ITT.1 Basic internal TSF data transfer protection

FPT_ITT.1.1 If applicable due to a distributed TOE, 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:** ***IPsec, SSH as defined in the Functional Package for SSH, TLS as defined in the Functional Package for TLS, DTLS as defined in the Functional Package for TLS, HTTPS*****]**.

FAU_STG_EXT.5 Protected Remote Audit Event Storage for Distributed TOEs

FAU_STG_EXT.5.1 If applicable due to a distributed TOE, each TOE component which does not store security audit data locally shall be able to buffer security audit data locally until it has been transferred to another TOE component that stores or forwards it. All transfer of audit records between TOE components shall use a protected channel according to [**Selection:** *FPT_ITT.1, FTP_ITC.1*].

Security Assurance Requirements for Flaw Remediation (ALC_FLR)

A.8.3. Systematic flaw remediation (ALC_FLR.3)

This component is targeted at the flaw remediation procedures applied by the developer to ensure that all reported security flaws in each release of the TOE are tracked and corrected. In addition, the developer's flaw remediation guidance is analysed to ensure that users are aware how to correctly report security flaws to the developer. Flaw remediation procedures of the developer need to describe how users can register to receive flaw reports and corrections. The procedures also need to ensure timely responses to reports of security flaws and automatic distribution of security flaw reports. The evaluator performs the CEM work units associated with ALC_FLR.3.

Functional Package for Transport Layer Security (TLS) Version 2.1 Selections

FCS_TLS_EXT.1 TLS Protocol

FCS_TLS_EXT.1.1 The TSF shall implement [**selection**:

- *TLS as a client*
- *TLS as a server*
- ~~*DTLS as a client*~~
- *DTLS as a server*

].

FCS_TLSS_EXT.1 TLS Server Protocol

The inclusion of this selection-based component depends upon selection in FCS_TLS_EXT.1.1.

FCS_TLSS_EXT.1.1 The TSF shall implement [**selection**: *TLS 1.2 (RFC 5246)*, *TLS 1.3 (RFC 8446)*] as a server that supports additional functionality for session renegotiation protection and [**selection**:

- *mutual authentication*
- *supplemental downgrade protection*
- ~~*session resumption*~~
- ~~*no optional functionality*~~

] and shall reject connection attempts from clients supporting only TLS 1.1, TLS 1.0, or SSL versions.

FCS_TLSS_EXT.1.2 The TSF shall be able to support the following [**selection**:

- *TLS 1.2 ciphersuites: [**selection**:*
  - *CNSA 1.0 compliant [**selection**:*

- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and RFC 8422*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and RFC 8422*
- ~~*TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*~~
- *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 and RFC 8422*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 and RFC 8422*
- ~~*ciphersuites using pre-shared secrets: [**selection:***~~
  - ~~*TLS_ECDHE_PSK_WITH_AES_256_GCM_SHA384 as defined in RFC 8442*~~
  - ~~*TLS_DHE_PSK_WITH_AES_256_GCM_SHA384 as defined in RFC 5487*~~
  - ~~*TLS_RSA_PSK_WITH_AES_256_GCM_SHA384 as defined in RFC 5487*~~

  ~~*]*~~

]

- ~~*non-CNSA compliant [**selection:***~~
  - ~~*TLS_RSA_WITH_AES_256_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_GCM_SHA256 as defined in RFC 5289*~~
  - ~~*TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*~~
  - ~~*TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*~~
  - ~~*TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*~~
  - ~~*TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*~~
  - ~~*TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*~~
  - ~~*TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246]*~~
  - ~~*ciphersuites using pre-shared secrets: [**selection:***~~
    - ~~*TLS_ECDHE_PSK_WITH_AES_128_GCM_SHA256 as defined in RFC 8442*~~
    - ~~*TLS_DHE_PSK_WITH_AES_128_GCM_SHA256 as defined in RFC 5487*~~
    - ~~*TLS_RSA_PSK_WITH_AES_128_GCM_SHA256 as defined in RFC 5487]*~~

    ~~*]*~~

~~*]*~~

] and no other TLS 1.2 ciphersuites,

- *TLS 1.3 ciphersuites* ***[selection:***
  - *CNSA 2.0 compliant TLS_AES_256_GCM_SHA384 as defined in RFC 8446 and no other TLS 1.3 ciphersuites*

- ~~*non-CNSA compliant [**selection**:*~~
  - ~~*TLS_AES_128_GCM_SHA256 as defined in RFC 8446*~~
  - ~~*[**assignment**: other TLS 1.3 ciphersuites]*~~

  ~~]~~

]

] offering the supported ciphersuites in a ClientHello message in preference order: [**assignment**: *list of supported ciphersuites*].

FCS_TLSS_EXT.1.4 The TSF shall be able to process the following TLS ClientHello message extensions:

- signature_algorithms extension (RFC 8446) indicating support for CNSA 1.0 compliant [**selection**:
  - *ecdsa_secp384r1_sha384 (RFC 8446)*
  - *rsa_pkcs1_sha384 (RFC 8446)*

  ], and [**selection**:
  - *CNSA 1.0 compliant [**selection**:*
    - *rsa_pss_pss_sha384 (RFC 8446)*
    - *rsa_pss_rsae_sha384 (RFC 8446)*

    ]
  - ~~*[**assignment**: other non-deprecated, non-CNSA compliant signature algorithms]*~~

  ] and no other signature algorithms, and

[**selection**:

- *signature_algorithms_cert extension (RFC 8446) indicating support for CNSA 1.0 compliant [**selection**:*
  - *ecdsa_secp384r1_sha384 (RFC 8446)*
  - *rsa_pkcs1_sha384 (RFC 8446)*

  *], and [**selection**:*
  - *CNSA 1.0-compliant [**selection**:*
    - *rsa_pss_pss_sha384 (RFC 8446)*
    - *rsa_pss_rsae_sha384 (RFC 8446)*

    ]
  - ~~*non-CNSA compliant [**selection**:*~~
    - ~~*rsa_pkcs1_sha256 (RFC 8446)*~~
    - ~~*rsa_pss_rsae_sha256 (RFC 8446)*~~

    ~~]~~
  - ~~*[**assignment**: other non-deprecated, non-CNSA compliant signature algorithms]*~~

  ] and no other signature algorithms
- *supported_versions extension (RFC 8446) indicating support for TLS 1.3 and [**selection**: TLS 1.2, no other versions]*
- *supported_groups extension indicating support for [**selection**:*
  - *CNSA 1.0 compliant [**selection**:*
    - *secp384r1 (RFC 8446)*

- *ffdhe3072 (RFC 7919)*
- *ffdhe4096 (RFC 7919)*

]

- *~~non-CNSA compliant [**selection**:~~*
  - *~~secp256r1 (RFC 8446)~~*
  - *~~ffdhe2048 (RFC 7919)~~*

  ~~]~~
- *and [**selection**:*
  - *~~secp521r1 (RFC 8446)~~*
  - *ffdhe6144(RFC 7919)*
  - *ffdhe8192 (RFC 7919)*
  - *no other supported groups*

  ]

]

- *key_share extension (RFC 8446)*
- *post_handshake_auth (RFC 8446), pre_shared_key (RFC 8446), tls_cert_with_extern_psk (RFC 8773), and psk_key_exchange_modes (RFC 8446) indicating psk_dhe_ke (DHE or ECDHE) mode*
- *extended_master_secret extension (RFC 7627) enforcing server support, and [**selection**: allowing legacy clients, no other enforcement mode]*
- *no other extensions*

].

FCS_TLSS_EXT.1.5 The TSF shall perform key establishment for TLS using [**selection**:

- *RSA with [**selection**:*
  - *CNSA 1.0 compliant size [**selection**: 3072, 4096]*
  - *~~non-CNSA compliant size 2048~~*

  ] bits and no other sizes
- *[**selection**:*
  - *CNSA 1.0 compliant Diffie-Hellman groups [**selection**: ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192]*
  - *~~non-CNSA compliant Diffie-Hellman group ffdhe2048~~*

  *] and no other groups, consistent with the client's supported_groups extension and [**selection**: key_share extension, no other] extension*
- *ECDHE parameters using [**selection**:*
  - *CNSA 1.0 compliant elliptic curves [**selection**: secp384r1, ~~secp521r1~~]*
  - *~~non-CNSA compliant elliptic curve secp256r1~~*

  *] and no other curves, consistent with the client's supported_groups extension and [**selection**: key_share extension, no other] extension and using non-compressed formatting for points*

].

FCS_TLSS_EXT.4 TLS Server Support for Renegotiation

The inclusion of this selection-based component depends upon selection in FCS_TLS_EXT.1.1.

FCS_TLSS_EXT.4.1 The TSF shall support secure TLS renegotiation through the use of [**selection**: *the "renegotiation_info" TLS extension in accordance with RFC 5746, not allowing session renegotiation*].

FCS_TLSS_EXT.4.2 The TSF shall [**selection**: *indicate support for renegotiating a TLS 1.2 session by including the renegotiation_info extension in the ServerHello message when a ClientHello with the renegotiation_info extension is received and shall terminate a session if neither of the renegotiation_info or TLS_EMPTY_RENEGOTIATION_INFO_SCSV signaling ciphersuites are indicated in the client hello, not allow renegotiation*].

FCS_DTLSS_EXT.1 DTLS Server Protocol

The inclusion of this selection-based component depends upon selection in FCS_TLS_EXT.1.1.

FCS_DTLSS_EXT.1.1 The TSF shall implement [**selection**: ~~*DTLS 1.2 (RFC 6347)*~~, *DTLS 1.3 (RFC 9147)*] as a server that supports additional functionality for session renegotiation protection and [**selection**:

- *mutual authentication*
- *supplemental downgrade protection*
- ~~*session resumption*~~
- ~~*no optional functionality*~~

] and shall reject connection attempts from clients supporting only DTLS 1.0.

FCS_DTLSS_EXT.1.2 The TSF shall be able to support the following [**selection**:

- *TLS 1.2 ciphersuites: [**selection**:*
  - *CNSA 1.0 compliant [**selection**:*
    - *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and RFC 8422*
    - *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 and RFC 8422*
    - ~~*TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*~~
    - *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
    - *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 and RFC 8422*
    - *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 and RFC 8422*
  - ~~*ciphersuites using pre-shared secrets: [**selection**:*~~
    - ~~*TLS_ECDHE_PSK_WITH_AES_256_GCM_SHA384 as defined in RFC 8442*~~

- ~~*TLS_DHE_PSK_WITH_AES_256_GCM_SHA384 as defined in RFC 5487*~~
- ~~*TLS_RSA_PSK_WITH_AES_256_GCM_SHA384 as defined in RFC 5487*~~

~~*]*~~

]

- ~~*non-CNSA compliant [**selection**:*~~
  - ~~*TLS_RSA_WITH_AES_256_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_GCM_SHA256 as defined in RFC 5289*~~
  - ~~*TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*~~
  - ~~*TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*~~
  - ~~*TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*~~
  - ~~*TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*~~
  - ~~*TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*~~
  - ~~*TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 5246]*~~
  - ~~*ciphersuites using pre-shared secrets: [**selection**:*~~
    - ~~*TLS_ECDHE_PSK_WITH_AES_128_GCM_SHA256 as defined in RFC 8442*~~
    - ~~*TLS_DHE_PSK_WITH_AES_128_GCM_SHA256 as defined in RFC 5487*~~
    - ~~*TLS_RSA_PSK_WITH_AES_128_GCM_SHA256 as defined in RFC 5487]*~~

  ~~*]*~~

~~*]*~~

] and no other TLS 1.2 ciphersuites,

- *TLS 1.3 ciphersuites* ***[selection**:*
  - *CNSA 2.0 compliant TLS_AES_256_GCM_SHA384 as defined in RFC 8446 and no other TLS 1.3 ciphersuites*
  - ~~*non-CNSA compliant [**selection**:*~~
    - ~~*TLS_AES_128_GCM_SHA256 as defined in RFC 8446*~~
    - ~~*[**assignment**: other TLS 1.3 ciphersuites]*~~

  ~~*]*~~

]

] offering the supported ciphersuites in a ClientHello message in preference order: [**assignment**: *list of supported ciphersuites*].

FCS_DTLSS_EXT.1.4 The TSF shall be able to support the following DTLS ClientHello message extensions:

- signature_algorithms extension (RFC 8446) indicating support for CNSA 1.0 compliant [**selection**:
  - *ecdsa_secp384r1_sha384 (RFC 8446)*

- *rsa_pkcs1_sha384 (RFC 8446)*

], and [**selection**:

- *CNSA 1.0 compliant [**selection**:*
  - *rsa_pss_pss_sha384 (RFC 8446)*
  - *rsa_pss_rsae_sha384 (RFC 8446)*

]

- ~~*[**assignment**: other non-deprecated, non-CNSA compliant signature algorithms]*~~

] and no other signature algorithms, and

[**selection**:

- *signature_algorithms_cert extension (RFC 8446) indicating support for CNSA 1.0 compliant [**selection**:*
  - *ecdsa_secp384r1_sha384 (RFC 8446)*
  - *rsa_pkcs1_sha384 (RFC 8446)*

*], and [**selection**:*

- *CNSA 1.0-compliant [**selection**:*
  - *rsa_pss_pss_sha384 (RFC 8446)*
  - *rsa_pss_rsae_sha384 (RFC 8446)*

]

- ~~*non-CNSA compliant [**selection**:*~~
  - ~~*rsa_pkcs1_sha256 (RFC 8446)*~~
  - ~~*rsa_pss_rsae_sha256 (RFC 8446)*~~

~~*]*~~

- ~~*[**assignment**: other non-deprecated, non-CNSA compliant signature algorithms]*~~

] and no other signature algorithms

- *supported_versions extension (RFC 8446) indicating support for DTLS 1.3 and [**selection**: DTLS 1.2, no other versions]*
- *supported_groups extension indicating support for [**selection**:*
  - *CNSA 1.0 compliant [**selection**:*
    - *secp384r1 (RFC 8446)*
    - *ffdhe3072 (RFC 7919)*
    - *ffdhe4096 (RFC 7919)*

    ]
  - ~~*non-CNSA compliant [**selection**:*~~
    - ~~*secp256r1 (RFC 8446)*~~
    - ~~*ffdhe2048 (RFC 7919)*~~

    ~~*]*~~
  - *and [**selection**:*
    - ~~*secp521r1 (RFC 8446)*~~
    - *ffdhe6144(RFC 7919)*
    - *ffdhe8192 (RFC 7919)*
    - *no other supported groups*

    ]

]

- *key_share extension (RFC 8446)*
- *post_handshake_auth (RFC 8446), pre_shared_key (RFC 8446), tls_cert_with_extern_psk (RFC 8773), and psk_key_exchange_modes (RFC 8446) indicating psk_dhe_ke (DHE or ECDHE) mode*
- *extended_master_secret extension (RFC 7627) enforcing server support, and [**selection**: allowing legacy clients, no other enforcement mode]*
- *no other extensions*

].

FCS_DTLSS_EXT.1.5 The TSF shall perform key establishment for DTLS using [**selection**:

- *RSA with [**selection**:*
  - *CNSA 1.0 compliant size [**selection**: 3072, 4096]*
  - ~~*non-CNSA compliant size 2048*~~

  ] bits and no other sizes
- *[**selection**:*
  - *CNSA 1.0 compliant Diffie-Hellman groups [**selection**: ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192]*
  - ~~*non-CNSA compliant Diffie-Hellman group ffdhe2048*~~

  *] and no other groups, consistent with the client's supported_groups extension and [**selection**: key_share extension, no other] extension*
- *ECDHE parameters using [**selection**:*
  - *CNSA 1.0 compliant elliptic curves [**selection**: secp384r1, ~~secp521r1~~]*
  - ~~*non-CNSA compliant elliptic curve secp256r1*~~

  *] and no other curves, consistent with the client's supported_groups extension and [**selection**: key_share extension, no other] extension and using non-compressed formatting for points*

].

FCS_DTLSS_EXT.4 DTLS Server Support for Renegotiation

The inclusion of this selection-based component depends upon selection in FCS_TLS_EXT.1.1.

FCS_DTLSS_EXT.4.1 The TSF shall support secure DTLS renegotiation through the use of [**selection**: *the "renegotiation_info" TLS extension in accordance with RFC 5746, not allowing session renegotiation*].

FCS_DTLSS_EXT.4.2 The TSF shall [**selection**:

- *indicate support for renegotiating a DTLS 1.2 session by including the renegotiation_info extension in the ServerHello message and shall terminate a DTLS 1.2 session if neither of the renegotiation_info or TLS_EMPTY_RENEGOTIATION_INFO_SCSV signaling ciphersuites are indicated in the ClientHello*
- *not allow renegotiation*

].

FIA_XCU_EXT.1 Implementation of X.509 Functions

FIA_XCU_EXT.1.1 The TSF shall [**selection**: *verify, assert*] identities included in X.509 certificates.

Application Note: The ST author claims "verify" when the TOE associates identities included in X.509 certificates with users, external entities or inter-TOE entities authorized to exercise TOE functionality. The ST author claims "assert" when the TOE represents itself or its functions to internal or external entities.

If "verify" is selected, then FIA_X509_EXT.1 and FIA_X509_EXT.2 must be included. If "assert" is selected, FIA_XCU_EXT.2 must be included.

FIA_ESTC_EXT.1 EST Client Certificate Enrollment

The inclusion of this selection-based component depends upon `if the TOE uses EST and` ***selection in FIA_X509_EXT.3.1.***

FIA_ESTC_EXT.1.4 The TSF shall [**selection**: *invoke platform-provided functionality*, *provide functionality*] to authenticate its certificate enrollment request to receive [**assignment**: *list of certificates*] from an authorized EST server using [**selection**:

- ~~*HTTP basic authentication transported over TLS (HTTPS) in accordance with RFC 7030 section 3.2.3*~~
- ~~*HTTP digest authentication using a cryptographic hash algorithm transported over TLS (HTTPS) in accordance with RFC 7030 section 3.2.3*~~
- *Certificate-based authentication in accordance with RFC 7030 section 3.3.2 using [**assignment**: pre-existing certificate authorized by the EST server]*

].

FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to validate certificates in accordance with the following rules:

- Certification path validation meets requirements of RFC 5280 for certificate paths of [**selection**: *unlimited path length*, *maximum path length of [**assignment**: number greater than or equal to 0] certificates*] and certificate paths exceeding the maximum path length are invalid.

- The current time is within the notBefore and notAfter values of all certificates in the certification path.
- The certification path shall terminate at a trust anchor element appropriate for the supported function.
- Certificates containing subjectUniqueID or issuerUniqueID fields are considered invalid.
- Certificates are signed using cryptographic signatures and hashes in accordance with RFC 8603, and [**selection**:
    - *[**assignment**: list of supported cryptographic algorithms]*
    - *no other algorithms*

  ] and certificates signed using other cryptographic algorithms are considered invalid.
- [**selection**:
    - *CRLs are signed using cryptographic signatures and hashes in accordance with RFC 8603 and [**selection**:*
        - *[**assignment**: list of supported cryptographic algorithms]*
        - *no other algorithms*

      ] and CRLs signed using other cryptographic algorithms are considered invalid;
    - *OCSP responses are signed using [**selection**:*
        - *sha384WithRSAEncryption with key size of 3072 bits or greater,*
        - *ecdsa-with-SHA384 using [**selection**: secp384r1, ~~secp521r1~~],*
        - *ecdsa-with-SHA512 using [**selection**: secp384r1, ~~secp521r1~~],*

      *] and [**selection**:*
        - *[**assignment**: list of other supported algorithms]*
        - *no other algorithms*

      ] requested using the preferredSignatureAlgorithm extension and OCSP responses are considered invalid if using other algorithms;
    - ~~*no other algorithm constraints*~~

]

FIA_X509_EXT.1.3 The TSF shall [**selection**: *invoke platform-provided functionality*, *implement functionality*] to validate revocation status of the certificate using `at least one of` [**selection**:

- *The Online Certificate Status Protocol (OCSP) as specified in RFC 6960*
- *Certificate Revocation Lists (CRL) as specified in RFC 5280 and refined by RFC 8603*

- *Certificate Revocation Lists as specified in RFC 5280*
- *Based on validity period: Certificates expiring within [**assignment**: time less than 24 hours] of the current time are considered valid when no other valid revocation status information is available*
- *Administrative notification of revocation: [**assignment**: administrative action upon notification] using [**assignment**: method to invalidate use of certificates in supported functions] when the certificate is revoked.*
- *Direct association with Certification Authority: [**assignment**: direct revocation status information implementations]*

].

FIA_X509_EXT.2 X.509 Certificate Support for Functions

FIA_X509_EXT.2.1The TSF shall [**selection**: *invoke platform-provided functionality to validate, validate*] X.509v3 certificates in accordance with FIA_X509_EXT.1 to support [**assignment**: *supported functions*] using [**selection**:

- *[**selection**: TLS, DTLS, IPsec or IKE , SMIME, SSH, [**assignment**: other authenticated communications protocol]]*
- *[**selection**: code signing for system software updates, code signing for software integrity testing, integrity verification for TSF protected data, administrator authentication, user authentication , [**assignment**: other uses]]*

]

FIA_X509_EXT.2.2 For each function indicated in FIA_X509_EXT.2.1, the TSF shall [**selection**: *invoke the TOE platform to determine, determine*] whether the [**selection**: *administrator is allowed to configure certificate acceptance, supported function determines acceptance via [**assignment**: method of determining acceptance], ~~certificate is accepted~~, certificate is not accepted*] when valid certificate revocation status information cannot be obtained from a source indicated in FIA_X509_EXT.1.3.

FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1 The TSF shall [**selection**: *invoke the TOE platform to generate, generate*] Certificate Requests as specified by [**selection**:

- *RFC 2986 (PKCS-10)*

] and [selection:

- *RFC 7030 as updated by RFC 8996 (EST)*
- *RFC 5272 as updated by RFC 6402 (CMC)*
- *RFC 5272 as updated by RFC 8756 (CNSA CMC)*
- *RFC 4210 as updated by RFC 6712 and RFC 9481 (v2 CMP)*
- *RFC 4210 as updated by RFC 6712 and RFC 9480 (v3 CMP)*
- *no other*

]

] and be able to provide the following information in the request: public key, [**selection**:

- *Subject DN consisting of values for [**selection**:*
  - *U*
  - *O*
  - *OU*
  - *CN*
  - *[**assignment**: other subject attributes]*

]

- *one or more of the following SAN types [**selection**:*
  - *rfc822Name*
  - *dNSName*
  - *directoryName*
  - *uniformResourceIdentifier*
  - *iPAddress*
  - *[**assignment**: other SAN types]*

]

] and [**selection**:

- *[**assignment**: list of other certificate field and extension values]*
- *[**assignment**: list of identifying information]*
- *no other information.*

]

Application Note: The supported certificate request mechanisms are claimed in the first selection. PKCS-10 is claimed whether a manual request process is used or if the PKCS-10 request is posted to the certification authority (for embedded CAs or as in ACME implementations). Other options embed the certificate request in a formalized certificate management protocol. If EST is claimed, FIA_ESTC_EXT.1 is also claimed; if CMC or CNSA CMC are claimed, FIA_CMCC_EXT.1 is also claimed; if v2 or v3 CMP is claimed, FIA_CMPC_EXT.1 is also claimed.

FIA_XCU_EXT.2 X.509 Certificate Acquisition

FIA_XCU_EXT.2.1 The TSF shall [**selection**:

- *request certificates from an [**selection**: external, embedded] CA,*
- *obtain certificates from an embedded CA*

] to represent [**assignment**: *TOE functions*] for [**selection**:

- *[**selection**: TLS, DTLS, IPsec or IKE, SMIME, SSH, [**assignment**: other authenticated communications protocol]]*
- *[**selection**: code signing for system software updates, code signing for software integrity testing, integrity verification for TSF protected data,*

*administrator authentication, user authentication, [**assignment**: other uses]]*

].

Functional Package for Secure Shell (SSH) Version 2.0 Selections

FCS_SSH_EXT.1

If the TOE uses SSH, the auditable events specified in this Package are included in an ST if the incorporating PP, cPP, or PP-Module supports audit event reporting through FAU_GEN.1, and if all other criteria in the incorporating PP or PP-Module are met.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_SSH_EXT.1		
	[selection: *Failure to establish SSH connection, None*]	**[selection:** *Reason for failure and non-TOE endpoint of attempted connection (IP Address), No additional information*]
	[selection: *Establishment of SSH connection, None*]	[selection: *Non-TOE endpoint of connection (IP Address), No additional information*]
	[selection: *Termination of SSH connection session, None*]	[selection: *Non-TOE endpoint of connection (IP Address), No additional information*]
	[selection: *Dropping of packets outside defined size limits, None*]	**[selection:** *Packet size, No additional information*]

FCS_SSH_EXT.1.1

If the TOE uses SSH, the TOE shall implement SSH acting as a [**selection**: *client, server*] that complies with RFCs 4251, 4252, 4253, 4254, [**selection**: *4256, ~~4344~~, 5647, 5656, 6187, 6668, 8268, 8308, 8332, ~~no other RFCs~~] and [no other standard*].

FCS_SSH_EXT.1.2

If the TOE uses SSH, the TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [**selection**:

- *password, complying with [**selection**: RFC 4252, RFC 4256 keyboard-interactive methods]*
- *publickey" (RFC 4252): [**selection**:*
    - *rsa-sha2-512 (RFC 8332)*
    - *ecdsa-sha2-nistp384 (RFC 5656)*
    - *~~ecdsa-sha2-nistp521 (RFC 5656)~~*

    - *x509v3-ecdsa-sha2-nistp384 (RFC 6187)*
    - ~~*x509v3-ecdsa-sha2-nistp521 (RFC 6187)*~~

  ]

] and no other methods.

FCS_SSH_EXT.1.4

If the TOE uses SSH, the TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [**selection**:

- *AEAD_AES_256_GCM (RFC 5647)*
- *aes256-gcm@openssh.com (RFC 5647)*

] and no other mechanisms.

FCS_SSH_EXT.1.5

If the TOE uses SSH, the TSF shall protect data in transit from modification, deletion, and insertion using: [**selection**:

- *AEAD_AES_256_GCM (RFC 5647)*
- *implicit*

] and no other mechanisms.

FCS_SSH_EXT.1.6

If the TOE uses SSH, the TSF shall stablish a shared secret with its peer using: [**selection**:

- *diffie-hellman-group15-sha512 (RFC 8268)*
- *diffie-hellman-group16-sha512 (RFC 8268)*
- ~~*diffie-hellman-group17-sha512 (RFC 8268)*~~
- ~~*diffie-hellman-group18-sha512 (RFC 8268)*~~
- *ecdh-sha2-nistp384 (RFC 5656)*
- ~~*ecdh-sha2-nistp521 (RFC 5656)*~~

] and no other mechanisms.

FCS_SSHC_EXT.1.1

If the TOE has an SSH client, the TSF shall authenticate its peer (SSH server) using: [**selection**:

- *a local database by associating each host name with a public key corresponding to the following list: [**selection**:*
    - *rsa-sha2-512 (RFC 8332)*
    - *ecdsa-sha2-nistp384 (RFC 5656)*
    - ~~*ecdsa-sha2-nistp521 (RFC 5656)*~~

  ]
- *a list of trusted certification authorities when the public key is in the following formats: [**selection**:*
    - *x509v3-ecdsa-sha2-nistp384 (RFC 6187)*
    - ~~*x509v3-ecdsa-sha2-nistp521 (RFC 6187)*~~

  ]

] as described in RFC 4251, Section 4.1.

FCS_SSHS_EXT.1.1

If the TOE has an SSH server, the TSF shall authenticate itself to its peer (SSH client) using:
[**selection**:

- *rsa-sha2-512 (RFC 8332)*
- *ecdsa-sha2-nistp384 (RFC 5656)*
- ~~*ecdsa-sha2-nistp521 (RFC 5656)*~~
- *x509v3-ecdsa-sha2-nistp384 (RFC 6187)*
- ~~*x509v3-ecdsa-sha2-nistp521 (RFC 6187)*~~

]