Security Requirements for Network Devices

Errata #3

3 November 2014

1 APPLICABILITY

These errata apply to the *Security Requirements for Network Devices* Protection Profile, version 1.1, dated 8 June 2012 (NDPP). It contains a summary of the changes/clarifications to the NDPP, followed by replacement text. It is anticipated that the errata will be a dynamic document updated as needed to clarify or correct the NDPP. ST authors using these errata will indicate this use in the "Conformance" section of the ST.

These errata detail changes that correct errors in version 1.1 of the PP (typographical errors, unclear or incorrectly specified requirements or assurance activities); add options to or remove mandatory portions of existing requirements; add minor to moderate assurance activities as clarifications of expected activity, align requirements with current policy.

These errata amend version 1.0 of the previous NDPP errata (Errata #1, dated 19 December 2013) to clarify the "exact compliance" statement; remove the restrictions on the use of TLS; remove 112 bits as a selection in the FCS_RBG_EXT.1 requirements; and make it clear that pre-shared keys are mandatory if IPsec is included in the evaluated configuration (it was incorrectly made optional in the first version of the errata); no other changes (other than typo fixes), additions, or deletions have been made in this version of the errata compared with the previous version of the errata.

The TLS requirement is being re-worked and a more stringent version that complies with CNSSP 15 and NIST SP 800-131A will be included in the next version of the NDPP.

2 SUMMARY OF CHANGES

The changes outlined in these errata address communications with an (optional) NTP server, integrity functions used by cryptographic protocols specified in the PP, ephemeral key pair generation, removal of unused options for symmetric encryption, and entropy requirements for the Random Bit Generator.

These changes are summarized below, with ~~strikethroughs~~ indicating removal and underlines indicating addition for the NDPP appearing in the following sections.

2.1 Near-term Changes

In version 1.1, it is unclear that "exact compliance" is mandated for compliant TOEs. *Exact Compliance* is defined as the ST containing all of the requirements in section 4 of the NDPP, and appropriate requirements from Appendix C of the NDPP. While iteration is allowed, no additional requirements (from the CC parts 2 or 3) are allowed to be included in the ST. Further, no requirements in section 4 of the NDPP are allowed to be omitted.

In version 1.1 of the NDPP, it is not required that the TOE communicate with an external NTP server (should one be supported by the TOE) using a cryptographic protocol. In earlier versions of the NDPP this was required, and references to this capability were incompletely removed in version 1.1.

Version 1.1 of the NDPP is imprecise in requiring the specification of integrity functions to be used in the IPSEC and SSH protocols, and the assurance activities aimed at verifying the integrity functions that are used by the Target of Evaluation (TOE) in the operation of trusted channels and trusted path. In order to clarify what is required in both the Security Target (ST) and the associated evaluator assurance activities, the assurance activities for IPSEC (FCS_IPSEC_EXT.1), SSH (FCS_SSH_EXT.1), trusted channels (FTP_ITC.1), and trusted path (FTP_TRP.1) have been modified. The general notion is that the ST (in the TOE Summary Specification (TSS)) the integrity algorithms actually used by the TOE are identified, and then the evaluator confirms that these algorithms are actually capable of being used through testing. This can be done by observing the protocol negotiation of the algorithms through packet captures to ensure the allowed algorithms are agreed upon; no cryptoanalysis of the encrypted traffic to try to determine how it's encrypted is necessary. In cases where this negotiation may take place inside an encrypted tunnel (so it is not visible in the packet captures), alternative means are usually available (for example, TOE-based administrative commands showing the characteristics (including encryption and integrity algorithms being used) of particular communication endpoints).

The requirements for the IPSEC protocol implementation for use in PPs has been refined; refinements which are minor in nature are reflected in the errata through revised elements, application notes, and assurance activities in the FCS_IPSEC_EXP requirement, as well as the addition of the FIA_PSK_EXT.1 requirement for pre-shared key composition.

In Version 1.1 of the NDPP there were four mandatory ciphersuites specified for TLS implementations. In light of the current state of TLS ciphersuites that are actually implemented, a "minimum bar" (TLS_RSA_WITH_AES_128_CBC_SHA) is sufficient to comply with the requirement, while the other (formerly mandatory) ciphersuites have been made optional. As noted above, the TLS requirement will be modified to support the requirements of CNSSP 15 and NIST SP 800-131A in the next version of the NDPP. Implementers are encouraged to include ciphersuites and underlying operations that meet this policy and SP in the interim.

Newer RFCs have been published with respect to the SSH protocol that allow a greater range of cryptographic choices; these RFCs and cryptographic elements have been added as selections for the FCS_SSH_EXT.1 component.

Version 1.1 of the NDPP requires that key pair generation (used for certain cryptographic protocols—such as IPsec—in negotiations to set up an encrypted communication channel) conform either to NIST Special Pub 800-56A or 800-56B, depending on the type of keys being generated. FCS_COP.1(2) allows RSA implementations to comply with 186-2 or 186-3 (although 186-2 will be phased out in the next full release of the NDPP). While SP 800-56B does not required a FIPS-validated implementation of the RSA Key Generation algorithm for compliance, it does provide a reference to only 186-3 (not 186-2) for the Key Generation specifics. Further, the assurance requirement for the FCS_CKM.1 requirement references the 186-3 RSA Validation System and not the 186-2 RSA Validation System. For the purposes of determining compliance to 800-56B, the evaluators will use the appropriate RSA Validation System (to correspond to the implementation in the TOE, which may be either 186-2 or 186-3 validated) for this portion of the requirement. This requirement also contained assurance activities related to the specification of optional components of SP 800-56A/56B. This language was changed to align the assurance activities with similar requirements/activities for RFC-based mechanisms.

Version 1.1 of the NDPP requires digital signature operations be performed according to 186-2 (for rDSA) or 186-3 (for any of the three schemes). The assurance activity, however, does not provide the correct verification references for the selections available in the requirement; the assurance activity is adjusted to match the mandates of the requirement.

Version 1.1 of the NDPP allows ST authors to choose 192-bit AES symmetric encryption. However, none of the functions required by the NDPP use 192-bit symmetric encryption, making this an option that should never to chosen. To avoid confusion on the part of ST authors, this option has been removed.

The Random Bit Generation requirement in version 1.1 (FCS_RBG_EXT.1) specifies the minimum bits of entropy required to seed the RBG. Given that random numbers are used for algorithms such as SHA and HMAC (with 512-bit keys) and public key generation with 2048-bit moduli, the wording of the requirement is awkward. The intent is that the bits of entropy correspond with the *security strength* of the greatest key-length for the algorithms implemented in the TOE. Security strength is defined in NIST SP 800-57A (tables 2 and 3), and for the NDPP no algorithm that is able to be selected has a security strength greater than 256 bits; therefore, 256 bits should be the maximum amount of entropy that is required in order to seed the RBG. Additionally, a new, 3-part version of 800-90 has been released, so references to that Special Publication have been updated.

3 DETAILED ERRATA

The following changes are to be made to the requirements in the NDPP. Actual changed text is marked with a change bar; for convenience, the entire component, application notes, and assurance activity is included, even portions that have not changed.

3.1 Protection of the TSF (FPT)

The following changes are to the *Compliant Target of Evaluations* section on page 1 of the NDPP.

It is intended that the set of requirements in this PP is limited in scope in order to promote quicker, less costly evaluations that provide some value to end users. ~~STs that include a large amount of additional functionality (and requirements) are discouraged. Future modules~~ Extended Packages will be used to specify sets of additional functionality (e.g., Firewalls, VPNs), which can then be used by ST writers looking to specify additional functionality. In order to be conformant to this PP, a TOE must demonstrate *Exact Compliance*. *Exact Compliance*, as subset of *Strict Compliance* as defined by the CC, is defined as the ST containing all of the requirements in section 4 of the NDPP, and potentially requirements from Appendix C of the NDPP. While iteration is allowed, no additional requirements (from the CC parts 2 or 3) are allowed to be included in the ST. Further, no requirements in section 4 of the NDPP are allowed to be omitted.

3.2 Protection of the TSF (FPT)

The following changes are to the FPT_STM component on page 27 of the NDPP.

FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

Assurance Activity:

The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time. The TSS provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

The evaluator examines the operational guidance to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the operational guidance instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

- *Test 1: The evaluator uses the operational guide to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.*

- *Test2: [conditional] If the TOE supports the use of an NTP server; the evaluator shall use the operational guidance to configure the NTP client on the TOE, and set up a communication path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple ~~cryptographic~~ protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the operational guidance.*

3.3 Trusted Path/Channels (FTP)

The following changes are to the FTP_ITC and FTP_TRP components on pages 30-33 of the NDPP.

Trusted Channel (FTP_ITC)

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: audit server, [selection: authentication server, assignment: [other capabilities]]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data**.

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

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

Application Note:

The intent of the above requirement is to use a cryptographic protocol to protect external communications with authorized IT entities that the TOE interacts with to perform its functions. This is not, however, to be used to specify VPN Gateway functionality; a separate VPN Protection Profile should be used in these instances. Protection (by one of the listed protocols) is required at least for communications with the server that collects the audit information. If it communicates with an authentication server (e.g., RADIUS), then the ST author chooses "authentication server" in FTP_ITC.1.1 and this connection must be protected by one of the listed protocols. If other authorized IT entities (e.g., NTP server) are protected, the ST author makes the appropriate assignments (for those entities) and selections (for the protocols that are used to protect those connections). After the ST author has made the selections, they are to select the detailed requirements in Annex C corresponding to their protocol selection to put in the ST. To summarize, the connection to an external audit collection server is required to be protected by one of the listed protocols. If an external authentication server is supported, then it is required to protect that connection with one of the listed protocols. For any other external server, external communications are not required to be protected, but if protection is claimed, then it must be protected with one of the identified protocols.

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

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

Assurance Activity:

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken. The evaluator shall also perform the following tests:

- *Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*
- *Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE.*
- *Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data ~~is~~ are not sent in plaintext.*
- ~~*Test 4: The evaluator shall ensure, for each communication channel with an authorized IT entity, modification of the channel data is detected by the TOE.*~~
- *Test 4 ~~5~~: The evaluators shall, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.*

Further assurance activities are associated with the specific protocols.

Trusted Path (FTP_TRP)

FTP_TRP.1 **Trusted Path**

FTP_TRP.1.1 **Refinement:** The TSF shall **use [selection, choose at least one of: IPsec, SSH, TLS, TLS/HTTPS]** 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 **Refinement:** The TSF shall permit **remote administrators** to initiate communication via the trusted path.

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

Application Note:

This requirement ensures that authorized remote administrators initiate all communication with the TOE via a trusted path, and that all communications with the TOE by remote administrators is performed over this path. The data passed in this trusted communication channel are encrypted as defined the protocol chosen in the first selection. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Annex C corresponding to their selection are copied to the ST if not already present.

Assurance Activity:

The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method. The evaluator shall also perform the following tests:

- *Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.*
- *Test 2: For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative sessions without invoking the trusted path.*
- *Test 3: The evaluator shall ensure, for each method of remote administration, the channel data ~~is~~ are not sent in plaintext.*
- ~~*Test 4: The evaluator shall ensure, for each method of remote administration, modification of the channel data is detected by the TOE.*~~

Further assurance activities are associated with the specific protocols.

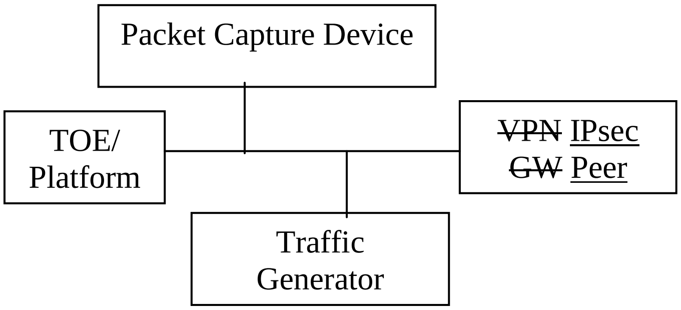
3.4 IPSEC Protocol

The following changes are to the FCS_IPSEC_EXP component on pages 48-52 of the NDPP.

FCS_IPSEC_EXT.1 Explicit: IPSEC

In order to show that the TSF implements the RFCs in accordance with the requirements of this PP, the evaluator shall perform the assurance activities listed below. In future versions of this PP, assurance activities may be augmented, or new ones introduced that cover more aspects of RFC compliance than are currently described in this publication.

The TOE is required to use the IPsec protocol to establish connections used to communicate with a IPsec Peer.



The evaluators shall minimally create a test environment equivalent to the test environment illustrated above. It is expected that the traffic generator is used to construct network packets and will provide the evaluator with the ability manipulate fields in the ICMP, IPv4, IPv6, UDP, and TCP packet headers. The evaluators must provide justification for any differences in the test environment.

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

Assurance Activity:

The evaluator shall examine the operational guidance to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for DISCARD, BYPASS and PROTECT.
The evaluator uses the operational guidance to configure the TOE and platform to carry out the following tests:

- *Test 1: The evaluator shall configure the SPD such that there is a rule for DISCARD, BYPASS, PROTECT. The selectors used in the construction of the rule shall be different such that the evaluator can send in three network packets with the appropriate fields in the packet header that each packet will match one of the three rules. The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packet was dropped, allowed through without modification, was encrypted by the IPsec implementation.*
- *Test 2: The evaluator shall devise two equal SPD entries with alternate operations – BYPASS and PROTECT. The entries should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first entry is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation.*

- *Test 3: The evaluator shall repeat the procedure above, except that the two entries should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.*

FCS_IPSEC_EXT.1.2 The TSF shall implement [selection: tunnel mode, transport mode].

Assurance Activity:

The evaluator checks the TSS to ensure it states that the VPN can be established to operate in tunnel mode and/or transport mode (as selected). The evaluator shall confirm that the operational guidance contains instructions on how to configure the connection in each mode selected.

The evaluator shall perform the following test(s) based on the selections chosen:

- *Test 1 (conditional): If tunnel mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in tunnel mode and also configures a IPsec Peer to operate in tunnel mode. The evaluator configures the TOE and the IPsec Peer to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the client to connect to the IPsec Peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.*

- *Test 2 (conditional): If transport mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in transport mode and also configures a IPsec Peer to operate in transport mode. The evaluator configures the TOE and the IPsec Peer to use any of the allowed cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator then initiates a connection from the TOE to connect to the IPsec Peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.*

FCS_IPSEC_EXT.1.3 The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

Assurance Activity:

The evaluator shall examine the TSS to verify that the TSS provides a description of how a packet is processed against the SPD and that if no “rules” are found to match, that a final rule exists, either implicitly or explicitly, that causes the network packet to be discarded.

The evaluator checks that the operational guidance provides instructions on how to construct the SPD and uses the guidance to configure the TOE for the following tests.

The evaluator shall perform the following test:

- *Test 1: The evaluator shall configure the SPD such that it has entries that contain operations that DISCARD, BYPASS, and PROTECT network packets. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches a BYPASS entry and send that packet. The evaluator should observe that the network packet is passed to the proper destination interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator-created entries (there may be a "TOE created" final entry that discards packets that do not match any previous entries). The evaluator sends the packet, and observes that the packet was not permitted to flow to any of the TOE's interfaces.*

FCS_IPSEC_EXT.1.4 ~~1~~ The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using [**selection:** the cryptographic algorithms AES-CBC-128 (as specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, AES-CBC-256 (as specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, ~~(both specified by RFC 3602), [selection: no other algorithms, AES-GCM-128~~ as specified in RFC 4106, AES-GCM-256 as specified in RFC 4106], ~~and using [selection, choose at least one of: IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]; IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), 4307, and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]].~~

~~*Application Note: The first selection is used to identify additional cryptographic algorithms supported. Either IKEv1 or IKEv2 support must be provided, although conformant TOES can provide both; the second selection is used to make this choice. For IKEv1, the requirement is to be interpreted as requiring the IKE implementation conforming to RFC 2409 with the additions/modifications as described in RFC 4109. RFC 4868 identifies additional hash functions for use with both IKEv1 and IKEv2; if these functions are implemented, the third (for IKEv1) and fourth (for IKEv2) selection can be used. IKEv2 will be required after January 1st, 2014.*~~

Assurance Activity:

The evaluator shall examine the TSS to verify that the symmetric encryption algorithms selected (along with the SHA-based HMAC algorithm, if AES-CBC is selected) are describedl. If selected, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in FCS_COP.1(4) Cryptographic Operations (for keyed-hash message authentication.

The evaluator checks the operational guidance to ensure it provides instructions on how to configure the TOE to use the algorithms selected by the ST author. The evaluator shall also perform the following tests:

- Test 1: The evaluator shall configure the TOE as indicated in the operational guidance configuring the TOE to using each of the selected algorithms, and attempt to establish a connection using ESP. The connection should be successfully established for each algorithm.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [selection: IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, [selection: no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]; IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), 4307, and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]].

Application Note: Either IKEv1 or IKEv2 support must be provided, although conformant TOEs can provide both; the first selection is used to make this choice. For IKEv1, the requirement is to be interpreted as requiring the IKE implementation conforming to RFC 2409 with the additions/modifications as described in RFC 4109. RFC 4304 identifies support for extended sequence numbers, which compliant TOEs can specify using the second selection. RFC 4868 identifies additional hash functions for use with both IKEv1 and IKEv2; if these functions are implemented, the third (for IKEv1) and fourth (for IKEv2) selection can be used.

Assurance Activity:

The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.

The evaluator shall check the operational guidance to ensure it instructs the administrator how to configure the TOE to use IKEv1 and/or IKEv2 (as selected), and uses the guidance to configure the TOE to perform NAT traversal for the following test if IKEv2 is selected.

Test 1 (conditional): The evaluator shall configure the TOE/platform so that it will perform NAT traversal processing as described in the TSS and RFC 5996, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [selection: IKEv1, IKEv2] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and [selection: AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm].

Assurance Activity:

The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/or IKEv2 payload, and that the algorithms AES-CBC-128, AES-CBC-256 are specified, and if others are chosen in the selection of the requirement, those are included in the TSS discussion.

The evaluator ensures that the operational guidance describes the configuration of the mandated algorithms, as well as any additional algorithms selected in the requirement. The guidance is then used to configure the TOE to perform the following test for each ciphersuite selected.

Test 1: The evaluator shall configure the TOE to use the ciphersuite under test to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using the indicated ciphersuite. The evaluator will confirm the algorithm was that used in the negotiation.

FCS_IPSEC_EXT.1.7 ~~2~~ The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

Assurance Activity:
~~*The evaluator shall examine the TSS to verify that it describes how "confidentiality only" ESP mode is disabled. The evaluator shall also examine the operational guidance to determine that it describes any configuration necessary to ensure that "confidentiality only" mode is disabled, and that an advisory is present indicating that tunnel mode is the preferred ESP mode since it protects the entire packet.*~~

The evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol supported by the TOE, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. It may be that this is a configurable option.

If ~~this~~ the mode requires configuration of the TOE prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance. The evaluator shall also perform the following tests:

- Test 1 (conditional): The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported. This test is not applicable if IKEv1 is not selected above in the FCS_IPSEC_EXT.1.5 protocol selection.

- ~~Test 2: The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using ESP in "confidentiality only" mode. This attempt should fail. The evaluator shall then establish a connection using ESP in confidentiality and integrity mode.~~

FCS_IPSEC_EXT.1.8 ~~3~~ The TSF shall ensure that [selection: IKEv2 SA lifetimes can be established based on [selection: number of packets/number of bytes; length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs]; IKEv1 SA lifetimes can be established based on [selection: number of packets/number of bytes ; length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs]]. ~~IKEv1 SA lifetimes are able to be limited to 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs.~~

Application Note: *The ST Author is afforded a selection based on the version of IKE in their implementation. If the lifetime limitations are configurable, then the evaluator verifies that the appropriate instructions for configuring these values are included in the operational guidance.*

As far as SA lifetimes are concerned, the TOE can limit the lifetime based on the number of bytes transmitted, or the number of packets transmitted. Either packet-based or volume-based SA lifetimes are acceptable; the ST author makes the appropriate selection to indicate which type of lifetime limits are supported ~~The above requirement can be accomplished either by providing Security Administrator-configurable lifetimes (with appropriate FMT requirements and instructions~~

~~*in documents mandated by AGD_OPE, as necessary), or by "hard coding" the limits in the implementation.*~~

Assurance Activity:

The evaluator verifies that the values for SA lifetimes can be configured and that the instructions for doing so are located in the operational guidance. If time-based limits are supported, the evaluator ensures that the values allow for Phase 1 SAs values for 24 hours and 8 hours for Phase 2 SAs. Currently there are no values mandated for the number of packets or number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

When testing this functionality, the evaluator needs to ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection ~~*The evaluator checks to ensure that the TSS describes how lifetimes for IKEv1 SAs (both Phase 1 and Phase 2) are established. If they are configurable, then the evaluator verifies that the appropriate instructions for configuring these values are included in the operational guidance. The evaluator also performs the following test:*~~

- Test 1 (Conditional): *The evaluator shall configure a maximum lifetime in terms of the # of packets (or bytes) allowed following the operational guidance. The evaluator shall establish an SA and determine that once the allowed # of packets (or bytes) through this SA is exceeded, the connection is closed*.

- Test 2 ~~1~~ (Conditional): The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.

- Test 3 ~~2~~ (Conditional): The evaluator shall perform a test similar to Test 1 for Phase 2 SAs, except that the lifetime will be 8 hours instead of 24.

~~**FCS_IPSEC_EXT.1.4** The TSF shall ensure that IKEv1 SA lifetimes are able to be limited to [assignment: *number between 100 - 200*] MB of traffic for Phase 2 SAs.~~

~~*Application Note: The above requirement can be accomplished either by providing Security Administrator-configurable lifetimes (with appropriate FMT requirements and instructions in*~~

~~*documents mandated by AGD_OPE), or by "hard coding" the limits in the implementation. The ST author selects the amount of data in the range specified by the requirement.*~~

~~*In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be specified through FMT requirements and included in the administrative guidance generated for AGD_OPE.*~~

~~***Assurance Activity:***~~
~~*The evaluator checks to ensure that the TSS describes how lifetimes for IKEv1 Phase 2 SAs--with respect to the amount of traffic that is allowed to flow using a given SA--are established. If the value is configurable, then the evaluator verifies that the appropriate instructions for configuring these values are included in the operational guidance. The evaluator also performs the following test:*~~

- ~~Test 1: The evaluator shall construct a test where a Phase 2 SA is established and attempted to be maintained while more data than is specified in the above assignment flows over the connection. The evaluator shall observe that this SA is closed or renegotiated before the amount of data specified is exceeded. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.~~

FCS_IPSEC_EXT.1.9 ~~5~~ The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [selection: *24 (2048-bit MODP with 256-bit POS), 19 (256-bit Random ECP), 20 (384-bit Random ECP, 5 (1536-bit MODP))*, [assignment: *other DH groups that are implemented by the TOE*], *no other DH groups*].

Application Note: The above requires that the TOE support DH Group 14. If other groups are supported, then those should be selected (for groups 24, 19, ~~and~~ 20, and 5) or specified in the assignment above; otherwise "no other DH groups" should be selected. This applies to IKEv1 ~~and (if implemented)~~ /IKEv2 exchanges.

In future publications of this PP DH Groups 19 (256-bit Random ECP) and 20 (384-bit RandomECP) will be required.

Assurance Activity:
The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer. The evaluator shall also perform the following test (this test may be combined with other tests for this component, for instance, the tests associated with FCS_IPSEC_EXT.1.1):

- Test 1: For each supported DH group, the evaluator shall test to ensure that all IKE protocols can be successfully completed using that particular DH group.

FCS_IPSEC_EXT.1.10 ~~6~~ The TSF shall ensure that all IKE protocols ~~implement~~ perform Peer Authentication using the [selection: ~~*DSA, rDSA*~~ *RSA*, *ECDSA*] algorithm and Pre-shared Keys.

Application Note: The selected algorithm should correspond to an appropriate selection for FCS_COP.1(2). If IPsec is included in the TOE, the ST author also includes FIA_PSK_EXT from Appendix C.

Assurance Activity:
The evaluator shall check that the TSS contains a description of the IKE peer authentication process used by the TOE, and that this description covers the use of the signature algorithm or algorithms specified in the requirement. The evaluator shall also perform the following test:

- Test 1: For each supported signature algorithm, the evaluator shall test that peer authentication using that algorithm can be successfully achieved and results in the successful establishment of a connection.

~~FCS_IPSEC_EXT.1.7 The TSF shall support the use of pre-shared keys (as referenced in the RFCs) for use in authenticating its IPsec connections.~~

~~***Assurance Activity:***~~
~~*The evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The evaluator shall check that the operational guidance describes how pre-shared keys are to be generated and established for a TOE. The description in the TSS and the operational guidance shall also indicate how pre-shared key establishment is accomplished for both TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key. The evaluator shall also perform the following test:*~~

- ~~Test 1: The evaluator shall generate a pre-shared key and use it, as indicated in the operational guidance, to establish an IPsec connection between two peers. If the TOE supports generation of the pre-shared key, the evaluator shall ensure that establishment of the key is carried out for an instance of the TOE generating the key as well as an instance of the TOE merely taking in and using the key.~~

~~FCS_IPSEC_EXT.1.8 The TSF shall support the following:~~

1. ~~*Pre-shared keys shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters: [selection: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [assignment: other characters]];*~~

2. ~~*Pre-shared keys of 22 characters and [selection: [assignment: other supported lengths], no other lengths].*~~

~~*Application Note: The ST author selects the special characters that are supported by TOE; they may optionally list additional special characters supported using the assignment. For the length of the pre-shared keys, a common length (22 characters) is required to help promote interoperability. If other lengths are supported they should be listed in the assignment; this assignment can also specify a range of values (e.g., "lengths from 5 to 55 characters") as well.*~~

~~***Assurance Activity:***~~

~~*The evaluator shall check the operational guidance to ensure that it describes the generation of pre-shared keys, including guidance on generating strong keys and the allowed character set. The evaluator shall check that this guidance does not limit the pre-shared key in a way that would not satisfy the requirement. It should be noted that while the administrator (in contravention to the operational guidance) can choose a key that does not conform to the requirement, there is no requirement that the TOE check the key to ensure that it meets the rules specified in this component. However, should the administrator choose to create a password that conforms to the rules above (and the operational guidance); the TOE should not prohibit such a choice. The evaluator shall also perform the following test; this may be combined with Test 1 for FCS_IPSEC_EXT.1.7:*~~

- ~~Test 1: The evaluator shall generate a pre-shared key that is 22 characters long that meets the composition requirements above. The evaluator shall then use this key to successfully establish an IPsec connection. While the evaluator is not required to test that all of the special characters or lengths listed in the requirement are supported, it is required that they justify the subset of those characters chosen for testing, if a subset is indeed used.~~

3.5 Pre-Shared Key Composition (FIA)

The following addition is to be inserted as the last item in Appendix C section C.1.1, immediately before section C1.2 on page 57 of the NDPP.

FIA_PSK_EXT (Extended: Pre-Shared Key Composition)

The TOE must support pre-shared keys for use in the IPsec protocol. There are two types of pre-shared keys--text-based (which are required) and bit-based (which are optional)--supported by the TOE, as specified in the requirements below. The first type is referred to as “text-based pre-shared keys”, which refer to pre-shared keys that are entered by users as a string of characters from a standard character set, similar to a password. Such pre-shared keys must be conditioned so that the string of characters is transformed into a string of bits, which is then used as the key.

The second type is referred to as “bit-based pre-shared keys” (for lack of a standard term); this refers to keys that are either generated by the TSF on a command from the administrator, or input in "direct form" by an administrator. "Direct form" means that the input is used directly as the key, with no "conditioning" as was the case for text-based pre-shared keys. An example would be a string of hex digits that represent the bits that comprise the key.

The requirements below mandate that the TOE must support text-based pre-shared keys and optionally support bit-based pre-shared keys, although generation of the bit-based pre-shared keys may be done either by the TOE or in the operational environment.

FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition

FIA_PSK_EXT.1.1 The TSF shall be able to use pre-shared keys for IPsec.

FIA_PSK_EXT.1.2 The TSF shall be able to accept text-based pre-shared keys that:

- are 22 characters and *[selection: [assignment: other supported lengths], no other lengths]*;
- composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”).

FIA_PSK_EXT.1.3 The TSF shall condition the text-based pre-shared keys by using [selection: SHA-1, SHA-256, SHA-512, [assignment: *method of conditioning text string*]] and be able to [selection: use no other pre-shared keys; accept bit-based pre-shared keys; generate bit-based pre-shared keys using the random bit generator specified in FCS_RBG_EXT.1].

Application Note:

For the length of the text-based pre-shared keys, a common length (22 characters) is required to help promote interoperability. If other lengths are supported they should be listed in the assignment; this assignment can also specify a range of values (e.g., "lengths from 5 to 55 characters") as well.

In the second selection for FIA_PSK_EXT.1.3, the ST author fills in the method by which the text string entered by the administrator is “conditioned” into the bit string used as the key. This can be done by using one of the specified hash functions, or some other method through the assignment *statement. If “bit-based pre-shared keys” is selected, the ST author specifies whether the TSF merely accepts bit-based pre-shared keys, or is capable of generating them. If it generates them, the requirement specified that they must be generated using the RBG specified by the requirements. If the use of bit-based pre-shared keys is not supported, the ST author chooses “*use no other pre-shared keys”.

Assurance Activity:

The evaluator shall examine the operational guidance to determine that it provides guidance on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the merits of shorter or longer pre-shared keys. The guidance must specify the allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2.

The evaluator shall examine the TSS to ensure that it states that text-based pre-shared keys of 22 characters are supported, and that the TSS states the conditioning that takes place to transform the text-based pre-shared key from the key sequence entered by the user (e.g., ASCII representation) to the bit string used by IPsec, and that this conditioning is consistent with the first selection in the FIA_PSK_EXT.1.3 requirement. If the assignment is used to specify conditioning, the evaluator will confirm that the TSS describes this conditioning.

If “bit-based pre-shared keys” is selected, the evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both). The evaluator shall also examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1.

The evaluator shall also perform the following tests:

- *Test 1: The evaluator shall compose at least 15 pre-shared keys of 22 characters that cover all allowed characters in various combinations that conform to the operational guidance, and demonstrates that a successful protocol negotiation can be performed with each key.*

- *Test 2 [conditional]: If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length; the maximum length; and an invalid length. The minimum and maximum length tests should be successful, and the invalid length must be rejected by the TOE.*

- *Test 3 [conditional]: If the TOE supports bit-based pre-shared keys but does not generate such keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.*

- *Test 4 [conditional]: If the TOE supports bit-based pre-shared keys and does generate such keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.*

3.6 TLS Protocol

The following changes are to the FCS_TLS_EXP component on pages 52-54 of the NDPP.

FCS_TLS_EXT.1 Explicit: TLS

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
~~TLS_RSA_WITH_AES_256_CBC_SHA~~
~~TLS_DHE_RSA_WITH_AES_128_CBC_SHA~~
~~TLS_DHE_RSA_WITH_AES_256_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_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
].

Application Note: The ST author must make the appropriate selections and assignments to reflect the TLS implementation. ~~The ST author must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done either by adding elements to this component, or by additional detail in the TSS.~~

The ciphersuites to be ~~used~~ tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suite~~s~~, then "None" should be selected. If administrative steps need to be taken so that the suites negotiated by the implementation are limited to those in this requirement, the appropriate instructions need to be contained in the guidance called for by AGD_OPE.

The Suite B algorithms (RFC 5430) listed above are the preferred algorithms for implementation. The TLS requirement will be changed in the next version of the NDPP to comply with CNSSP 15 and NIST SP 800-131A. ~~Since the Dec. 2010 publication of NDPP v1.0, there has been limited progress~~

~~*with respect to extending the prevalence of TLS 1.2 support in commercial network devices. Future publications of this PP will require support for TLS 1.2 (RFC 5246); however, it is likely that an NDPP v2.0 will be published in late 2012 which will not include a requirement for TLS 1.2 support, but will require that the TOE offer a means to deny all connection attempts using SSL 2.0 or SSL 3.0.*~~

Assurance Activity:
The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that ~~optional characteristics (e.g., extensions supported, client authentication supported) are specified, and~~ the ciphersuites supported are specified ~~as well~~. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of a HTTPS session. It is sufficient to observe ~~(on the wire)~~ the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

- Test 2: The evaluator shall setup a man-in-the-middle tool between the TOE and the TLS Peer and shall perform the following modifications to the traffic:
    - [Conditional: TOE is a server] Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the server denies the client's Finished handshake message.
    - [Conditional: TOE is a client] Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.
    - [Conditional: TOE is a client] If a DHE or ECDHE ciphersuite is supported, modify the signature block in the Server's KeyExchange handshake message, and verify that the client rejects the connection after receiving the Server KeyExchange.
    - [Conditional: TOE is a client] Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.

3.7 SSH Protocol

The following changes are to the FCS_SSH_EXP component on pages 54-56 of the NDPP.

FCS_SSH_EXT.1 Explicit: SSH

FCS_SSH_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, ~~and~~ 4254, and [selection: 5656, 6668, no other RFCs].

Application Note: The ST author ~~must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done either by adding elements to this component, or by additional detail in the TSS.~~ selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted).

*In the next version of this PP, a requirement will be added regarding rekeying. The requirement will read "*The TSF shall ensure that the SSH connection be rekeyed after no more than 2^{28} packets have been transmitted using that key."

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

Assurance Activity:
The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSH_EXT.1.5, and ensure that password-based authentication methods are also allowed. The evaluator shall also perform the following tests:

- Test 1: The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the operational guidance.

- Test 2: Using the operational guidance, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.

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

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

Assurance Activity:
The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled. The evaluator shall also perform the following test:

- Test 1: The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

FCS_SSH_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM, *no other algorithms*].

Application Note: In the assignment, the ST author can select the AES-GCM algorithms, or "no other algorithms" if AES-GCM is not supported. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST. ~~Since the Dec. 2010 publication of NDPP v1.0, there has been consider progress with respect to the prevalence of AES-GCM support in commercial network devices. It is likely that an NDPP v2.0 will be published in late 2012 which will require AES-GCM and AES-CBC will become optional.~~

Assurance Activity:
The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of ~~a protocol~~ the algorithm to satisfy the intent of the test.

FCS_SSH_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses [selection: SSH_RSA, ecdsa-sha2-nistp256] and [selection: *PGP-SIGN-RSA, PGP-SIGN-DSS, ecdsa-sha2-nistp384, no other public key algorithms,*] as its public key algorithm(s)

~~*Application Note: RFC 4253 specifies required and allowable public key algorithms. This requirement makes SSH-RSA "required" and allows two others to be claimed in the ST. The ST author should make the appropriate selection, selecting "no other public key algorithms" if only SSH_RSA is implemented.*~~

Application Note: Implementations that select only SSH_RSA will not achieve the 112-bit security strength in the digital signature generation for SSH authentication as is recommended in NIST SP 800-131A. Future versions of this profile will likely disallow the option of selecting only SSH_RSA.

Assurance Activity:
The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.

FCS_SSH_EXT.1.6 The TSF shall ensure that data integrity algorithms used in SSH transport connection is [*selection: hmac-sha1, hmac-sha1-96, ~~hmac-md5, hmac-md5-96~~ hmac-sha2-256, hmac-sha2-512*].

Application Note: RFC 6668 specifies the use of the sha2 algorithms in SSH.

Assurance Activity:
The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component. The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the "none" MAC algorithm is not allowed). The evaluator shall also perform the following test:

- Test 1: The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

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

Assurance Activity:
The evaluator shall ensure that operational guidance contains configuration information that will allow the security administrator to configure the TOE so that all key exchanges for SSH are performed using DH group 14 and any groups specified from the selection in the ST. If this capability is "hard-coded" into the TOE, the evaluator shall check the TSS to ensure that this is stated in the discussion of the SSH protocol. The evaluator shall also perform the following test:

- Test 1: The evaluator shall attempt to perform a diffie-hellman-group1-sha1 key exchange, and observe that the attempt fails. For each allowed key exchange method, ~~T~~ the evaluator shall then attempt to perform a ~~diffie-hellman-group14-sha1~~ key exchange using that method, and observe that the attempt succeeds.

3.8 Generation of Asymmetric Keys

The following changes are to the FCS_CKM.1 component on page 14 of the NDPP.

FCS_CKM.1 Cryptographic Key Generation (for asymmetric keys)

FCS_CKM.1.1 **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 186-3, "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*]

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

Application Note:

This component requires that the TOE be able to generate the public/private key pairs that are used for key establishment purposes for the various cryptographic protocols used by the TOE (e.g., IPsec). If multiple schemes are supported, then the ST author should iterate this requirement to capture this capability. The scheme used will be chosen by the ST author from the selection.

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 to the protocols specified in this PP.

SP 800-56B references (but does not mandate) key generation according to FIPS 186-3. For purposes of compliance in this version of the NDPP, RSA key pair generation according to FIPS 186-2 or FIPS 186-3 is allowed in order for the TOE to claim conformance to SP 800-56B.

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.

Assurance Activity:

The evaluator shall use the key pair generation portions of "The FIPS 186-3 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and either "The RSA Validation System (RSA~~2~~VS)" (for FIPS 186-2) or "The 186-3 RSA Validation System (RSA2VS)" (for FIPS 186-3) 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.*

The evaluator shall ensure that the TSS contains a description of how the ~~In order to show that 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.~~, the evaluator shall ensure that the TSS contains the following information:The TSS shall list all sections of the appropriate 800-56 standard(s) to which the TOE complies.~~

~~For each applicable section listed in the TSS, for all statements that are not "shall" (that is, "shall not", "should", and "should not"), if the TOE implements such options it shall be described in the TSS. If the included functionality is indicated as "shall not" or "should not" in the standard, the TSS shall provide a rationale for why this will not adversely affect the security policy implemented by the TOE;~~

~~For each applicable section of 800-56A and 800-56B (as selected), any omission of functionality related to "shall" or "should" statements shall be described;~~

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

3.9 Cryptographic Operations: Digital Signatures

The following changes are to the FCS_COP.1(2) component on page 16 of the NDPP.

FCS_COP.1(2) Cryptographic Operation (for cryptographic signature)

FCS_COP.1.1(2) **Refinement:** The TSF shall perform **cryptographic signature services** in accordance with a ***[selection:***

(1) Digital Signature Algorithm (DSA) with a key size (modulus) of 2048 bits or greater,

(2) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater, or

(3) Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater]

Application Note: As the preferred approach for cryptographic signature, elliptic curves will be required in future publications of this PP.

that meets the following:

Case: Digital Signature Algorithm

- **FIPS PUB 186-3, "Digital Signature Standard"**

Case: RSA Digital Signature Algorithm

- **FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"**

Case: Elliptic Curve Digital Signature Algorithm

- **FIPS PUB 186-3, "Digital Signature Standard"**
- **The TSF shall implement "NIST curves" P-256, P-384 and [selection: P-521, no other curves] (as defined in FIPS PUB 186-3, "Digital Signature Standard").**

Application Note: The ST Author should choose the algorithm implemented to perform digital signatures; if more than one algorithm is available, this requirement (and the corresponding FCS_CKM.1 requirement) should be iterated to specify the functionality. For the algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.

For elliptic curve-based schemes, the key size refers to the log_2 of the order of the base point. As the preferred approach for digital signatures, ECDSA will be required in future publications of this PP.

Assurance Activity:

The evaluator shall use the signature generation and signature verification portions of "The Digital Signature Algorithm Validation System” (~~DSAVS or~~ DSA2VS), "The Elliptic Curve Digital Signature Algorithm Validation System” (~~ECDSAVS or~~ ECDSA2VS), and "The RSA Validation System” (RSAVS (for 186-2) or RSA2VS (for 186-3)) as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e., FIPS PUB 186-2 or FIPS PUB 186-3). This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

3.10 Allowed Symmetric Encryption Bit Lengths

The following changes are to the FCS_COP.1(1) component on page 16 of the NDPP.

FCS_COP.1(1) Cryptographic Operation (for data encryption/decryption)

FCS_COP.1.1(1) **Refinement:** The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES operating in* [**selection:** CBC, GCM, [**assignment:** ***one or more modes***]]] and cryptographic key sizes 128-bits~~,~~ and 256-bits~~, and [~~~~**selection: 192 bits, no other key sizes**~~~~]~~ that meets the following:

- **FIPS PUB 197, “Advanced Encryption Standard (AES)”**
- **[Selection: NIST SP 800-38A, NIST SP 800-38B, NIST SP 800-38C, NIST SP 800-38D, NIST SP 800-38E]**

Application Note: For the ~~assignment~~ first selection, the ST author should choose the mode or modes in which AES operates to support the cryptographic protocols chosen for FTP_ITC and FTP_TRP. If any other modes are used to support requirements in the ST, those should be filled in through the assignment. ~~For the first selection, the ST author should choose the key sizes that are supported by this functionality.~~ For the second selection, the ST author should choose the standards that describe the modes specified in the first selection and the assignment.

Assurance Activity:

The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", "The XTS-AES Validation System (XTSVS)", The CMAC Validation System (CMACVS)", "The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)" (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

3.11 Entropy for Random Bit Generation

The following changes are to the FCS_RBG_EXT.1 component on page 18 of the NDPP.

FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: NIST Special Publication 800-90 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES), Dual_EC_DRBG (any)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulated entropy from [selection, one or both of: a software-based noise source; a TSF-hardware-based noise source].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [selection, choose one of: 128 bits, 256 bits] of entropy at least equal to the greatest ~~bit length~~ security strength of the keys and ~~authorization factors~~ hashes that it will generate.

Application Note: NIST Special Pub 800-90~~, Appendix C~~ B describes the minimum entropy measurement that will probably be required future versions of FIPS-140. If possible this should be used immediately and will be required in future versions of this PP.

For the first selection in FCS_RBG_EXT.1.1, the ST author should select the standard to which the RBG services comply (either 800-90B or 140-2 Annex C).

SP 800-90B contains four different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used (if 800-90B is selected), and include the specific underlying cryptographic primitives used in the requirement or in the TSS. While any of the identified hash functions (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CT_DRBG are allowed. While any of the curves defined in 800-90B are allowed for Dual_EC_DRBG, the ST author not only must include the curve chosen, but also the hash algorithm used.

For the second selection in FCS_RBG_EXT.1.1, the ST author indicates whether the sources of entropy are software-based, hardware-based, or both. If there are multiple sources of entropy, the ST will elaborate each entropy sources and whether it is hardware- or software-based. Hardware-based noise sources are preferred.

Note that for FIPS Pub 140-2 Annex C, currently only the method described in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3 is valid. If the key length for the AES implementation used here is different than that used to encrypt the user data, then FCS_COP.1 may have to be adjusted or iterated to reflect the different key length. For the selection in FCS_RBG_EXT.1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG.

The ST author also ensures that any underlying functions are included in the baseline requirements for the TOE.

For the selection in FCS_RBG_EXT.1.2, the ST author selects the appropriate number of bits of entropy that corresponds to the greatest security strength of the algorithms included in the ST. Security strength is defined in Tables 2 and 3 of NIST SP 800-57A. For example, if the implementation includes 2048-bit RSA (security strength of 112 bits), AES 128 (security strength 128 bits), and HMAC-512 (security strength 256 bits), then the ST author would select 256 bits.

Assurance Activity:

Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Annex D, Entropy Documentation and Assessment.

The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.

Implementations Conforming to FIPS 140-2, Annex C

The reference for the tests contained in this section is *The Random Number Generator Validation System (RNGVS)* [RNGVS]. The evaluator shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.

The evaluator shall perform a Variable Seed Test. The evaluator shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within the set. The evaluator ensures that the values returned by the TSF match the expected values.

The evaluator shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluator then invokes the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in *NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms*, Section 3. The evaluator ensures that the 10,000th value produced matches the expected value.

Implementations Conforming to NIST Special Publication 800-90

The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RBG functionality.

If the RBG has prediction resistance enabled, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90).

If the RBG does not have prediction resistance, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

Entropy input: the length of the entropy input value must equal the seed length.

Nonce: If a nonce is supported (CTR_DRBG with no df does not use a nonce), the nonce bit length is one-half the seed length.

Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.

Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

3.12 Labeling the TOE

The following changes are to the ALC_CMC.1 component on page 41 of the NDPP.

4.3.5.1 ALC_CMC.1 Labeling of the TOE

This component is targeted at identifying the TOE such that it can be distinguished from other products or versions from the same vendor and can be easily specified when being procured by an end user.

Developer action elements:

ALC_CMC.1.1D The developer shall provide the TOE and a reference for the TOE.

Content and presentation elements:

ALC_CMC.1.1C The TOE shall be labeled with its unique reference.

Evaluator action elements:

ALC_CMC.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

Assurance Activity:

The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. The evaluator shall ensure that this identifier is sufficient for an acquisition entity to use in procuring the TOE (including the appropriate administrative guidance) as specified in the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

3.13 Auditing Shutdown of Audit

The following changes are to the FAU_GEN.1 component on page 11 of the NDPP.

2.1.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) *All administrative actions*;
d) [*Specifically defined auditable events listed in Table 1*].

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

Many auditable aspects of the SFRs included in this document deal with administrative actions. Item c above requires all administrative actions to be auditable, so no additional specification of the auditability of these actions is specified in Table 1.

Assurance Activity:
The evaluator shall check the administrative guide and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP is described and that the description of the fields contains the information required in FAU_GEN1.2, and the additional information specified in Table 1.

The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP. The evaluator shall examine the administrative guide and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security relevant with respect to this PP. The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in table 1 and administrative actions. This should include all instances of an event--for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.