Functional Package for Secure Shell (SSH)

Version: 2.0
2024-12-12
National Information Assurance Partnership

Revision History

Version	Date	Comment
1.0	2021-05-13	Converted SSH EP to a Functional Package and incorporated CCUF CWG input.
2.0	2024-12-12	Updated for CC:2022 conformance, incorporated applicable errata.

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.

This *Functional Package (FP) for Secure Shell* provides a collection of SSH protocol related Security Functional Requirements (SFRs) and Evaluation Activities (EAs) covering audit, authentication, cryptographic algorithms, and protocol negotiation. The intent of this package is to provide Protection Profile (PP), collaborative Protection Profile (cPP), and Protection Profile Module (PP-Module) authors with a readily consumable collection of SFRs and EAs to be integrated into their documents.

The functional components defined for this package were chosen to ensure that a conformant TOE implements SSH in a secure manner by requiring that the TSF implement protocol-specific details that are not captured in the FTP_PRO SFR family.

1.2 Terms

The following sections list Common Criteria and technology terms used in this document.

1.2.1 Common Criteria Terms

Assurance	Grounds for confidence that a TOE meets the SFRs [CC].
Base Protection Profile (Base-PP)	Protection Profile used as a basis to build a PP-Configuration.
Collaborative Protection Profile (cPP)	A Protection Profile developed by international technical communities and approved by multiple schemes.
Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408).
Common Criteria Testing Laboratory	Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Distributed TOE	A TOE composed of multiple components operating as a logical whole.
Functional Package (FP)	A document that collects SFRs for a particular protocol, technology, or functionality.
Operational Environment (OE)	Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.
Protection Profile Configuration (PP-Configuration)	A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module.
Protection Profile Module (PP-Module)	An implementation-independent statement of security needs for a TOE type complementary to one or more Base-PPs.
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.

Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
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 an ST.

1.2.2 Technical Terms

Connection	The SSH transport layer between a client and a server. Within a connection there can be multiple sessions.
Rekey	Where the connection renegotiates the shared secret and each session subsequently derives a new encryption key.
Secure Shell (SSH)	Cryptographic network protocol for initiating text-based shell sessions on remote systems.
Session	A discrete stream of data within a connection.

1.3 Compliant Targets of Evaluation

The TOE in this FP is a product that acts as an SSH client, SSH server, or both. This FP describes the extended security functionality of SSH in terms of [CC].

The contents of this FP must be appropriately incorporated into a PP, cPP, or PP-Module. When this package is incorporated as such, the ST must include selection-based requirements in accordance with the selections or assignments indicated in the incorporating document.

The PP, cPP, or PP-Module that instantiates this Package must typically include the following components in order to satisfy dependencies of this Package. It is the responsibility of the PP, cPP, or PP-Module author who incorporates this FP to ensure that dependence on these components is satisfied, either by the TOE or by assumptions about its OE.

An ST must identify the applicable version of the PP, cPP, or PP-Module, and of this FP in its conformance claims.

Component	Explanation
FCS_CKM.1	To support key generation for SSH, the PP or PP-Module must include FCS_CKM.1 and specify the corresponding algorithms.
FCS_CKM.2	To support key establishment for SSH, the PP or PP-Module must include FCS_CKM.2 and specify the corresponding algorithms.
FCS_COP.1	To support the cryptography needed for SSH communications, the PP or PP-Module must include FCS_COP.1 (iterating as needed) to specify AES with corresponding key sizes and modes, digital signature generation and verification function (at least one of RSA or ECDSA), a cryptographic hash function, and a keyed-hash message authentication function. In particular, the incorporating document must support AES-GCM as defined in NIST SP 800-38D with key sizes of 256 bits.
FCS_RBG.1	To support random bit generation needed for SSH key generation, the PP or PP-Module must include FCS_RBG.1 or an extended SFR that defines comparable functionality.
FIA_X509_EXT.1	To support establishment of SSH communications using a public key algorithm that includes X.509, the PP or PP-Module must include FIA_X509_EXT.1. Note however that support for X.509 is selectable and not mandatory.

FIA_X509_EXT.2	To support establishment of SSH communications using a public key algorithm that includes X.509, the PP or PP-Module must include FIA_X509_EXT.2. Note however that support for X.509 is selectable and not mandatory.
FPT_STM.1	To support establishment of SSH communications using a public key algorithm that includes X.509, the PP or PP-Module must include FPT_STM.1 or some other requirement that ensures reliable system time. Note however that support for time-based rekey thresholds is selectable and not mandatory.

2 Conformance Claims

Conformance Statement

An ST must claim exact conformance to this Functional Package.

The evaluation methods used for evaluating the TOE are a combination of the workunits defined in [CEM] as well as the Evaluation Activities for ensuring that individual SFRs and SARs have a sufficient level of supporting evidence in the Security Target and guidance documentation and have been sufficiently tested by the laboratory as part of completing ATE_IND.1. Any functional packages this PP claims similarly contain their own Evaluation Activities that are used in this same manner.

CC Conformance Claims

This Functional Package is conformant to Part 2 (extended) of Common Criteria CC:2022, Revision 1 as corrected and interpreted in [ERR], Version 1.1.

PP Claim

This Functional Package does not claim conformance to any Protection Profile.

There are no PPs or PP-Modules that are allowed in a PP-Configuration with this Functional Package.

Package Claim

This Functional Package is not conformant to any Functional or Assurance Packages.

3 Security Functional Requirements

This chapter describes the security requirements which have to be fulfilled by the product under evaluation. Those requirements comprise functional components from Part 2 of [CC]. The following conventions are used for the completion of operations:

- **Refinement** operation (denoted by **bold text** or ~~strikethrough text~~): Is used to add details to a requirement or to remove part of the requirement that is made irrelevant through the completion of another operation, and thus further restricts a requirement.
- **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: Is indicated by appending the SFR name with a slash and unique identifier suggesting the purpose of the operation, e.g. "/EXAMPLE1."

3.1 Auditable Events for Mandatory SFRs

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

Table 1: Auditable Events for Mandatory Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FCS_SSH_EXT.1	[**selection, choose one of**: *Failure to establish SSH connection*, *none*]	[**selection, choose one of**: *Reason for failure and non-TOE endpoint of attempted connection (IP Address)*, *None*]
	[**selection, choose one of**: *Establishment of SSH connection*, *none*]	[**selection, choose one of**: *Non-TOE endpoint of connection (IP Address)*, *None*]
	[**selection, choose one of**: *Termination of SSH connection session*, *none*]	[**selection, choose one of**: *Non-TOE endpoint of connection (IP Address)*, *None*]
	[**selection, choose one of**: *Dropping of packets outside defined size limits*, *none*]	[**selection, choose one of**: *Packet size*, *None*]

3.2 Cryptographic Support (FCS)

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1

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

Application Note: The following mapping is provided as a guide to ST authors to ensure the appropriate RFC selections are made based on applicable selections in subsequent SFRs.

- RFC 4256: Select for keyboard-interactive authentication
- RFC 4344: Select for AES-256-CTR
- RFC 5647: Select for AEAD_AES_256_GCM or aes256-gcm@openssh.com
- RFC 5656: Select for elliptic curve cryptography (ECC)
- RFC 6187: Select for X.509 certificate use
- RFC 6668: Select for HMAC-SHA-2 algorithms
- RFC 8268: Select for FFC DH groups with SHA-2
- RFC 8308: Select if RFC 8332 is selected
- RFC 8332: Select if SHA-2 is available with ssh-rsa

The ST author selects 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 CC. The RFC selections for this requirement must be consistent with selections in later elements of this FP (e.g., cryptographic algorithms permitted).

For the purposes of this package (and subsequent integration into cPPs), only the claimed algorithms listed in the package must be enabled for use.

RFC 4251 defines support for the general implementation of the SSH protocol.

RFC 4252 defines support for the required SSH authentication method.

RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED." This means that from the Internet Engineering Task Force's perspective, the implementation must include support, not that the algorithms must be enabled for use. For the purposes of this SFR's EA and this FP overall, it is not necessary to ensure that algorithms listed as "REQUIRED" by the RFC but not listed in later elements of this FP are actually implemented.

RFC 4254 defines support for the general implementation of the SSH connection protocol.

RFC 4256 must be selected if "keyboard-interactive" is selected in FCS_SSH_EXT.1.2.

RFC 4344 must be selected if aes256-ctr is selected in FCS_SSH_EXT.1.4.

RFC 5647 must be selected when AEAD_AES_256_GCM or aes256-gcm@openssh.com is selected as an encryption algorithm in FCS_SSH_EXT.1.4 and when AEAD_AES_256_GCM is selected as a MAC algorithm in FCS_SSH_EXT.1.5.

RFC 5656 must be selected when ecdsa-sha2-nistp384 or ecdsa-sha2-nistp521 is selected as a public key algorithm in FCS_SSH_EXT.1.2, or when ecdh-sha2-nistp384 or ecdh-sha2-nistp521 is selected as a key exchange algorithm in FCS_SSH_EXT.1.6, or when "RFC 5656" is selected in FCS_SSH_EXT.1.7.

RFC 6187 must be selected when x509v3-ecdsa-sha2-nistp384 or x509v3-ecdsa-sha2-nistp521 is selected as a public key algorithm in FCS_SSH_EXT.1.2.

RFC 6668 must be selected when hmac-sha2-512 is selected as a MAC algorithm in FCS_SSH_EXT.1.5.

RFC 8268 must be selected when diffie-hellman-group15-sha512, diffie-hellman-group16-sha512, diffie-hellman-group17-sha512, or diffie-hellman-group18-sha512 is selected as a key exchange algorithm in FCS_SSH_EXT.1.6.

RFC 8308 defines support for secure negotiation of protocol extensions, and must be claimed when RFC 8332 is claimed.

RFC 8332 must be selected when rsa-sha2-512 is selected as a public key algorithm in FCS_SSH_EXT.1.2.

If "client" is selected, then the ST must include FCS_SSHC_EXT.1.

If "server" is selected, then the ST must include FCS_SSHS_EXT.1.

FCS_SSH_EXT.1.2

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [**selection**:

- *password, complying with [**selection**: RFC 4252, RFC 4256 keyboard-interactive methods]*
- *“publickey” (RFC 4252): [**selection**:*
  - *rsa-sha2-512 (RFC 8332)*
  - *ecdsa-sha2-nistp384 (RFC 5656)*
  - *ecdsa-sha2-nistp521 (RFC 5656)*
  - *x509v3-ecdsa-sha2-nistp384 (RFC 6187)*
  - *x509v3-ecdsa-sha2-nistp521 (RFC 6187)*

  ]

] and no other methods.

Application Note: Within SSH there are two types of authentication: user authentication and peer authentication. This SFR deals with the options supported for user authentication. Peer authentication is covered in FCS_SSHC_EXT.1.1 (for clients) and FCS_SSHS_EXT.1.1 (for servers).

FCS_SSH_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [**assignment**: *number of bytes between 35 KB and 1 GB (inclusive)*] in an SSH transport connection are dropped.

Application Note: RFC 4253, Section 6.1 provides for the acceptance of “large packets” with the caveat that the packets should be of “reasonable length” or dropped. The ST author should fill in the assignment with the maximum packet size accepted, thus defining "reasonable length for the TOE.

The upper bound on the packet size is driven by the size identified in FCS_SSH_EXT.1.8.

FCS_SSH_EXT.1.4

The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [**selection**:

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

] and no other mechanisms.

Application Note: As described in RFC 5647, AEAD_AES_256_GCM needs the corresponding MAC algorithm to be selected in FCS_SSH_EXT.1.5.

FCS_SSH_EXT.1.5

The TSF shall protect data in transit from modification, deletion, and insertion using: [**selection**:

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

] and no other mechanisms.

Application Note: As described in RFC 5647, AEAD_AES_256_GCM needs the corresponding encryption algorithm to be selected. In AES-GCM mode, integrity is not provided using a MAC, it is implicit in the AES-GCM mode itself. There is no need for a corresponding FCS_COP element. The FCS_COP element for AES would already cover this.

If the negotiated encryption algorithm is aes256-gcm@openssh.com algorithms, then the MAC field is ignored during negotiation and AES-GCM is implicitly selected for the MAC. The selection “implicit” is not an SSH identifier and will not be seen on the wire; however, the negotiated MAC might be decoded as “implicit.”

FCS_SSH_EXT.1.6

The TSF shall establish a shared secret with its peer using: [**selection**:

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

] and no other mechanisms.

Application Note: The values "strict-kex-c-v00@openssh.com" and "strict-kex-s-v00@openssh.com" may also be present in the key exchange offering. These values are used to protect against SSH prefix data truncation (i.e. Terrapin attack). These values are used to ensure that key exchange proceeds in an appropriate order; they are not exchange methods in and of themselves.

FCS_SSH_EXT.1.7

The TSF shall use an SSH key derivation function (KDF) as defined in [**selection**:

- *RFC 4253, Section 7.2*
- *RFC 5656, Section 4*

] to derive the following cryptographic keys from a shared secret: session keys.

Application Note: RFC 4253 must be selected when the key establishment scheme (selected in FCS_SSH_EXT.1.6) uses finite field cryptography (FFC) and RFC 5656 when it uses ECC.

RFC 4253, Section 7.2 defines two KDFs for FFC-based key establishment schemes. Therefore, RFC 4253 should be selected if any of the RFC 4253 or RFC 8268 key establishment schemes are selected.

FCS_SSH_EXT.1.8

The TSF shall ensure that [**selection**:

- *a rekey of the session keys*
- *connection termination*

] occurs when any of the following thresholds are met:

- [**assignment**: *length of time lesser than or equal to one hour*] connection time
- no more than [**assignment**: *number of bytes less than or equal to one gigabyte*] of transmitted data, or
- no more than [**assignment**: *number of bytes less than or equal to one gigabyte*] of received data.

Application Note: This SFR defines three thresholds that need to be implemented. These thresholds were arrived at to ensure that the cryptographic key space for the symmetric session keys is not exhausted (more detail can be found in RFC 4344 and RFC 4253). A rekey or connection termination needs to be performed whenever a threshold is reached for a given connection. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.

It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR. If a threshold is configurable, the guidance documentation needs to specify how to configure that threshold.

It is possible that hardware limitations may prevent reaching the data transfer threshold in less than one hour. In cases where the data transfer threshold could not be reached due to hardware limitations, it is acceptable to omit testing of this (SSH rekeying based on data transfer threshold). See EAs for details.

Evaluation Activities

FCS_SSH_EXT.1.1
TSS
The evaluator shall ensure that the selections indicated in the ST are consistent with selections in this element and subsequent element. Otherwise, this SFR is evaluated by activities for other SFRs.

Guidance
There are no guidance EAs for this element. This SFR is evaluated by activities for other SFRs.

Tests
There are no test EAs for this element. This SFR is evaluated by activities for other SFRs.

FCS_SSH_EXT.1.2
TSS
The evaluator shall check to ensure that:

- *the authentication methods listed in the TSS are identical to those claimed in this element;*
- *if password-based authentication methods have been claimed in the ST, then the TSS also describes these*

Guidance
The evaluator shall check the guidance documentation to ensure the configuration options, if any, for authentication mechanisms provided by the TOE are described.

Tests

- *Test FCS_SSH_EXT.1.2:1: [conditional] If the TOE is acting as an SSH server:*
    a. *The evaluator shall use a suitable SSH client to connect to the TOE, enable debug messages in the SSH client, and examine the debug messages to determine that only the configured authentication methods for the TOE were offered by the server.*
    b. *[conditional] If the SSH server supports X.509-based client authentication options:*
        i. *The evaluator shall initiate an SSH session from a client where the username is associated with the X.509 certificate. The evaluator shall verify the session is successfully established.*
        ii. *Next, the evaluator shall use the same X.509 certificate as above, but include a username not associated with the certificate. The evaluator shall verify that the session does not establish.*
        iii. *Finally, the evaluator shall use the correct username (from step a above), but use a different X.509 certificate which is not associated with the username. The evaluator shall verify that the session does not establish.*
    c. *If the TOE supports keyboard-interactive password authentication, the evaluator shall verify that the TOE issues an authentication challenge to the SSH client for each supported method when the client specifies the use of a keyboard-interactive method.*
    d. *For each supported authentication method, the evaluator shall verify that an SSH client can be used to establish a session with the TOE while using the supported method.*
    e. *For each supported authentication method, the evaluator shall use an SSH client provided with bad authentication data (e.g., incorrectly generated certificate or incorrect password) and verify that the connection is rejected.*
- *Test FCS_SSH_EXT.1.2:2: [conditional] If the TOE is acting as an SSH client, the evaluator shall initiate an SSH session using each supported authentication method and verify that the session is successfully established. Specifically, if password authentication is supported, then the evaluator shall use interactive, non-interactive, or both, depending on the claims made in FCS_SSH_EXT.1.2. If public key authentication is supported, then the evaluator*

shall use each supported algorithm claimed in FCS_SSH_EXT.1.2.

- *Test FCS_SSH_EXT.1.2:3: [conditional] If the TOE is acting as an SSH client, the evaluator shall verify that the connection fails upon configuration mismatch as follows:*
  a. *The evaluator shall configure the client with an authentication method not supported by the server.*
  b. *The evaluator shall verify that the connection fails.*

  If the client supports only one authentication method, the evaluator can test this failure of connection by configuring the server with an authentication method not supported by the client. In order to facilitate this test, it is acceptable for the evaluator to configure an authentication method that is outside of the selections in the SFR.

FCS_SSH_EXT.1.3

TSS

The evaluator shall check that the TSS describes how "large packets" are detected and handled.

Guidance

There are no guidance EAs for this element.

Tests

- *Test FCS_SSH_EXT.1.3:1: The evaluator shall demonstrate that the TOE accepts the maximum allowed packet size.*
- *Test FCS_SSH_EXT.1.3:2: This test is performed to verify that the TOE drops packets that are larger than the size specified in the element.*
  a. *The evaluator shall establish a successful SSH connection with the peer.*
  b. *Next, the evaluator shall craft a packet that is slightly larger than the maximum size specified in this element and send it through the established SSH connection to the TOE. The packet should not be greater than the maximum packet size plus 16 bytes. If the packet is larger, the evaluator shall justify the need to send a larger packet.*
  c. *The evaluator shall verify that the packet was dropped by the TOE. The method of verification will vary by the TOE. Examples include reviewing the TOE audit log for a dropped packet audit or observing that the TOE terminates the connection.*

FCS_SSH_EXT.1.4

TSS

The evaluator shall check the description of the implementation of SSH in the TSS to ensure the encryption algorithms supported are specified. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this element.

Guidance

The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Tests

The evaluator shall perform the following tests.

If the TOE can be both a client and a server, these tests must be performed for both roles.

- *Test FCS_SSH_EXT.1.4:1: The evaluator shall ensure that only claimed algorithms and cryptographic primitives are used to establish an SSH connection. To verify this, the evaluator shall establish an SSH connection with a remote endpoint. The evaluator shall capture the traffic exchanged between the TOE and the remote endpoint during protocol negotiation (e.g., using a packet capture tool or information provided by the endpoint, respectively). The evaluator shall verify from the captured traffic that the TOE offers only the algorithms defined in the ST for the TOE for SSH connections. The evaluator shall perform one successful negotiation of an SSH connection and verify that the negotiated algorithms were included in the advertised set. If the evaluator detects that not all algorithms defined in the ST for SSH are advertised by the TOE or the TOE advertises additional algorithms not defined in the ST for SSH, the test shall be regarded as failed.*

  The data collected from the connection above shall be used for verification of the advertised hashing and shared secret establishment algorithms in FCS_SSH_EXT.1.5 and FCS_SSH_EXT.1.6 respectively.
- *Test FCS_SSH_EXT.1.4:2: For the connection established in Test 1, the evaluator shall terminate the connection and observe that the TOE terminates the connection.*
- *Test FCS_SSH_EXT.1.4:3: The evaluator shall configure the remote endpoint to only allow a mechanism that is not included in the ST selection. The evaluator shall attempt to connect to the TOE and observe that the attempt fails.*

FCS_SSH_EXT.1.5

TSS

The evaluator shall check the description of the implementation of SSH in the TSS to ensure the hashing algorithms supported are specified. The evaluator shall check the TSS to ensure that the hashing algorithms specified are identical to those listed for this element.

Guidance
The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Tests

- *Test FCS_SSH_EXT.1.5:1: The evaluator shall use the test data collected in FCS_SSH_EXT.1.4, Test 1 to verify that appropriate mechanisms are advertised.*
- *Test FCS_SSH_EXT.1.5:2: The evaluator shall configure an SSH peer to allow only a hashing algorithm that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection and observe that the connection is rejected.*

FCS_SSH_EXT.1.6
TSS
The evaluator shall check the description of the implementation of SSH in the TSS to ensure the shared secret establishment algorithms supported are specified. The evaluator shall check the TSS to ensure that the shared secret establishment algorithms specified are identical to those listed for this element.

Guidance
The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Tests

- *Test FCS_SSH_EXT.1.6:1:*

  The evaluator shall use the test data collected in FCS_SSH_EXT.1.4, Test 1 to verify that appropriate mechanisms are advertised.

  Note that it is permissible for the TOE to advertise the "strict-kex-c-v00@openssh.com" or "strict-kex-s-v00@openssh.com" value alongside the other key exchange mechanisms. These values are associated with client and server, respectively.
- *Test FCS_SSH_EXT.1.6:2: The evaluator shall configure an SSH peer to allow only a key exchange method that is not included in the ST selection. The evaluator shall attempt to establish an SSH connection and observe that the connection is rejected.*
- *Test FCS_SSH_EXT.1.6:3: The evaluator shall configure the SSH peer to omit the strict-kex value from the kex_algorithms field. The evaluator shall verify that the TOE is able to successfully establish a connection with the SSH peer.*
- *Test FCS_SSH_EXT.1.6:4: The evaluator shall configure the SSH peer to include the strict-kex value in the kex_algorithms field. The evaluator shall verify that the TOE is able to successfully establish a connection with the SSH peer.*

FCS_SSH_EXT.1.7
TSS
The evaluator shall check the description of the implementation of SSH in the TSS to ensure the supported KDFs are specified. The evaluator shall check the TSS to ensure that the KDFs specified are identical to those listed for this element.

Guidance
There are no guidance EAs for this element.

Tests
There are no test EAs for this element.

FCS_SSH_EXT.1.8
TSS
The evaluator shall check the TSS to ensure that if the TOE enforces connection rekey or termination limits lower than the maximum values, that these lower limits are identified.

In cases where hardware limitation will prevent reaching the data transfer threshold in less than one hour, the evaluator shall check the TSS to ensure it contains:

a. *An argument describing this hardware-based limitation and*
b. *Identification of the hardware components that form the basis of such argument.*

For example, if a specific Ethernet Controller or Wi-Fi radio chip is the root cause of such limitation, these subsystems shall be identified.

Guidance
The evaluator shall check the guidance documentation to ensure that if the connection rekey or termination limits are configurable, it contains instructions to the administrator on how to configure the relevant connection rekey or termination limits for the TOE.

Tests

The evaluator shall ensure that the test harness is configured so that its connection rekey or termination limits are greater than the limits supported by the TOE. It is expected that the test harness should not be initiating the connection rekey or termination.

- *Test FCS_SSH_EXT.1.8:1: The evaluator shall establish an SSH connection, wait until the identified connection rekey limit is met, and observe that a connection rekey or termination is initiated. This may require traffic to periodically be sent or a keep-alive setting for the connection to be applied to ensure that the connection is not closed due to an idle timeout.*
- *Test FCS_SSH_EXT.1.8:2: The evaluator shall establish an SSH connection, transmit data from the TOE until the identified connection rekey or termination limit is met, and observe that a connection rekey or termination is initiated.*
- *Test FCS_SSH_EXT.1.8:3: The evaluator shall establish an SSH connection, send data to the TOE until the identified connection rekey limit or termination is met, and observe that a connection rekey or termination is initiated.*

Appendix A - Optional Requirements

As indicated in the introduction to this Functional Package, the baseline requirements (those that must be performed by the TOE) are contained in the body of this Functional Package. This appendix contains three other types of optional requirements:

The first type, defined in Appendix A.1 Strictly Optional Requirements, are strictly optional requirements. If the TOE meets any of these requirements the vendor is encouraged to claim the associated SFRs in the ST, but doing so is not required in order to conform to this Functional Package.

The second type, defined in Appendix A.2 Objective Requirements, are objective requirements. These describe security functionality that is not yet widely available in commercial technology. Objective requirements are not currently mandated by this Functional Package, but will be mandated in the future. Adoption by vendors is encouraged, but claiming these SFRs is not required in order to conform to this Functional Package.

The third type, defined in Appendix A.3 Implementation-dependent Requirements, are Implementation-dependent requirements. If the TOE implements the product features associated with the listed SFRs, either the SFRs must be claimed or the product features must be disabled in the evaluated configuration.

A.1 Strictly Optional Requirements

This Functional Package does not define any Strictly Optional requirements.

A.2 Objective Requirements

This Functional Package does not define any Objective requirements.

A.3 Implementation-dependent Requirements

This Functional Package does not define any Implementation-dependent requirements.

Appendix B - Selection-based Requirements

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

B.1 Auditable Events for Selection-based Requirements

The auditable events in the table below are included in a Security Target if both the associated requirement is included and the incorporating PP or PP-Module supports audit event reporting through FAU_GEN.1 and any other criteria in the incorporating PP or PP-Module are met.

Table 2: Auditable Events for Selection-based Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FCS_SSHC_EXT.1	No events specified	N/A
FCS_SSHS_EXT.1	No events specified	N/A

B.2 Cryptographic Support (FCS)

FCS_SSHC_EXT.1 SSH Client Protocol

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

FCS_SSHC_EXT.1.1

The TSF shall authenticate its peer (SSH server) using: [**selection**:

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

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

  ]

] as described in RFC 4251, Section 4.1.

Application Note: The local database may be implemented using any equivalent local storage mechanism.
Validation of X.509 certificates is expected to conform to the Functional Package for X.509.

Evaluation Activities

FCS_SSHC_EXT.1
TSS
There are no TSS EAs for this component.

Guidance
The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Tests
The evaluator shall perform the following tests:

- *Test FCS_SSHC_EXT.1:1: [conditional] If using a local database by associating each host name with its corresponding public key, the evaluator shall configure the TOE with only a single host name and corresponding public key in the local database. The evaluator shall verify that the TOE can successfully connect to the host identified by the host name.*

- *Test FCS_SSHC_EXT.1:2: [conditional] If using a local database by associating each host name with its corresponding public key, the evaluator shall configure the TOE with only a single host name and non-corresponding public key in the local database. The evaluator shall verify that the TOE fails to connect to the configured host due to a public key mismatch.*
- *Test FCS_SSHC_EXT.1:3: [conditional] If using a local database by associating each host name with its corresponding public key, the evaluator shall try to connect to a host not configured in the local database. The evaluator shall verify that the TOE either fails to connect to a host identified by the host name or there is a prompt provided to store the public key in the local database.*
- *Test FCS_SSHC_EXT.1:4: [conditional] If using a list of trusted certification authorities, the evaluator shall configure the TOE with only a single trusted certification authority corresponding to the host's certificate. The evaluator shall verify that the TOE can successfully connect to the host identified by the host name.*
- *Test FCS_SSHC_EXT.1:5: [conditional] If using a list of trusted certification authorities, the evaluator shall configure the TOE with only a single trusted certification authority that does not correspond to the host's certificate. The evaluator shall verify that the TOE fails to connect to the host identified by the host name.*
- *Test FCS_SSHC_EXT.1:6: [conditional] If using a list of trusted certification authorities, the evaluator shall configure the test server with a certificate that does not belong to it, but is otherwise valid (i.e., it has an invalid reference identifier). This certificate shall be signed by a certificate authority that is one of the trusted certificate authorities configured on the TOE. The evaluator shall verify that the TOE fails to connect to the test server.*

FCS_SSHS_EXT.1 SSH Server Protocol

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

FCS_SSHS_EXT.1.1

The TSF shall authenticate itself to its peer (SSH client) using: [**selection**:

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

].

Application Note: These requirements relate to the server authenticating to the client. The client authenticating to the server is covered in FCS_SSHC_EXT.1.1.

Validation of X.509 certificates is expected to conform to the Functional Package for X.509.

Evaluation Activities

FCS_SSHS_EXT.1
TSS
There are no TSS EAs for this component.

Guidance
The evaluator shall check the guidance documentation to ensure that it contains instructions to the administrator on how to ensure that only the allowed mechanisms are used in SSH connections with the TOE.

Tests
The evaluator shall perform the following tests:

- *Test FCS_SSHS_EXT.1:1: The evaluator shall use a suitable SSH client to connect to the TOE and examine the list of server host key algorithms in the SSH_MSG_KEXINIT packet sent from the server to the client to determine that only the configured server authentication methods for the TOE were offered by the server.*
- *Test FCS_SSHS_EXT.1:2: The evaluator shall test for a successful configuration setting of each server authentication method as follows. The evaluator shall initiate an SSH session using the authentication method configured and verify that the session is successfully established. Repeat this process for each independently configurable server authentication method supported by the server.*
- *Test FCS_SSHS_EXT.1:3: The evaluator shall configure the peer to only allow an authentication mechanism that is not included in the ST selection. The evaluator shall*

attempt to connect to the TOE and observe that the TOE sends a disconnect message.

Appendix C - Extended Component Definitions

This appendix contains the definitions for all extended requirements specified in the Functional Package.

C.1 Extended Components Table

All extended components specified in the Functional Package are listed in this table:

Table 3: Extended Component Definitions

Functional Class	Functional Components
Cryptographic Support (FCS)	FCS_SSHC_EXT SSH Client Protocol FCS_SSHS_EXT SSH Server Protocol FCS_SSH_EXT SSH Protocol

C.2 Extended Component Definitions

C.2.1 Cryptographic Support (FCS)

This Functional Package defines the following extended components as part of the FCS class originally defined by CC Part 2:

C.2.1.1 FCS_SSH_EXT SSH Protocol

Family Behavior

This family defines requirements for implementation of the SSH protocol that goes beyond the level of detail specified for trusted communications in CC Part 2.

Component Leveling

FCS_SSH_EXT ——— 1

FCS_SSH_EXT.1, SSH Protocol, requires the TSF to specify the details of its SSH protocol implementation.

Management: FCS_SSH_EXT.1

No specific management functions are identified.

Audit: FCS_SSH_EXT.1

The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP, PP-Module, FP, or ST:

- Failure to establish SSH connection
- Establishment of SSH connection
- Termination of SSH connection
- Dropping of packets outside defined size limits

FCS_SSH_EXT.1 SSH Protocol

Hierarchical to: No other components.

Dependencies to: FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Derivation
FCS_COP.1 Cryptographic Operation
FCS_RBG.1 Random Bit Generation

FCS_SSH_EXT.1.1

The TOE shall implement SSH acting as a [**selection**: *client*, *server*] that complies with RFCs 4251, 4252, 4253, 4254, [**assignment**: *other RFCs*] and [**assignment**: *other standards*].

FCS_SSH_EXT.1.2

The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [**assignment**: *supported authentication methods*] and no other methods.

FCS_SSH_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [**assignment**: *number of bytes between 35 KB and 1 GB (inclusive)*] in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4

The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [**assignment**: *confidentiality mechanisms*] and no other mechanisms.

FCS_SSH_EXT.1.5

The TSF shall protect data in transit from modification, deletion, and insertion using: [**assignment**: *data integrity mechanisms*] and no other mechanisms.

FCS_SSH_EXT.1.6

The TSF shall establish a shared secret with its peer using: [**assignment**: *key agreement mechanisms*] and no other mechanisms.

FCS_SSH_EXT.1.7

The TSF shall use an SSH key derivation function (KDF) as defined in [**selection**:

- *RFC 4253, Section 7.2*
- *RFC 5656, Section 4*

] to derive the following cryptographic keys from a shared secret: session keys.

FCS_SSH_EXT.1.8

The TSF shall ensure that [**selection**:

- *a rekey of the session keys*
- *connection termination*

] occurs when any of the following thresholds are met:

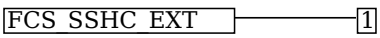
- [**assignment**: *length of time lesser than or equal to one hour*] connection time
- no more than [**assignment**: *number of bytes less than or equal to one gigabyte*] of transmitted data, or
- no more than [**assignment**: *number of bytes less than or equal to one gigabyte*] of received data.

C.2.1.2 FCS_SSHC_EXT SSH Client Protocol

Family Behavior

This family defines requirements for implementation of the SSH protocol when the TSF is acting as a client.

Component Leveling



FCS_SSHC_EXT.1, SSH Client Protocol, requires the TSF to specify the details of its SSH client implementation.

Management: FCS_SSHC_EXT.1

No specific management functions are identified.

Audit: FCS_SSHC_EXT.1

There are no auditable events foreseen.

FCS_SSHC_EXT.1 SSH Client Protocol

Hierarchical to: No other components.

Dependencies to: FCS_SSH_EXT.1 SSH Protocol

FCS_SSHC_EXT.1.1

The TSF shall authenticate its peer (SSH server) using: [**assignment**: *peer authentication method*] as described in RFC 4251, Section 4.1.

C.2.1.3 FCS_SSHS_EXT SSH Server Protocol

Family Behavior

This family defines requirements for implementation of the SSH protocol when the TSF is acting as a server.

Component Leveling

FCS_SSHS_EXT.1, SSH Server Protocol, requires the TSF to specify the details of its SSH server implementation.

Management: FCS_SSHS_EXT.1

No specific management functions are identified.

Audit: FCS_SSHS_EXT.1

There are no auditable events foreseen.

FCS_SSHS_EXT.1 SSH Server Protocol

Hierarchical to: No other components.

Dependencies to: FCS_SSH_EXT.1 SSH Protocol

FCS_SSHS_EXT.1.1

The TSF shall authenticate itself to its peer (SSH client) using: [**assignment**: *peer authentication method*].

Appendix D - Acronyms

Table 4: Acronyms

Acronym	Meaning
Base-PP	Base Protection Profile
CC	Common Criteria
CEM	Common Evaluation Methodology
cPP	Collaborative Protection Profile
EA	Evaluation Activity
ECC	Elliptic Curve Cryptography
FP	Functional Package
KDF	Key Derivation Function
OE	Operational Environment
PP	Protection Profile
PP-Configuration	Protection Profile Configuration
PP-Module	Protection Profile Module
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SSH	Secure Shell
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification

Appendix E - Bibliography

Table 5: Bibliography

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and general model, CCMB-2022-11-001, CC:2022, Revision 1, November 2022. • Part 2: Security functional requirements, CCMB-2022-11-002, CC:2022, Revision 1, November 2022. • Part 3: Security assurance requirements, CCMB-2022-11-003, CC:2022, Revision 1, November 2022. • Part 4: Framework for the specification of evaluation methods and activities, CCMB-2022-11-004, CC:2022, Revision 1, November 2022. • Part 5: Pre-defined packages of security requirements, CCMB-2022-11-005, CC:2022, Revision 1, November 2022.
[CEM]	Common Methodology for Information Technology Security Evaluation - • Evaluation methodology, CCMB-2022-11-006, CC:2022, Revision 1, November 2022.
[ERR]	Errata and Interpretation for CC:2022 (Release 1) and CEM:2022 (Release 1), Version 1.1, CCMB-2024-02-002, 22 July 2024.
[CEM]	Common Methodology for Information Technology Security - Evaluation Methodology, CCMB-2022-11-006, CEM:2022, Revision 1, November 2022.
[RFC 4251]	The Secure Shell (SSH) Protocol Architecture
[RFC 4252]	The Secure Shell (SSH) Authentication Protocol
[RFC 4253]	The Secure Shell (SSH) Transport Layer Protocol
[RFC 4254]	The Secure Shell (SSH) Connection Protocol
[RFC 4256]	Generic Message Exchange Authentication for the Secure Shell Protocol (SSH)
[RFC 4344]	The Secure Shell (SSH) Transport Layer Encryption Modes
[RFC 5647]	AES Galois Counter Mode for the Secure Shell Transport Layer Protocol
[RFC 5656]	Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer
[RFC 6187]	X.509v3 Certificates for Secure Shell Authentication
[RFC 6668]	SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol
[RFC 8268]	More Modular Exponentiation (MODP) Diffie-Hellman (DH) Key Exchange (KEX) Groups for Secure Shell (SSH)
[RFC 8308]	Extension Negotiation in the Secure Shell (SSH) Protocol
[RFC 8332]	Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol
[openssh-portable/ PROTOCOL]	OpenSSH's deviations and extensions (1.6 transport: AES-GCM)