Proposed SFR Updates to HCD PP for Version 1.1

I. New Proposed Changes

Key: Proposed changes are in red.

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 shutdown of the audit functions;
b) All auditable events for the **not specified** level of audit;

c Resetting passwords (name of related user account shall be logged) (Version 1.1); and

d) All auditable events specified in Table 1, [assignment: *other specifically defined auditable events*].

FAU_STG_EXT.1 Extended: External Audit Trail Storage

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.

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.

FMT_MTD.1/CryptoKeys Management of TSF data

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

FPT_STM_EXT.1 Extended: Reliable Time Stamps

FPT_STM_EXT.1.1 The TSF shall be able to provide reliable time stamps.

FPT_STM_EXT.1.2 The TSF shall [selection: allow the Security Administrator to set the time, synchronise time with external time sources].

FTA_SSL.3 TSF-initiated termination

FTA_SSL.3.1 The TSF shall terminate an interactive session after a Security Administrator-configurable time interval of session inactivity.

FCS_HTTPS_EXT TSF-initiated termination

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 HTTPS using TLS as specified in FCS_TLS_EXT.1.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [selection: not require client authentication, not establish the connection, request authorization to establish the connection, *[assignment: other action]*]] if the peer certificate is deemed invalid.

FCS_IPSEC_EXT Extended: IPsec selected

FCS_IPSEC_EXT.1.11 The TSF shall generate the secret value x used in the IKE DiffieHellman key exchange (“x” in g^x mod 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.

FCS_IPSEC_EXT.1.12 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
] .

FCS_IPSEC_EXT.1.13 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.

FCS_IPSEC_EXT.1.14 The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following types: [selection: IP address, Fully Qualified Domain Name (FQDN), user FQDN, Distinguished Name (DN)] and [selection: no other reference identifier type, [*assignment: other supported reference identifier types*]].

FCS_TLS_EXT.1 Extended: TLS selected (TLS Client)

FCS_TLS_EXT.1.1 Same as current HCD PP FCS_TLS_EXT.1.1

FCS_TLSC_EXT.1.2 The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [selection: not establish the connection, request authorization to establish the connection, *[assignment: other action]*]

FCS_TLSC_EXT.1 TLS Server Protocol

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [selection: *TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)*] supporting the following ciphersuites:

Mandatory Ciphersuites:
• TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:
[selection:
• *None*
• *TLS_RSA_WITH_AES_256_CBC_SHA*
• *TLS_DHE_RSA_WITH_AES_128_CBC_SHA*
• *TLS_DHE_RSA_WITH_AES_256_CBC_SHA*
• *TLS_RSA_WITH_AES_128_CBC_SHA256*
• *TLS_RSA_WITH_AES_256_CBC_ SHA256*
• *TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256*
• *TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256*
• *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA*
• *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA*
• *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA*
• *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA*
• *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256*
• *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384*
• *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256*
• *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384*
• *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256*
• *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384*
• *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256*
• *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384*
].

FCS_TLSS_EXT.1.2 The TSF shall [selection: perform RSA key establishment with key size [selection: 2048 bits, 3072 bits, 4096 bits]; generate EC Diffie-Hellman parameters over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; generate DiffieHellman parameters of size [selection: 2048, bits, 3072 bits]].

FCS_TLSS_EXT.1.3 The TSF shall not establish a trusted channel if the client certificate is invalid. If the client certificate is deemed invalid, then the TSF shall [selection: not establish the connection, request authorization to establish the connection, *[assignment: other action]*].

FPT_APW_EXT Protection of Administrator Passwords

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.

FPT_TUP_EXT Extended: Trusted Update

FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide authorized administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide authorized administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3 The TSF shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and [selection: *published hash*, *no other functions*] prior to installing those updates.

FPT_TUD_EXT.2 Trusted Update based on certificates

FPT_TUD_EXT.2.1 The TSF shall not install an update if the code signing certificate is deemed invalid.

FCS_COP.1(e) Cryptographic Operation (Key Transport)

FCS_COP.1.1(e) Refinement: The TSF shall perform [*key transport*] in accordance with a specified cryptographic algorithm [*RSA in the following modes* ***[selection: KTS-OAEP, KTS-KEM-KWS]***] and the cryptographic key size [***selection: 2048 bits, 3072 bits***] that meet the following: [*NIST SP 800-56B, Revision 1*].

FCS_COP.1(d) Cryptographic Operation (Key Wrapping)

FCS_COP.1.1(d) Refinement: The TSF shall perform [*key wrapping*] in accordance with a specified cryptographic algorithm [*AES*] **in the following modes [selection: KW, KWP, GCM, CCM]** and the cryptographic key size [***selection: 128 bits, 256 bits***] that meet the following: [*AES as specified in ISO/IEC 18033-3,* ***[selection: NIST SP 800-38F, ISO/IEC 19772, no other standards]***]

FCS_PCC_EXT.1 Extended: Cryptographic Password Construct and Conditioning

FCS_PCC_EXT.1.1 A password used by the TSF to generate a password authorization factor shall enable up to [*assignment: positive integer of 64 or more*] characters in the set of {upper case characters, lower case characters, numbers, and [*assignment: other supported special characters*]} and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm HMAC-[selection: SHA-256, SHA-512], with [*assignment: positive integer of 1000 or more*] iterations, and output cryptographic key sizes [selection: 128 bits, 256 bits] that meet the following: [*NIST SP 800-13*].

II. NIAP Technical Decisions

FCS_CKM.4 in the HCD PP is replaced with the following:

FCS_CKM.4.1(a) The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [selection:

- *For volatile memory, the destruction shall be executed by a [selection: single overwrite consisting of [selection: a pseudo-random pattern using the TSF's RBG, zeroes, ones, a new value of a key, [assignment: any value that does not contain any CSP]], removal of power to the memory, destruction of reference to the key directly followed by a request for garbage collection];*
- *For non-volatile memory the destruction shall be executed by a [selection: [selection: single, [assignment: ST author defined multi-pass]] overwrite consisting of [selection: zeroes, ones, pseudo-random pattern, a new value of a key of the same size, [assignment: any value that does not contain any CSP]], block erase];*

]

that meets the following: *No Standard*.

> ***Application Note:*** *In the first selection, the ST Author is presented options for destroying disused cryptographic keys based on whether they are in volatile memory or non-volatile memory within the TOE.*
>
> *The selection of block erase for non-volatile memory applies only to flash memory.*
>
> *Within the selections is the option to overwrite the memory location with a new value of a key. The intent is that a new value of a key (as specified in another SFR within the PP) can be used to "replace" an existing key.*
>
> *Several selections allow assignment of a 'value that does not contain any CSP'. This means that the TOE uses some other specified data not drawn from a source that may contain key material or reveal information about key material, 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.*

TD0253: Assurance Activities for Key Transport

Publication Date
2017.11.08

Protection Profiles
PP_HCD_V1.0

Other References
FCS_COP.1.1(i)

Issue Description

There is no assurancy activity for the key transport SFR.

Resolution

Effective 06 February 2018, the following assurance activites are added for **FCS_COP.1(i) Cryptographic operation (Key Transport)**:

Assurance Activity

TSS

The evaluator shall verify the TSS provides a high level description of the RSA scheme and the cryptographic key size that is being used, and that the asymmetric algorithm being used for key transport is RSA. If more than one scheme/key size are allowed, then the evaluator shall make sure and test all combinations of scheme and key size. There may be more than one key size to specify – an RSA modulus size (and/or encryption exponent size), an AES key size, hash sizes, MAC key/MAC tag size.

If the KTS-OAEP scheme was selected, the evaluator shall verify that the TSS identifies the hash function, the mask generating function, the random bit generator, the encryption primitive and decryption primitive.

If the KTS-KEM-KWS scheme was selected, the evaluator shall verify that the TSS identifies the key derivation method, the AES-based key wrapping method, the secret value encapsulation technique, and the random number generator.

Operational Guidance

There are no AGD evaluation activities for this SFR.

KMD

There are no KMD evaluation activities for this SFR.

Test

For each supported key transport schema, the evaluator shall initiate at least 25 sessions that require key transport with an independently developed remote instance of a key transport entity, using known RSA key-pairs. The evaluator shall observe traffic passed from the sender-side and to the receiver-side of the TOE, and shall perform the following tests, specific to which key transport scheme was employed.

If the KTS-OAEP scheme was selected, the evaluator shall perform the following tests:

1. The evaluator shall inspect each cipher text, C, produced by the RSA-OAEP encryption operation of the TOE and make sure it is the correct length, either 256 or 384 bytes depending on RSA key size. The evaluator shall also feed into the TOE's RSA-OEAP decryption operation some cipher texts that are the wrong length and verify that the erroneous input is detected and that the decryption operation exits with an error code.

2. The evaluator shall convert each cipher text, C, produced by the RSA-OAEP encryption operation of the TOE to the correct cipher text integer, c, and use the decryption primitive to compute em = RSADP((n,d),c) and convert em to the encoded message EM. The evaluator shall then check that the first byte of EM is 0x00. The evaluator shall also feed into the TOE's RSA-

OEAP decryption operation some cipher texts where the first byte of EM was set to a value other than 0x00, and verify that the erroneous input is detected and that the decryption operation exits with an error code.

3. The evaluator shall decrypt each cipher text, C, produced by the RSA-OAEP encryption operation of the TOE using RSADP, and perform the OAEP decoding operation (described in NIST SP 800-56B section 7.2.2.4) to recover HA' || X. For each HA', the evalutor shall take the corresponding A and the specified hash algorithm and verify that HA' = Hash(A). The evaluator shall also force the TOE to perform some RSA-OAEP decryptions where the A value is passed incorrectly, and the evaluator shall verify that an error is detected.

4. The evaluator shall check the format of the 'X' string recovered in OAEP.Test.3 to ensure that the format is of the form PS || 01 || K, where PS consists of zero or more consecutive 0x00 bytes and K is the transported keying material. The evaluator shall also feed into the TOE's RSA-OEAP decryption operation some cipher texts for which the resulting 'X' strings do not have the correct format (i.e., the leftmost non-zero byte is not 0x01). These incorrectly formatted 'X' variables shall be detected by the RSA-OEAP decrypt function.

5. The evaluator shall trigger all detectable decryption errors and validate that the returned error codes are the same and that no information is given back to the sender on which type of error occurred. The evaluator shall also validate that no intermediate results from the TOE's receiver-side operations are revealed to the sender.

If the KTS-KEM-KWS scheme was selected, the evaluator shall perform the following tests:

1. The evaluator shall inspect each cipher text, C, produced by RSA-KEM-KWS encryption operation of the TOE and make sure the length (in bytes) of the cipher text, cLen, is greater than nLen (the length, in bytes, of the modulus of the RSA public key) and that cLen - nLen is consistent with the byte lengths supported by the key wrapping algorithm. The evaluator shall feed into the RSA-KEM-KWS decryption operation a cipher text of unsupported length and verify that an error is detected and that the decryption process stops.

2. The evaluator shall separate the cipher text, C, produced by the sender-side of the TOE into its C0 and C1 components and use the RSA decryption primitive to recover the secret value, Z, from C0. The evaluator shall check that the unsigned integer represented by Z is greater than 1 and less than n-1, where n is the modulus of the RSA public key. The evaluator shall construct examples where the cipher text is created with a secret value Z = 1 and make sure the RSA-KEM-KWS decryption process detects the error. Similarly, the evaluator shall construct examples where the cipher text is created with a secret value Z = n – 1 and make sure the RSA-KEM-KWS decryption process detects the error.

3. The evaluator shall attempt to successfully recover the secret value Z, derive the key wrapping key, KWK, and unwrap the KWA-cipher text following the RSAKEM-KWS decryption process given in NISP SP 800-56B section 7.2.3.4. If the key-wrapping algorithm is AES-CCM, the evaluator shall verify that the value of any (unwrapped) associated data, A, that was passed with the wrapped keying material is correct The evaluator shall feed into the TOE's RSA-KEM-KWS decryption operation examples of incorrect cipher text and verify that a decryption error is detected. If the key-wrapping algorithm is AES-CCM, the evaluator shall attempt at least one decryption where the wrong value of A is given to the RSA-KEM-KWS decryption operation and verify that a decryption error is detected. Similarly, if the key-wrapping algorithm is AES-

CCM, the evaluator shall attempt at least one decryption where the wrong nonce is given to the RSA-KEM-KWS decryption operation and verify that a decryption error is detected.

4. The evaluator shall trigger all detectable decryption errors and validate that the resulting error codes are the same and that no information is given back to the sender on which type of error occurred. The evaluator shall also validate that no intermediate results from the TOE's receiver-side operations (in particular, no Z values) are revealed to the sender.

Justification
See issue description.

TD0219: NIAP Endorsement of Errata for HCD PP v1.0

Publication Date
2017.07.07

Protection Profiles
PP_HCD_V1.0

Other References
—

Issue Description

This errata applies to the "Protection Profile for Hardcopy Devices version 1.0, dated September 10, 2015" (HCD PP 1.0) and intend to correct editorial errors mainly in relation to the SFR definition.

Resolution

NIAP has endorsed the Errata for the Hard Copy Device Protection Profile v1.0 (HCD PP v1.0).

The ST author shall refer to this errata after applying the contents of the HCD PP v1.0 within the ST.

Justification
See Resolution.

TD0176: FDP_DSK_EXT.1.2 - SED Testing

Publication Date
2017.04.11

Protection Profiles
PP_HCD_V1.0

Other References
FDP_DSK_EXT.1.2

Issue Description

The FDP_DSK_EXT.1.2 test assurance activity within the HCD PPv1.0 may be impractical for testing self-encrypting drives (SEDs). The SEDs are required by HCD PPv1.0 to be separately CC certified to conform to the FDE EE cPP.

Resolution

FDP_DSK_EXT.1.2 The TSF shall encrypt all protected data without user intervention.

Application Note:

The intent of this requirement is to specify that encryption of any confidential data will not depend on a user electing to protect that data. The encryption specified in FDP_DSK_EXT.1 occurs transparently to the user and the decision to protect the data is outside the discretion of the user.

If a vendor makes the selection "use a self-encrypting Field-Replaceable Nonvolatile Storage Device that is separately CC certified to conform to the FDE EE cPP" in FDP_DSK_EXT.1.1 , testing is not required as SED testing is performed within the FDE EE cPP already.

The TSS, KMD, and test sections only apply to parts of the TOE which fall under the selection "perform encryption in accordance with FCS_COP.1(d)".

TSS:

If the self-encrypting device option is selected, the device must be certified in conformance to the current Full Disk Encryption Protection Profile. The tester shall confirm that the specific SED is listed in the TSS, documented and verified to be CC certified against the FDE EE cPP.

The evaluator shall examine the TSS to ensure that the description is comprehensive in how the data is written to the Device and the point at which the encryption function is applied.

For the cryptographic functions that are provided by the Operational Environment, the evaluator shall check the TSS to ensure it describes the interface(s) used by the TOE to invoke this functionality.

The evaluator shall verify that the TSS describes the initialization of the Device at shipment of the TOE, or by the activities the TOE performs to ensure that it encrypts all the storage devices entirely when a user or administrator first provisions the Device. The evaluator shall verify the TSS describes areas of the Device that it does not encrypt (e.g., portions that do not contain confidential data boot loaders, partition tables, etc.). If the TOE supports multiple Device encryptions, the evaluator shall examine the administration guidance to ensure the initialization procedure encrypts all Devices.

Justification
The SEDs are required by HCD PPv1.0 to be separately CC certified to conform to the FDE EE cPP.

TD0157: FCS_IPSEC_EXT.1.1 - Testing SPDs

Publication Date
2017.06.15

Protection Profiles
PP_HCD_V1.0

Other References
FCS_IPSEC_EXT.1.1

Issue Description

Some HCDs do not permit administrators to manually configure or edit the IPsec SPD, nor are BYPASS operations supported. The SPD is automatically configured from the configured list of systems authorized to communicate with an HCD. All IP traffic is required to use IPsec. Therefore, no BYPASS operations are supported.

Resolution

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

Application Note:

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.

Assurance Activity:

TSS:

The evaluator shall examine the TSS and determine that it describes what takes place when a packet is processed by the TOE, e.g., the algorithm used to process the packet. The TSS describes how the SPD is implemented and the rules for processing both inbound and outbound packets in terms of the IPsec policy. The TSS describes the rules that are available and the resulting actions available after matching a rule. The TSS describes how those rules and actions form the SPD in terms of the BYPASS (e.g., no encryption), DISCARD (e.g., drop the packet) and PROTECT (e.g., encrypt the packet) actions defined in RFC 4301.

As noted in section 4.4.1 of RFC 4301, the processing of entries in the SPD is non-trivial and the evaluator shall determine that the description in the TSS is sufficient to determine which rules will be applied given the rule structure implemented by the TOE. For example, if the TOE allows specification of ranges, conditional rules, etc., the evaluator shall determine that the description of rule processing (for both inbound and outbound packets) is sufficient to determine the action that will be applied, especially in the case where two different rules may apply. This description shall cover both the initial packets (that is, no SA is established on the interface or for that particular packet) as well as packets that are part of an established SA.

Operational Guidance:

The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for processing a packet. The description includes all three cases – a rule that ensures packets are encrypted/decrypted, dropped, and flow through the TOE without being encrypted. The evaluator shall determine that the description in the guidance documentation is consistent with the description in the TSS, and that the level of detail in the guidance documentation is sufficient to allow the administrator to set up the SPD in an unambiguous fashion. This includes a discussion of how ordering of rules impacts the processing of an IP packet.

Test:

The evaluator uses the guidance documentation to configure the TOE to carry out the following tests:

a) Test 1: The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and (if configurable) allowing a packet to flow in plaintext. The selectors used in the construction of the rule shall be different such that the evaluator can generate a packet and send packets to the gateway with the appropriate fields (fields that are used by the rule - e.g., the IP addresses, TCP/UDP ports) in the packet header. The evaluator performs both positive and negative test cases for each type of rule (e.g. a packet that matches the rule and another that does not match the rule). The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packets were dropped, allowed to flow without modification, encrypted by the IPsec implementation.

b) Test 2: The evaluator shall devise several tests that cover a variety of scenarios for packet processing. As with Test 1, the evaluator ensures both positive and negative test cases are constructed. These scenarios must exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and guidance documentation. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the guidance documentation.

Justification
Aligned FCS_IPSEC_EXT.1.1 with the NDcPP v1.0; removed BYPASS.

TD0074: FCS_CKM.1(a) Requirement in HCD PP v1.0

Publication Date
2015.12.15

Protection Profiles
PP_HCD_V1.0

Other References
—

Issue Description

The Security Functional Requirement for FCS_CKM.1(a) should be considered an optional requirement in the HCD PP v1.0. The SFR and associated Tests in the Assurance Activity are being moved to "Appendix C Optional Requirements."

Pages 38 to 40, Section 4.5.1 FCS_CKM.1(a) currently reads:

Section 4.5.1 FCS_CKM.1(a) Cryptographic Key Generation (for asymmetric keys)

(for O.COMMS_PROTECTION)

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key distribution, or

FCS_COP.1(b) Cryptographic Operation (for signature generation/ verification)]

FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

FCS_CKM.1.1(a) Refinement: The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance **with [selection:**

• ***NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;***

• ***NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [selection: P-521, no other curves] (as defined in FIPS PUB 186-4, "Digital Signature Standard")***

• ***NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes***

¶ *190* **] and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.**

¶ *191* ***Application Note:***

¶ *192 The ST author selects the key generation scheme used for key establishment and device authentication. If multiple schemes are supported, then the ST author should iterate this component to capture this capability. When key generation is used for device authentication, the public key is expected to be associated with an X.509v3 certificate. If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.*

¶ *193 Since the domain parameters to be used are specified by the requirements of the protocol in this PP, it is not expected that the TOE will generate domain parameters, and therefore there is no additional domain parameter validation needed when the TOE complies with the protocols specified in this PP.*

¶ *194 SP 800-56B references (but does not mandate) key generation according to FIPS 186-3. For purposes of compliance in this version of the HCD PP, RSA key pair generation according to FIPS 186-4 is allowed in order for the TOE to claim conformance to SP 800-56B.*

¶ *195 The generated key strength of 2048-bit DSA and rDSA keys need to be equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.*

196 **Assurance Activity:**

¶ *197* ***TSS:***

¶ *198* The evaluator shall ensure that the TSS contains a description of how the TSF complies with 800-56A and/or 800-56B, depending on the selections made. This description shall indicate the sections in 800-56A and/or 800-56B that are implemented by the TSF, and the evaluator shall ensure that key establishment is among those sections that the TSF claims to implement.

¶ *199* Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described in the TSS.

¶ *200* The TSS may refer to the Key Management Description (KMD), described in Appendix F , that may not be made available to the public.

¶ *201* ***Test:***

¶ *202* The evaluator shall use the key pair generation portions of "The FIPS 186-4 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-4 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and "The 186-4 RSA Validation System (RSA2VS)" as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

Resolution

Section 4.5.1 FCS_CKM.1(a) referenced above is being moved to “Appendix C Optional Requirements.”

Justification
FCS_CKM.1(a) SFR and Assurance Activity is optional requirement in HCD PP v1.0.

III. Protection Profile for Hardcopy Devices – v1.0 Errata #1, June 2017, Section 3.1 Changes

3.1.1 Class FCS: Cryptographic Support

FCS_COP.1(b) Cryptographic Operation (for signature generation/verification)

FCS_COP.1.1(b) Refinement: The TSF shall perform **cryptographic signature services** in accordance with a [selection:

- ***Digital Signature Algorithm (DSA) with key sizes (modulus) of [assignment: 2048 bits or greater],***
- ***RSA Digital Signature Algorithm (rDSA) with key sizes (modulus) of [assignment: 2048 bits or greater], or***
- ***Elliptic Curve Digital Signature Algorithm (ECDSA) with key sizes of [assignment: 256 bits or greater]]***

that meets the following **[selection:**

~~Case: Digital Signature Algorithm~~

- ~~FIPS PUB 186-4, "Digital Signature Standard"~~

~~Case: RSA Digital Signature Algorithm~~

- ~~FIPS PUB 186-4, "Digital Signature Standard"~~

~~Case: Elliptic Curve Digital Signature Algorithm~~

- ~~FIPS PUB 186-4, "Digital Signature Standard"~~
- ~~The TSF shall implement "NIST curves" P-256, P384 and [selection: P521, no other curves] (as defined in FIPS PUB 186-4, "Digital Signature Standard").~~

Case: Digital Signature Algorithm

- ***FIPS PUB 186-4, "Digital Signature Standard"***

Case: RSA Digital Signature Algorithm

- ***FIPS PUB 186-4, "Digital Signature Standard"***

Case: Elliptic Curve Digital Signature Algorithm

- ***FIPS PUB 186-4, "Digital Signature Standard"***

- ***The TSF shall implement "NIST curves" P-256, P384 and [selection: P521, no other curves] (as defined in FIPS PUB 186-4, "Digital Signature Standard").***

].

FCS_COP.1(g) Cryptographic Operation (for keyed-hash message authentication)

FCS_COP.1.1(g) Refinement: The TSF shall perform ~~keyed-hash message authentication~~ **keyed-hash message authentication** in accordance with a specified cryptographic algorithm ~~HMAC-~~**HMAC-**[selection: ~~*SHA-1, SHA-224, SHA-256, SHA-384, SHA-512*~~***SHA-1, SHA-224, SHA-256, SHA-384, SHA-512***]**, key size** [assignment: ~~*key size (in bits) used in HMAC*~~ **key size (in bits) used in HMAC**], ~~and message digest sizes [selection: *160, 224, 256, 384, 512*] bits~~ **and message digest sizes [selection: *160, 224, 256, 384, 512*] bits** that meet the following: ~~FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code, and FIPS PUB 180-3, "Secure Hash Standard."~~**FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code, and FIPS PUB 180-3, "Secure Hash Standard."**

FCS_COP.1(h) Cryptographic Operation (for keyed-hash message authentication)

FCS_COP.1.1(h) Refinement: The TSF shall perform [**keyed-hash message authentication**] in accordance with [**selection:** ~~*HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512*~~***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512***] and cryptographic key sizes [assignment: ~~*key size (in bits) used in HMAC*~~ ***key size (in bits) used in HMAC***] that meet the following: [**ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"; ISO/IEC 10118**].

FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

FCS_CKM_EXT.4.1 The TSF shall destroy ~~**all plaintext secret and private cryptographic keys and cryptographic critical security parameters**~~ all plaintext secret and private cryptographic keys and cryptographic critical security parameters when no longer needed.

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 Refinement: The TSF shall ~~**destroy**~~ destroy cryptographic keys in accordance with a specified cryptographic key ~~**destruction**~~ destruction method [~~selection~~ **selection**:

~~**For volatile memory, the destruction shall be executed by [selection:** ***powering off a device*****, [assignment:** ***other mechanism that ensures keys are destroyed*****]].**~~

~~**For nonvolatile storage, the destruction shall be executed by a [selection:** ***single, three or more times*****] overwrite of key data storage location consisting of [selection:** ***a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1), a static pattern*****], followed by a [selection:** ***read-verify, none*****]. If read-verification of the overwritten data fails, the process shall be repeated again;**~~

For volatile memory, the destruction shall be executed by [selection: powering off a device, [assignment: other mechanism that ensures keys are destroyed]].

For nonvolatile storage, the destruction shall be executed by a [selection: single, three or more times] overwrite of key data storage location consisting of [selection: a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1), a static pattern], followed by a [selection: read-verify, none]. If read-verification of the overwritten data fails, the process shall be repeated again;] that meets the following: [**selection:** ***NIST SP800-88, no standard***].

FCS_SNI_EXT.1 Extended: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)

FCS_SNI_EXT.1.1 The TSF shall only use salts that are generated by a ~~**RNG as specified in FCS_RBG_EXT.1**~~ RNG as specified in FCS_RBG_EXT.1.

FCS_SNI_EXT.1.2 The TSF shall only use unique nonces with a minimum size of [64] bits. **FCS_SNI_EXT.1.3** The TSF shall create IVs in the following manner: [

- CBC: IVs shall be non-repeating,
- CCM: Nonce shall be non-repeating.
- XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer,
- GCM: IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key.

].

3.1.2 Class FAU: Security Audit

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 shutdown of the audit functions;
b) All auditable events for the **not specified** level of audit; and
c) ~~All auditable events specified in Table 1,~~ **~~All auditable events specified in Table 1,~~** ~~[assignment: other specifically defined auditable events].~~

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 (if applicable), 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 PP/ST, **additional information specified in Table 1,** [assignment: other audit relevant information].

Table 1 Auditable Events

Auditable event	Relevant SFR	Additional information
Job completion	FDP_ACF.1	Type of job
Unsuccessful User authentication	FIA_UAU.1	None
Unsuccessful User identification	FIA_UID.1	None
Use of management functions	FMT_SMF.1	None
Modification to the group of Users that are part of a role	FMT_SMR.1	None

Changes to the time	FPT_STM.1	None
Failure to establish session	FTP_ITC.1, FTP_TRP.1(a), FTP_TRP.1(b)	Reason for failure

3.1.3 Class FMT: Security Management

The portions of an SFR that has been completed in this protection profile are required to be in **Bold** typeface. The Authorized roles and Data in Table 4 should be in **Bold** typeface as following:

FMT_MTD.1 Management of TSF data

FMT_MTD.1.1 Refinement: The TSF shall restrict the ability to **perform the specified operations on the specified TSF Data to the roles specified in Table 4.**

Table 4 Management of TSF Data

Data	Operation	Authorised role(s)
[assignment: *list of TSF Data* ~~*owned by a U.NORMAL or associated with Documents or jobs owned by a U.NORMAL*~~ ***owned by a U.NORMAL or associated with Documents or jobs owned by a U.NORMAL***]	[selection: *change default, query, modify, delete, clear*, [assignment: *other operations*]]	~~U.ADMIN, the owning U.NORMAL.~~ **U.ADMIN, the owning U.NORMAL.**
[assignment: *list of TSF Data* ~~*not owned by a U.NORMAL*~~ ***not owned by a U.NORMAL***]	[selection: *change default, query, modify, delete, clear*, [assignment: *other operations*]]	~~U.ADMIN~~ **U.ADMIN**
[assignment: ***list of software, [redacted] firmware, and related configuration data***]	[selection: *change default, query, modify, delete, clear*, [assignment: *other operations*]]	~~U.ADMIN~~ **U.ADMIN [redacted]**

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 ~~Refinement:~~ The TSF shall be capable of performing the following management functions: [assignment: *list of management functions provided by the TSF*].

3.1.4 Class FPT: Protection of the TSF

FPT_KYP_EXT.1 Extended: Protection of Key and Key Material
(for O.KEY_MATERIAL)

FPT_KYP_EXT.1.1 Refinement: The TSF shall not store plaintext keys that are part of the keychain specified by FCS_KYC_EXT.1 in **any Field-Replaceable Nonvolatile Storage Device**.

3.1.5 Class FTP: Trusted Path/Channels

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 Refinement: The TSF shall **use [selection: *IPsec, SSH, TLS, TLS/HTTPS*] to** provide **a trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: [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 **Refinement:** The TSF shall permit **the TSF, or the authorized IT entities,** to initiate communication via the trusted channel

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

FTP_TRP.1(a) Trusted path (for Administrators)

FTP_TRP.1.1(a) Refinement: The TSF shall **use [selection, choose at least one of: IPsec, SSH, TLS, TLS/HTTPS] to** provide **a trusted** communication path between itself and **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 detection of modification of the communicated data.**

FTP_TRP.1.2(a) Refinement: The TSF shall permit **remote administrators** to initiate communication via the trusted path

FTP_TRP.1.3(a) Refinement: The TSF shall require the use of the trusted path for **initial administrator authentication and all remote administration actions.**

FTP_TRP.1(b) Trusted path (for Non-administrators)

FTP_TRP.1.1(b) Refinement : The TSF shall **use [selection, *choose at least one of: IPsec, SSH, TLS, TLS/HTTPS*] to** provide **a trusted** communication path between itself and **remote** users 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 detection of modification of the communicated data.**

FTP_TRP.1.2(b) Refinement: The TSF shall permit [selection: ***the TSF, remote users***] to initiate communication via the trusted path

FTP_TRP.1.3(b) Refinement: The TSF shall require the use of the trusted path for **initial user authentication and all remote user actions**.

NOTE: There were additional changes to the Extended Component Definitions, missing definitions of terms and updates to the SFR dependencies that were not listed above. These are included in the full E a document.


pp_hcd_v1.0-err.doc pp_hcd_v1.0-err.pdf

pp_hcd_v1.0-err.pdf


pp_hcd_v1.0-err.pdf pp_hcd_v1.0-err.pdf