Extended Package for Secure Shell (SSH)

Version: 1.0

2016-02-19

National Information Assurance Partnership

Revision History

Version	Date	Comment
1.0	2016-02-19	Initial Release - EP for OS PP, MDM PP, App PP

Contents

1. Introduction

1.1 Overview

Secure Shell (SSH) is a protocol for secure remote login and other secure network services over an untrusted network. SSH software can act as a client, server, or both.

1.2 Terms

The following sections provide both Common Criteria and technology terms used in this Extended Package.

1.2.1 Common Criteria Terms

Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation.
Extended Package (EP)	An implementation-independent set of security requirements for a category of products, which extends those in a Protection Profile.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation (TOE)	The product under evaluation.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in a ST.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Assurance Requirement	A requirement to assure the security of the TOE.

(SAR)	

1.2.2 Technology Terms

Secure Shell (SSH)	Cryptographic network protocol for initiating text-based shell sessions on remote systems.

1.3 Compliant Targets of Evaluation

The Target of Evaluation (TOE) in this EP is a product which acts as an SSH client or server, or both.
This EP describes the extended security functionality of SSH in terms of [CC]. This EP can extend the Protection Profiles for *Application Software*, *General-Purpose Operating Systems*, or *Mobile Device Management*. It is expected that the content of this EP will be appropriately combined with the base PP to include selection-based requirements in accordance with the selections and/or assignments made, and any optional and/or objective components.
This EP combind with the [AppPP] must include selection-based requirements in accordance with the selections and/or assignments made, and any optional and/or objective components to include: FCS_CKM.2.1, FCS_COP.1.1(*), FCS_RBG_EXT.2.*, FCS_TLSC_EXT.1.*, FIA_X509_EXT.1.*, FIA_X509_EXT.2.*.
This EP combind with the [GPOSPP] must include selection-based requirements in accordance with the selections and/or assignments made, and any optional and/or objective components to include: FCS_CKM.2.1, FCS_COP.1.1(*), FCS_RBG_EXT.1.*, FCS_TLSC_EXT.1.*, FIA_X509_EXT.1.*, FIA_X509_EXT.2.*.
This EP combind with the [MDMPP] must include selection-based requirements in accordance with the selections and/or assignments made, and any optional and/or objective components to include: FCS_CKM.1.1, FCS_COP.1.1(*), FCS_RBG_EXT.1.*, FCS_TLSC_EXT.1.*, FIA_X509_EXT.1.*, FIA_X509_EXT.2.*.
An ST must identify the applicable version of the base PP and this EP in its conformance claims.

2. Conformance Claims

Conformance Statement

This EP serves to extend the PP baselines with additional SFRs and associated Assurance Activities specific to SSH clients and servers. Assurance Activities are the actions that the evaluator performs in order to determine an SSH client or server's compliance to the SFRs.

This EP conforms to Common Criteria [CC] for Information Technology Security Evaluation, Version 3.1, Revision 4. It is CC Part 2 extended and CC Part 3 conformant. In order to be conformant to this EP, the ST must include all components in this EP and the associated base PP that are:

- unconditional (which are always required)
- selection-based (which are required when certain selections are chosen in the unconditional requirements)

and may include optional and/or objective components that are desirable but not required for conformance.

In accordance with CC Part 1, dependencies are not included when they are addressed by other SFRs. The assurance activities provide adequate proof that any dependencies are also satisfied.

3. Security Problem Description

This Extended Package does not repeat the threats, assumptions, and organizational security policies identified in any base PPs, though they all apply given the conformance and hence dependence of this EP on it. Together the threats, assumptions and organizational security policies of the base PP and those defined in this EP describe those addressed by the product as the Target of Evaluation.

4. Security Requirements

This chapter describes the security requirements to be fulfilled by the product. Those requirements comprise functional components from Part 2 of [CC]. The following notations are used:

- **Selection** (denoted by *italicized text*): is used to select one or more options provided by the [CC] in stating a requirement.
- **Assignment** operation (denoted by *italicized text*): is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: are identified with a number inside parentheses (e.g. "(1)").

4.1 Security Functional Requirements

The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, with additional extended functional components.

4.1.1 Cryptographic Support (FCS)

FCS_COP.1(1) Cryptographic Operation - Encryption/Decryption (Refined)

FCS_COP.1.1(1) The SSH software shall perform encryption/decryption services for data in accordance with a specified cryptographic algorithm AES-CTR (as defined in NIST SP 800-38A) mode and cryptographic key sizes [**selection**: *128-bit*, *256-bit*] .

Application Note: If this EP is extending the Protection Profile for Application Software, it may not be necessary to include the FCS claims (in the base PP) since the SSH application could rely on the platform for this functionality. If the SSH software does provide its own crypto the FCS claims will need to reflect that in the FCS selections from the base PP.

Assurance Activity

The evaluator shell review the TSF of the base PP to verify consistency with the functionality that was claimed by the base PP to ensure that applicable dependencies are met.
The evaluator shall verify that the TSS describes the counter mechanism including rationale that the counter

values provided are unique.

AES-CTR Tests:

- ***Test 1:*** *Known Answer Tests (KATs)*
  There are four Known Answer Tests (KATs) described below. For all KATs, the plaintext, IV, and ciphertext values shall be 128-bit blocks. The results from each test may either be obtained by the validator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

  To test the encrypt functionality, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all zeros key, and the other five shall be encrypted with a 256-bit all zeros key. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input.

  To test the encrypt functionality, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value and an IV of all zeros. Five of the key values shall be 128-bit keys, and the other five shall be 256-bit keys. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using an all zero ciphertext value as input.

  To test the encrypt functionality, the evaluator shall supply the two sets of key values described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values an an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second shall have 256 256-bit keys. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N]. To test the decrypt functionality, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value *that results from decryption of the given ciphertext using the given key values and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit pairs. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros for i in [1, N]. The ciphertext value in each pair shall be the value that results in an all zeros plaintext when decrypted with its corresponding key.*

To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from encryption of the given plaintext using a 128-bit key value of all zeros and using a 256 bit key value of all zeros, respectively, and an IV of all zeros. Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128]. To test the decrypt functionality, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input.

- ***Test 2:*** *Multi-Block Message Test*
  The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10. For each i the evaluator shall choose a key, IV, and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation. The evaluator shall also test the decrypt functionality by decrypting an i-block message where 1 less-than i less-than-or-equal to 10. For each i the evaluator shall choose a key and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key using a known good implementation.
- ***Test 3:*** *Monte-Carlo Test*
  For AES-CTR mode perform the Monte Carlo Test for ECB Mode on the encryption engine of the counter mode implementation. There is no need to

test the decryption engine.

The evaluator shall test the encrypt functionality using 200 plaintext/key pairs. 100 of these shall use 128 bit keys, and 100 of these shall use 256 bit keys. The plaintext values shall be 128-bit blocks. For each pair, 1000 iterations shall be run as follows:

```
For AES-ECB mode
# Input: PT, Key
for i = 1 to 1000:
CT[i] = AES-ECB-Encrypt(Key, PT)
PT = CT[i]
```

The ciphertext computed in the 1000th iteration is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1 The SSH software shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254 and [**selection**: *5647*, *5656*, *6187*, *6668*, *no other RFCs*] as a [**selection**: *client*, *server*]

Application Note: The ST author selects which of the additional RFCs to which conformance is being claimed. An SSH product can implement additional RFCs, but only those listed in the selection can be claimed as conformant under common criteria. The RFC selections for this requirement need to be consistent with selections in later elements of this Extended Package (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that from the IETF's perspective the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in later elements of this Extended Package is out of scope for the assurance activity for this requirement.
If client is selected, then the ST must include the requirements from FCS_SSHC_EXT.1. If server is selected, then the ST must include the requirements from FCS_SSHS_EXT.1.

Assurance Activity

The evaluator will ensure that the selections indicated in the ST are consistent with selections in the dependent components.

A. Optional Requirements

The baseline requirements (those that must be performed by the TOE) are contained in the body of this EP. Additionally, there are three other types of requirements specified in Appendix A, Appendix B, and Appendix C. The first type (in this Appendix) are requirements that can be included in the ST, but are not required in order for products to claim conformance to this EP. The second type (in Appendix B) are requirements based on selections in the body of the EP: if certain selections are made, then additional requirements in that appendix must be included. The third type (in Appendix C) are components that are not required in order to conform to this EP, but will be included in the baseline requirements in future versions of this EP, so adoption by vendors is encouraged. Note that the ST author is responsible for ensuring that requirements that may be associated with those in Appendix A, Appendix B, and Appendix C but are not listed (e.g., FMT-type requirements) are also included in the ST.
None exists presently.

B. Selection-Based Requirements

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

FCS_SSHC_EXT.1 SSH Protocol - Client

FCS_SSHC_EXT.1.1 The SSH client shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, and [**selection**: *password-based*, *none*] .

This requirement depends upon selection in FCS_SSH_EXT.1.1.

Assurance Activity

The evaluator will 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_SSHC_EXT.1.4, and ensure that password-based authentication methods are also allowed.

- ***Test 1:*** *The evaluator will, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection to an SSH server. Any configuration activities required to support this test shall be performed according to instructions in the guidance documentation.*
- ***Test 2:*** *Using the guidance documentation, the evaluator will configure the TOE to perform password-based authentication to an SSH server, and demonstrate that a user can be successfully authenticated by the TOE to an SSH server using a password as an authenticator.*

FCS_SSHC_EXT.1.2 The SSH client shall ensure that, as described in RFC 4253, packets greater than [**assignment**: *number of bytes]* bytes in an SSH transport connection are dropped.

This requirement depends upon selection in FCS_SSH_EXT.1.1.

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 will check that the TSS describes how “large packets” in terms of RFC 4253 are detected and handled.
The evaluator will demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

FCS_SSHC_EXT.1.3 The SSH software shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, [**selection**: *aes128-cbc*, *aes256-cbc*, *AEAD_AES_128_GCM*, *AEAD_AES_256_GCM*, *no other algorithms*] .

This requirement depends upon selection in FCS_SSH_EXT.1.1.

Application Note: RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST.

Assurance Activity

The evaluator will 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 will check the TSS to ensure that the

encryption algorithms specified are identical to those listed for this component.
The evaluator will also check the guidance documentation 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).

- ***Test 1:*** *The evaluator will establish an SSH connection using each of the encryption 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.*
- ***Test 2:*** *The evaluator will configure an SSH server to only allow the 3des-cbc encryption algorithm and no other encryption algorithms. The evaluator will attempt to establish an SSH connection from the TOE to the SSH server and observe that the connection is rejected.*

FCS_SSHC_EXT.1.4 The SSH client shall ensure that the SSH transport implementation uses [**selection**: *ssh-rsa*, *ecdsa-sha2-nistp256*] and [**selection**: *ecdsa-sha2-nistp384*, *x509v3-ecdsa-sha2-nistp256*, *x509v3-ecdsa-sha2-nistp384*, *no other public key algorithms*] as its public key algorithm(s) and rejects all other public key algorithms.

This requirement depends upon selection in FCS_SSH_EXT.1.1.

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 document may remove ssh-rsa as a selection. If x509v3-ecdsa-sha2-nistp256 or x509v3-ecdsa-sha2-nistp384 are selected, then the list of trusted certification authorities must be selected in FCS_SSHC_EXT.1.8.
The SFRs for cryptographic key generation and certificate validation are inherited from the base PP.

Assurance Activity

The evaluator will check the description of the implementation of this protocol in the TSS to ensure that

optional characteristics are specified, and the public key algorithms supported are specified as well. The evaluator will check the TSS to ensure that the public key algorithms specified are identical to those listed for this component.
The evaluator will also check the guidance documentation 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).

- ***Test 1:*** *The evaluator will establish a SSH connection using each of the public key algorithms specified by the requirement to authenticate an SSH server to the TOE. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.*
- ***Test 2:*** *The evaluator will configure an SSH server to only allow the ssh-dsa public key algorithm and no other public key algorithms. The evaluator will attempt to establish an SSH connection from the TOE to the SSH server and observe that the connection is rejected.*

FCS_SSHC_EXT.1.5 The SSH client shall ensure that the SSH transport implementation uses [**selection**: *hmac-sha1*, *hmac-sha1-96*, *hmac-sha2-256*, *hmac-sha2-512*] and [**selection**: *AEAD_AES_128_GCM*, *AEAD_AES_256_GCM*, *no other MAC algorithms*] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

This requirement depends upon selection in FCS_SSH_EXT.1.1.

Application Note: RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH.
The SFRs for cryptographic operations, encryption and hashing, are inherited from the base PP.

Assurance Activity

The evaluator will 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 will also check the guidance documentation 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).

- ***Test 1:*** *The evaluator will 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.*
- ***Test 2:*** *The evaluator will configure an SSH server to only allow the "none" MAC algorithm. The evaluator will attempt to connect from the TOE to the SSH server and observe that the attempt fails.*
- ***Test 3:*** *The evaluator will configure an SSH server to only allow the hmac- md5 MAC algorithm. The evaluator will attempt to connect from the TOE to the SSH server and observe that the attempt fails.*

FCS_SSHC_EXT.1.6 The SSH client shall ensure that [**selection**: *diffie-hellman-group14-sha1*, *ecdh-sha2-nistp256*] and [**selection**: *ecdh-sha2-nistp384*, *ecdh-sha2-nistp521*, *no other methods*] are the only allowed key exchange methods used for the SSH protocol.

This requirement depends upon selection in FCS_SSH_EXT.1.1.

Assurance Activity

The evaluator will check the TSS to ensure that it lists the supported key exchange algorithms, and that that list corresponds to the list in this component.
The evaluator will also check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections with the TOE.

- ***Test 1:*** *The evaluator will configure an SSH server to permit all allowed key exchange methods. The evaluator will attempt to connect from the TOE to the SSH server using each allowed key exchange method, and observe that each attempt succeeds.*

FCS_SSHC_EXT.1.7 The SSH server shall ensure that the SSH connection be rekeyed after [**selection**: *no more than 2^{28} packets have been transmitted*, *no more than 1 Gigabyte of data has been transmitted*, *no more than 1 hour*] using that key.

This requirement depends upon selection in FCS_SSH_EXT.1.1.

Assurance Activity

- ***Test 1:*** *The evaluator will configure an SSH server to create a log entry when a rekey occurs. The evaluator will connect to an SSH server with the TOE and cause a rekey to occur according to the selection(s) in the ST, and subsequently review the audit log to ensure that a rekey occurred.*

FCS_SSHC_EXT.1.8 The SSH client shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or [**selection**: *a list of trusted certification authorities*, *no other methods*] as described in RFC 4251 section 4.1.

This requirement depends upon selection in FCS_SSH_EXT.1.1.

Application Note: The list of trusted certification authorities can only be selected if x509v3-ecdsa-sha2-nistp256 or x509v3-ecdsa-sha2-nistp384 are selected in FCS_SSHC_EXT.1.4.

Assurance Activity

- ***Test 1:*** *The evaluator will delete all entries in the TOE's list of recognized SSH server host keys and, if selected, all entries in the TOE's list of trusted certification authorities. The evaluator will initiate a connection from the TOE to an SSH server. The evaluator shall ensure that the TOE either rejects the connection or displays the SSH server's public key (either the key bytes themselves or a hash of the key using any allowed hash algorithm) and prompts the user to accept or deny the key before continuing the connection.*
- ***Test 2:*** *The evaluator will add an entry associating a host name with a public key into the TOE's local database. The evaluator will replace, on the corresponding SSH server, the server's host key with a different host key. The evaluator will initiate a connection from the TOE to the SSH server using password-based authentication, shall ensure that the TOE rejects the connection, and shall ensure that the password was not transmitted to the SSH server (for example, by instrumenting the SSH server with a debugging capability to output received passwords).*

FCS_SSHS_EXT.1 SSH Protocol - Server

FCS_SSHS_EXT.1.1 The SSH server shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, and [**selection**: *password-based*, *none*] .

This requirement depends upon selection in FCS_SSH_EXT.1.1.

Assurance Activity

The evaluator will 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_SSHS_EXT.1.4, and ensure that password-based authentication methods are also allowed.

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

- ***Test 2:*** *The evaluator shall choose one public key algorithm supported by the TOE. The evaluator shall generate a new key pair for that algorithm without configuring the TOE to recognize the public key for authentication. The evaluator shall use an SSH client to attempt to connect to the TOE with the new key pair and demonstrate that authentication fails.*
- ***Test 3:*** *Using the guidance documentation, the evaluator will configure the TOE to perform password-based authentication on a client, and demonstrate that a user can be successfully authenticated by the TOE using a password as an authenticator.*
- ***Test 4:*** *The evaluator shall use an SSH client, enter an incorrect password to attempt to authenticate to the TOE, and demonstrate that the authentication fails.*

FCS_SSHS_EXT.1.2 The SSH server shall ensure that, as described in RFC 4253, packets greater than [**assignment**: *number of bytes]* bytes in an SSH transport connection are dropped.

This requirement depends upon selection in FCS_SSH_EXT.1.1.

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 will check that the TSS describes how “large packets” in terms of RFC 4253 are detected and handled.
The evaluator will demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

FCS_SSHS_EXT.1.3 The SSH server shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, [**selection**: *aes128-cbc*, *aes256-cbc*, *AEAD_AES_128_GCM*, *AEAD_AES_256_GCM*, *no other algorithms*] .

This requirement depends upon selection in FCS_SSH_EXT.1.1.

Application Note: RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm.

Assurance Activity

The evaluator will 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 will check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component.
The evaluator will also check the guidance documentation 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).

- ***Test 1:*** *The evaluator will initiate an SSH connection using each of the encryption 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.*
- ***Test 2:*** *The evaluator will configure an SSH client to only propose the 3des-cbc encryption algorithm and no other encryption algorithms. The evaluator will attempt to establish an SSH connection from the client to the TOE server and observe that the connection is rejected.*

FCS_SSHS_EXT.1.4 The SSH server shall ensure that the SSH transport implementation uses [**selection**: *ssh-rsa*, *ecdsa-sha2-nistp256*] and [**selection**: *ecdsa-sha2-nistp384*, *x509v3-ecdsa-sha2-nistp256*, *x509v3-ecdsa-sha2-nistp256*, *no other public key algorithms*] as its public key algorithm(s) and rejects all other public key algorithms.

This requirement depends upon selection in FCS_SSH_EXT.1.1.

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 may remove ssh-rsa as a selection.
The SFRs for cryptographic key generation and certificate validation are inherited from the base PP.

Assurance Activity

The evaluator will check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the public key algorithms supported are specified as well. The evaluator will check the TSS to ensure that the public key algorithms specified are identical to those listed for this component.
The evaluator will also check the guidance documentation 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).

- ***Test 1:*** *Using an appropriately configured client, the evaluator will establish an SSH connection using each of the public key algorithms specified by the requirement to authenticate. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.*
- ***Test 2:*** *The evaluator will configure an SSH client to propose only the ssh-dsa public key algorithm and no other public key algorithms. Using this client, the evaluator will attempt to establish an SSH connection to the TOE and observe that the connection is rejected.*

FCS_SSHS_EXT.1.5 The SSH server shall ensure that the SSH transport implementation uses [**selection**: *hmac-sha1*, *hmac-sha1-96*, *hmac-sha2-256*, *hmac-sha2-512*] and [**selection**: *AEAD_AES_128_GCM*, *AEAD_AES_256_GCM*, *no other MAC algorithms*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

This requirement depends upon selection in FCS_SSH_EXT.1.1.

Application Note: RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH.
The SFRs for cryptographic operations, encryption and hashing, are inherited from the base PP.

Assurance Activity

The evaluator will 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 will also check the guidance documentation 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).

- ***Test 1:*** *Using an appropriately configured client, the evaluator will 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.*
- ***Test 2:*** *The evaluator will configure an SSH client to only propose the "none" MAC algorithm. Using this client, the evaluator will attempt to connect to the TOE and observe that the attempt fails.*
- ***Test 3:*** *The evaluator will configure an SSH client to only propose the hmac-md5 MAC algorithm. Using this client, the evaluator will attempt to connect to the TOE and observe that the attempt fails.*

FCS_SSHS_EXT.1.6 The SSH server shall ensure that [**selection**: *diffie-hellman-group14-sha1*, *ecdh-sha2-nistp256*] and [**selection**: *ecdh-sha2-nistp384*, *ecdh-sha2-nistp521*, *no other methods*] are the only allowed key exchange methods used for the SSH protocol.

This requirement depends upon selection in [FCS_SSH_EXT.1.1](#).

Assurance Activity

The evaluator will check the TSS to ensure that it lists the supported key exchange algorithms, and that that list corresponds to the list in this component.
The evaluator will also check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed key exchange algorithms are used in SSH connections to the TOE.

- ***Test 1:*** *For each of the allowed key exchange methods, the evaluator will configure an SSH client to propose only it and attempt to connect to the TOE and observe that each attempt succeeds.*
- ***Test 2:*** *The evaluator shall configure an SSH client to only allow the diffiehellman-group1-sha1 key exchange. The evaluator shall attempt to connect from the SSH client to the SSH Server and observe that the attempt fails.*

FCS_SSHS_EXT.1.7 The SSH server shall ensure that the SSH connection be rekeyed after [**selection**: *no more than* 2^{28} *packets have been transmitted*, *no more than 1 Gigabyte of data has been transmitted*, *no more than 1 hour*] using that key.

This requirement depends upon selection in [FCS_SSH_EXT.1.1](#).

Assurance Activity

- ***Test 1:*** *The evaluator will configure the TOE to create a log entry when a rekey occurs. The evaluator will connect to the TOE with an SSH* *client and cause a rekey to occur according to the selection(s) in the ST, and subsequently review the audit log to ensure that a rekey occurred.*

C. Objective Requirements

This Annex includes requirements that specify security functionality which also addresses threats. The requirements are not currently mandated in the body of this EP as they describe security functionality not yet widely-available in commercial technology. However, these requirements may be included in the ST such that the product is still conformant to this EP, and it is expected that they be included as soon as possible.
None exists presently.

D. References

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and General Model, CCMB-2012-09-001, Version 3.1 Revision 4, September 2012. • Part 2: Security Functional Components, CCMB-2012-09-002, Version 3.1 Revision 4, September 2012. • Part 3: Security Assurance Components, CCMB-2012-09-003, Version 3.1 Revision 4, September 2012.
[GPOSPP]	Protection Profile for General Purpose Operating Systems
[MDMPP]	Protection Profile for Mobile Device Management
[AppPP]	Protection Profile for Application Software

E. Acronyms

Acronym	Meaning
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
ECDSA	Elliptic Curve Digital Signature Algorithm
GCM	Galois/Counter Mode
IETF	Internet Engineering Task Force
IV	Initialization Vector
MAC	Message Authentication Code
NIST	National Institute of Standards and Technology
PBKDF	Password-Based Key Derivation Function
RFC	Request for Comment (IETF)
RSA	Rivest Shamir Adelman
SSH	Secure Shell