Huawei NetEngine 8000 Series Routers' Software Security Target

Date: 2023-05-25

Created by 

Change History

Version	Date	Author	Comment
1.4	5/25/2023	Chenpeng	Public Version

Table of contents

1 ST Introduction

1.1 ST Reference

Title: Huawei NetEngine 8000 Series Routers' Software Security Target

Version: v1.4

Author: Huawei

Evaluation Lab: atsec

Date of publication: 2023-5-25

This is the Security Target for the Common Criteria Evaluation of the Huawei NetEngine 8000 Series Routers' Software. The TOE defined in this ST strictly consists of the software running on the supported hardware platforms. The Routers' hardware is out of the scope of the Common Criteria evaluation and, hence, not considered as part of the Target of Evaluation.

This Security Target includes in section 6 a set of Security Functional Requirements that are the taken from **[CPP_ND]**. Such functional requirements have been minimally adapted or application notes have been added to them where required, since the original Protection Profile also considers the routers' hardware as part of the TOE.

In addition, the **[CPP_ND]** applies refinements to some of the SFRs in **[CC31R5P2]** where some assignments are refined into selections with a closed set of options. However, these refinements are not deemed necessary for the following SFRs

- FCS_COP.1/SigGen
- FCS_COP.1/Hash
- FCS_COP.1/KeyedHash
- FMT_MOF.1/Functions
- FTP_TRP.1
- FAU_STG.1
- FAU_STG_EXT.3/LocSpace has been replaced by FAU_STG.3

1.2 TOE Reference

TOE Name: Huawei NetEngine 8000 Series Routers' Software

TOE Developer: Huawei Technologies Co., Ltd.

TOE Version: V800R022C00SPC600

The TOE is defined as the software running on the hardware corresponding to the following Huawei routers: NetEngine 8000 X4, NetEngine 8000 X8, NetEngine 8000 X16.

1.3 TOE Overview

1.3.1 Introduction

The TOE is a part of the software running on the NetEngine 8000 series routers. These routers consist of both hardware (non-TOE) and software. The software running on the routers is denominated Versatile Routing Platform (VRP) and the OS developed by Huawei. VRP provides extensive security features, including different interfaces for administrators, enforcing authentications prior to establishment of administrative sessions, auditing of security relevant management activities.

1.3.2 TOE Type

The TOE is software running on a network device that is connected to the network and has an infrastructure role within the network.

1.3.3 TOE Usage & Major Security Features

1.3.3.1 TOE usage

TOE usage is summarized below:

- The TOE supports username/password, or public-key authentication mode and only users that are authenticated can access the TOE and its command line interface.
- The TOE is accessed by CLI locally or a Network Management Server (NMS) remotely over SSH so that a secure channel is established to protect the data between TOE and NMS. This channel is used for TOE management by administrators.
- For secure transmission of audit information between the TOE and the Syslog server a secure TLS channel is used.
- The TOE supports digital signature verification for software. Each of the software package or patch package released by Huawei includes a unique digital signature. When an NMS distributes the package to router, the TOE will verify the online digital signature before updating. The verification of the digital signature demonstrates the integrity and authenticity of the package. The package is only processed further after successful verification of the digital signature, otherwise the package will be discarded without processing. Software integrity is verified for software update packages and for installed firmware upon every boot.

1.3.3.2 Major Security Features

The TOE is comprised of several security features. Below are identified the security features that are considered TSF:

- Security audit
- Cryptographic support

- Identification and authentication
- Secure Management
- Protection of the TSF
- TOE access
- Trusted path and channels

See section 1.4.2 for details.

1.3.4 Non-TOE Hardware/Software/Firmware

The TOE provides security services when running in a single and secure device. It supports (in some cases optionally) the following hardware, software, and firmware in its environment when the TOE is configured as in the Figure below:

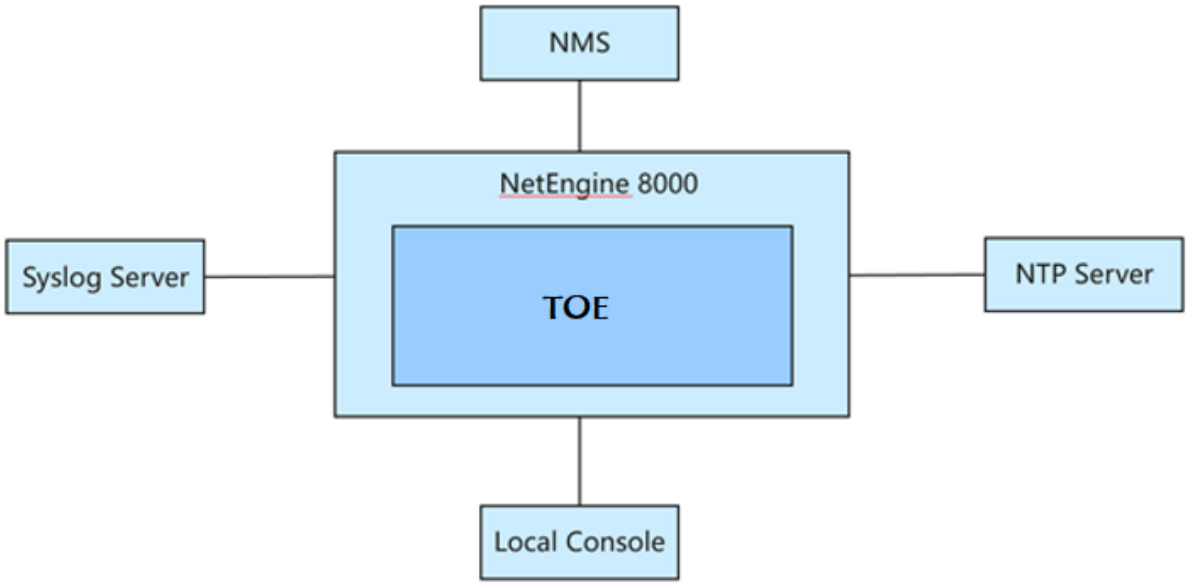


Figure 1 *Elements in the operational environment*

These IT entities (like the NTP server) should be physical protected in order to ensure that no one can attack them or steal information.

The TOE supports the following hardware, software, and firmware components in its operational environment. All of the following environment components are supported by the TOE.

Component	**Usage/Purpose Description for TOE performance**
NetEngine 8000 X4	The software runs on these hardware platforms.

NetEngine 8000 X8 NetEngine 8000 X16	Huawei NetEngine 8000 series routers are used to satisfy the requirements for networks of various scales. They are deployed at the edge of IP backbone networks, IP metropolitan area networks (MANs), and other large-scale IP networks, also can be used to access, aggregate, and transmit carrier-class Ethernet services on Fixed-Mobile Convergence (FMC) Metropolitan Area Networks (MANs). They providing network traffic processing capacity. Network traffic is processed and forwarded by the underlying hardware according to routing decisions downloaded from VRP software.
Network Management Server	This includes any Management workstation with an SSH client installed that is used to establish a protected channel with the TOE
Local Console	This includes any Console that is directly connected to the TOE via the Serial Console Port and is used by the TOE administrator to support TOE administration.
NTP Server	The TOE supports secure communications with an NTP server. In the evaluated configuration the TOE operates only as NTP Client to obtain a reliable source of time from the non-TOE NTP Server located in the operational environment to support the internal timestamp function. The TOE provides timestamps for internal use only.
Syslog Server	This includes any syslog server to which the TOE would transmit syslog messages.

Table 1 Hardware, software and firmware out of evaluation scope

Note: The TOE also supports a RADIUS AAA server in the operational environment providing user authentication to administrators. However, such setup is not contemplated in the evaluated configuration, where the administrators only authenticate against the TOE without relying on an external RADIUS AAA server.

1.4 TOE Description

1.4.1 Introduction

This section will introduce TOE from a software architectural view. The TOE scope consists of a part of the software running in the router device. The hardware is out of TOE scope.

The OS shown in Figure2 is based on a Linux Kernel. The OS provides basic services including memory management, scheduling management, file management, and device management.

The VRP software is a network operating platform, which has a distributed, multi-process, and component-based architecture. It builds upon the hardware development trend and will meet carriers' exploding service requirements.

The VRP software is responsible for functional management, routing information generation, receiving generated routing information and formatting them into hardware-specific data to direct traffic forwarding.

The diagram below describes the composition of the TOE.

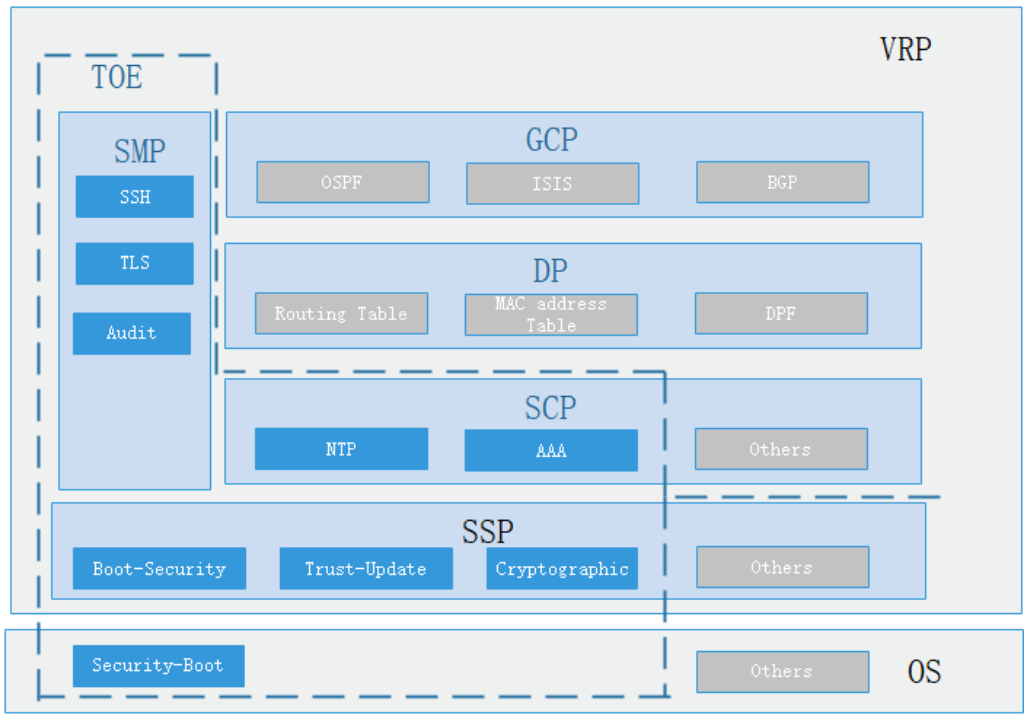


Figure 2 Architecture and boundaries of the Target of Evaluation

6 logical planes are defined for the complete software architecture of the routers, they are:

- System Manage Plane (SMP), implements management for external access, management for system configuration, information output on VRP;
- Service Control Plane (SCP), implements authentication, authorization, accounting and other serviceable functionality on VRP;
- System Service Plane (SSP), implements system internal scheduling, communication, management of signals, events, timers, etc. System security functions are also implemented at this plane.

- Operating System (OS, which is based on a Linux Kernel), provides hardware and software resource management.
- General Control Plane (GCP), implements routing information learning, ARP table entry learning, STP (Spanning Tree Protocol) topology management, and functionalities related to TCP/IP stack on VRP. GCP module is not part of the TOE;
- Data Plane (DP), implements traffic forwarding. Forwarding related information, e.g. routing information, ARP table entry, static MAC table entries are generated in GCP and downloaded via communication channel provided by SSP. DP module is not part of the TOE.

The table below presents all the software modules in the router and specifies which of them are TOE and which of them are not.

Modules	**TOE**	**Description**
AAA	YES	AAA (Authentication Authorization Accounting), implemented in accordance with related RFC, provides authentication, authorization and accounting functionalities.
BGP	NO	Border Gateway Protocol, the protocol backing the core routing decisions on the Internet. It maintains a table of IP networks or 'prefixes' which designate network reachability among autonomous systems (AS). It is described as a path vector protocol.
Cryptographic	YES	RSA, ECC, ECDSA, SHA, HMAC-SHA, AES, DRBG
DPF	NO	The DPF (Data Packet Forwarding) provides interfaces that send and receive packets on a router, while processing the packets at a high speed and switching data packets inside the router.
Audit	YES	Information Center, accepts, categorizes and filters information generated by all components

		and/or modules including log and alarm information, and outputs accordingly (e.g., to terminal, to log file).
IS-IS	NO	Intermediate System to Intermediate System (IS-IS) is a dynamic routing protocol initially designed by the International Organization for Standardization (ISO) for its Connectionless Network Protocol (CLNP). To support IP routing, the Internet Engineering Task Force (IETF) extends and modifies IS-IS in **[RFC-1195]**, which enables IS-IS to be applied to both TCP/IP and Open System Interconnection (OSI) environments. This type of IS-IS is called Integrated IS-IS or Dual IS-IS.
MAC Address Table	NO	Each device maintains a MAC address table. A MAC address table stores MAC addresses, VLAN IDs, and outbound interfaces learned from other devices, listed in Table 1. To forward data, the device searches the MAC address table to quickly locate the outbound interface based on the destination MAC address and VLAN ID in the data frame. This implementation reduces broadcast traffic.
OSPF	NO	Open Shortest Path First, is an adaptive routing protocol for Internet Protocol (IP) networks. It uses a link state routing algorithm and falls into the group of interior routing protocols, operating within a single autonomous system (AS).

NTP	YES	The Network Time Protocol (NTP) is an application layer protocol in the TCP/IP protocol suite. NTP synchronizes the time among a set of distributed time servers and clients. NTP module supports many operative modes: Manycast, Multicast, Broadcast. However, the TOE in its evaluated configuration works only as NTP client uses the NTP Server, located in the operating environment, to get reliable time synchronization.
Routing Table	NO	Routing tables store the routes discovered by various routing protocols. A router searches a routing table for routes, and each router maintains at least one routing table.
Boot-Security	YES	The TSF run a suite of self-tests during initial start-up to demonstrate the correct operation of the TSF, including software integration verification by digital signature check.
Trusted-Update	YES	Patches are a type of software compatible with system software. They are used to fix urgent bugs in system software. You can upgrade the system by installing patches, without having to upgrade the system software. You can select a proper operation to upgrade and maintain the device according to the real-world situation. Application scenarios of these operations are as follows: • System software upgrade

		System software upgrade can optimize device performance, add new features, and upgrade the current software version. • Patch installation
SSH	YES	Secure Shell (SSH v2.0), provides secure channel between end user and the TOE, and to protect the TOE from IP address fraud, password interception, etc.
TLS	YES	TLSv1.2 function performs loading digital certificate revocation list, and trusted CA file.
Others	NO	Functionalities which are not within the scope of this evaluation.
Security-Boot	YES	OS to checks the signature of VRP software.

Table 2 Modules specifications

1.4.2 TOE Logical Scope

The TOE is comprised of several security features. Each of the security features identified above consists of several security functionalities, as identified below.

1.4.2.1 Security audit

The log module of the host software records operations and events that occur on the appliance where the TOE runs. The recorded operations and events are log messages. Log messages provide evidence for diagnosing and maintaining a system. Log messages reflect the operating status of a device and are used to analyze the conditions of a network and to find out the causes of network failure or faults.

Key elements of log messages include timestamp, host name, Huawei identity, version, module name, severity, brief description, etc.

Audit component are the module processing, outputting log records. Information hierarchy is designed to help the user roughly differentiate between information about normal operation and information about faults. Since the information center needs to output information to the terminal, console, log buffer, and log file.

1.4.2.2 Cryptographic support

The TOE provides cryptography in support of secure connections that includes remote administrative management.
The cryptographic services provided by the TOE are described in Table below.

Cryptographic function	Use in the TOE
DRBG	Used in session establishment of TLS and SSH
RSA	Used for signature verification and generation in session establishment of TLS and SSH
SHA	Used to provide cryptographic hashing services
HMAC-SHA	Used to provide integrity and authentication verification
AES	Used to encrypt traffic transmitted through TLS and SSH
ECC	Used for signature verification and generation in session establishment of SSH

Table 3 *Cryptography provided by TOE*

1.4.2.3 Identification and authentication

The authentication functionality provides validation by user's account name and password. Public key authentication is supported for SSH users. Detailed functionalities, for example max idle-timeout period, max log-in attempts, UI lock, user kick out, can be configured by only administrator according to networking environment, customized security considerations.

1.4.2.4 Secure Management

The TOE restricts the ability to determine the behavior of and modify the behavior of the function's transmission of audit data to the security administrator. Only the security administrator can manage the cryptographic keys. Only the security administrator has the right of opening/closing the security services and creation/deletion/modification of the user accounts.

1.4.2.5 Protection of the TSF

The TOE protects the pre-shared keys, symmetric keys, and private keys from reading them by an unauthorized entity. The TOE stores the users or administrator passwords in non-plaintext and non-reversible form preventing them from reading. The TOE verifies the packet before their installation and uses the digital signature. The TOE provides Reliable Time-stamps functionality for internal use (e.g for associating a time stamp to a log) and synchronize its time using a NTP server as a reliable

source of time. The TOE performs self-test for integrity of the software and the cryptographic functions upon each boot. The TOE supports installation of software updates by administrators after a successful verification of their authenticity using secure and strong cryptographic algorithms based on digital signatures.

1.4.2.6 TOE access

The TOE supports to terminate the session, when a session is inactive for the configured period of time. Administrators can use a command to proactively terminate their interactive session in the TOE. The TOE provides default access banners, after the user login the TOE.

1.4.2.7 Trusted path and channels authentication

The TOE supports the trusted connections using TLS for the communication with the audit (syslog) server. The TOE supports the trusted connections using SSH for the communication with the remote users.

1.4.3 TOE Physical Scope

This section will define the physical scope of the Huawei NetEngine 8000 Series Routers' Software to be evaluated.

Type	Delivery Item	Delivery Method	Version
Product Guidance (AGD_PRE)	Huawei NetEngine 8000 Series Routers' Software V800R022C00 Preparative Procedures	Distributed by e-mail in PDF format	1.6
Product Guidance (AGD_OPE)	Huawei NetEngine 8000 Series Routers' Software V800R022C00 Operational User Guidance	Distributed by e-mail in PDF format	1.3
Upgrade Guide	NetEngine 8000 X V800R022C00SPC600 Upgrade Guide	Distributed by e-mail in .docx format or digital download available through the HUAWEI support website, through authenticated access.	1.0
Product Guidance (General)	NetEngine 8000 X V800R022C00SPC600 Product Documentation	Users can log in to the HUAWEI support website to read the	1.0

		document directly or download the product documentation in accordance to the version of the TOE. The download file format is *.hdx, user can download the *.hdx reader from the same website.	
TOE Software	V800R022C00SPC600.cc SHA256: 4d05057b27f4aad67b816e8150775483 5554e10e26a9714830e4e4c8af675268	Users can login the HUAWEI support website to download the software packet in accordance to the version of the TOE. Users can verify the software by digital signature(The digital signature is also published on HUAWEI support website)	V800R022C00SPC600
Product Guidance (Signature Verification)	OpenPGP Signature Verification Guide.pdf	Distributed by e-mail in PDF format	4

Table 4 *Physical Scope*

2 Conformance Claims

This Security Target and the TOE described are in accordance with the requirements of Common Criteria 3.1R5.

This Security Target claims conformance with the following parts of Common Criteria:

- Conformance with [CC31R5P2] extended.
- Conformance with [CC31R5P3] conformant.

This Security Target does not claim conformance with any protection profile.

The chosen assurance level chosen for this Common Criteria Evaluation is EAL2 augmented with ALC_FLR.2.

3 Security Problem Definition

This section describes the security aspects of the operational environment and its expected use in said environment. It includes the declaration of the TOE operational environment that identifies and describes:

- The alleged known threats that will be countered by the TOE
- The organizational security policies that the TOE has to adhere to
- The TOE usage assumptions in the suggested operational environment.

We will begin defining Assets and Agents of threats.

3.1 Assets

TOE ADMINISTRATOR ACCESS: Access to an authenticated session of an administrator user on the TOE. This allows to perform TOE configuration modifications at an administrative level. Potentially it could allow malicious actions that compromise the security functionality of the TOE and the network on which the hardware where the TOE runs.

TOE NETWORK TRAFFIC: Traffic incoming and outcoming of the TOE Network interfaces), exchanged with external entities in its operational environment. This traffic needs to be protected in confidentiality and integrity while it is in transit between the TOE and another communication endpoint.

TOE FIRMWARE: Executable firmware code of the TOE, which provides the TSF implementation. Integrity must be preserved.

TOE CONFIGURATION: Those configuration parameters on which the behavior of the TOE functionality depends. TOE configuration shall be protected against unauthorized modification (authorization is required) and undetected modification (traceability).

ADMINISTRATIVE PASSWORDS: Passwords used to authenticate as an administrator user on the TOE.

CRYPTOGRAPHIC KEYS: Cryptographic keys used by the TOE for providing its cryptographic services.

TSF INTEGRITY: Integrity of the TOE Security Functionality must be preserved in order to guarantee that the TOE Security Functionality is as described in this ST.

3.2 Threat Agents

ATTACKER: Any individual using the TOE services in a way that intends to vulnerate or compromise the security of the TOE assets, such as:

- An eavesdropper, who has access to communication channels through which TSF data are transferred.
- An unauthorized user of the TOE, who gains unauthorized access to the TOE.

3.3 Threats to Security

This section identifies the threats to assets that require protection by the TOE. The threats are defined in terms of assets concerned, attackers and the adverse action that materializes the threat.

T.UNAUTHORIZED_ADMINISTRATOR_ACCESS: An **Attacker** may attempt to gain **TOE Administrator Access** to the TOE by nefarious means such as: masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between network devices.

Successfully gaining **TOE Administrator Access** allows malicious actions that compromise the security functionality of the device and the network on which it resides.

T.WEAK_CRYPTOGRAPHY: An **Attacker** may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the **TOE Network Traffic** with minimal effort.

T.UNTRUSTED_COMMUNICATION_CHANNELS: An **Attacker** may attempt to target communication channels with the TOE that do not use standardized secure tunneling protocols to protect the critical **TOE Network Traffic**. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical **TOE Network Traffic**, and potentially could lead to a compromise of the TOE itself.

T.WEAK_AUTHENTICATION_ENDPOINTS: An **Attacker** may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical **TOE Network Traffic** is exposed and there could be a loss of confidentiality and integrity, and potentially the TOE itself could be compromised if **TOE Administrator Access** is gained.

T.UPDATE_COMPROMISE: An **Attacker** may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration, hence compromising the **TOE Firmware**.

T.UNDETECTED_ACTIVITY: An **Attacker** may attempt to access, change, and/or modify the security functionality of the TOE by modifying the **TOE Configuration** without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the TOE and the Administrator would have no knowledge that it has been compromised.

T.CRYPTO_KEY_COMPROMISE: An **Attacker** may compromise and TOE **Cryptographic Keys** in order to gain access to the TOE or TOE data by using the keys that the TOE uses for encryption of its communication channels.

T.ADMIN_PASSWORD_COMPROMISE: An **Attacker** may be able to take advantage of weak administrative passwords to use brute force attacks or massive authentication attempts to the TOE, obtaining access to **Administrative Passwords**, or non-secure mechanisms for storage of keys on the TOE may be exploited for the same purpose. If obtained, **TOE Administrator Access** could be gained in an illicit way.

T.SECURITY_FUNCTIONALITY_FAILURE: An **Attacker** could take advantage of a failure in the security functionality of the TOE during start-up or during operations, causing a compromise of the **TSF Integrity**, and leaving the device susceptible to attackers.

3.4 Organizational Security Policies

The organizational Security policies are defined as follows.

P.ACCESS_BANNER: The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

3.5 Assumptions

The assumptions when using the TOE are the following:

A.PHYSICAL_PROTECTION: The TOE is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the interconnections of the physical device where the TOE software runs, and correct operation. This protection is assumed to be sufficient to protect the device where the TOE runs. As a result, the ST will not include any requirements on physical tamper protection or other physical attack mitigations. The ST will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device where the TOE runs. The NTP server and the local console are physically protected.

A.NO_THRU_TRAFFIC_PROTECTION: A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).

A.TRUSTED_ADMINISTRATOR: The Security Administrator(s) for the TOE are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when

administering the TOE. The TOE is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the TOE.

A.REGULAR_UPDATES: The TOE firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

A.ADMIN_CREDENTIALS_SECURE: The Administrator's credentials used to access the network device are protected by the platform on which they reside.

A.RESIDUAL_INFORMATION: The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

4 Security Objectives

The security objectives are high level declarations, concise and abstract of the solution to the problem exposed in the former section, which counteracts the threats and fulfills the security policies and the assumptions. These consist of:

- the security objectives for the operational environment.
- the security objectives for the TOE

4.1 Security objectives for the TOE

The security objectives for the TOE must determine (to the desired extent) the responsibility of the TOE in countering the threats and in enforcing the OSPs. Each objective must be traced back to aspects of identified threats to be countered by the TOE and to aspects of OSPs to be met by the TOE.

O.ADMIN_AUTH: The TOE shall require identification and authentication of administrators before granting them access to the TOE management functions. The TOE management capabilities available to administrators shall be defined and limited, and actions available prior to authentication shall be limited and constrained. Administrators' authentication process shall consist in local authentication on the TOE using passwords through a secure communication channel, and authentication sessions will be closed by the TOE after a defined period of inactivity.

O.STRONG_CRYPTO: The TOE shall use robust cryptographic algorithms, compliant to industry approved standards, in order to provide robust cryptographic protection of network communications.

O.TRUSTED_COMM: The TOE shall implement secure channels that use standardized tunneling protocols to protect the critical network traffic, ensuring protection of the communications between the TOE and trusted external entities in confidentiality, integrity, and protection against communication replays.

O.STRONG_AUTHENTICATION_ENDPOINT: The TOE shall implement methods for robust and reliable authentication of trusted entities with the TOE, preventing attacks based on weak authentication methods (e.g. guessing or transported shared keys).

O.SECURE_UPDATES: The TOE shall provide to administrators the capability of installing software or firmware updates only after a successful verification of their authenticity using secure and strong cryptographic algorithms based on digital signatures.

O.ACTIVITY_AUDIT: The TOE shall generate audit records for relevant management actions carried by administrators. Audit records will be marked with precise timestamps, associated to user identity, and protected from unauthorized modification or deletion.

O.CRYPTO_KEY_PROTECTION: The TOE shall protect stored cryptographic keys in a way that prevents unauthorized access. Management of cryptographic keys shall be restricted to Security Administrators and key destruction shall be performed in a secure way that prevents key recover from residual information.

O.PASSWORD_PROTECTION: The TOE shall protect the passwords user for local administrator authentication by enforcing complexity and quality rules. Also, the TOE shall limit failed authentication attempts and limit the feedback given to users on failed authentications, in order to prevent brute force or guessing attacks. Also, the TOE shall perform secure storage of passwords, refraining from storing them in plaintext.

O.SELF_TEST: The TOE shall perform self-tests of the TSF functionality in order to detect potential failures during start-up or operation and prevent further situations that could lead to malfunction.

O.BANNER: The TOE shall display an advisory notice and consent about use of the TOE to administrators before establishing an administrative user session,

4.2 Security objectives for the operational environment

The security objectives for the Operational Environment determine the responsibility of the environment in countering the threats, enforcing the OSPs and upholding the assumptions. Each objective must be traced back to aspects of identified threats to be countered by the environment, to aspects of OSPs to be enforced by the environment and to assumptions to be uphold by the environment.

OE.PHYSICAL: Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.NO_THRU_TRAFFIC_PROTECTION: The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

OE.TRUSTED_ADMIN: TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

OE.UPDATES: The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

OE.ADMIN_CREDENTIALS_SECURE: The administrator's credentials used to access the TOE must be protected on any other platform on which they reside.

OE.RESIDUAL_INFORMATION: The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

4.3 Security Objectives Rationale

The following table provides a mapping of security objectives tracing each security objective for the TOE back to threats countered by that security objective and OSPs enforced by that security objective,

and each security objective for the operational environment back to threats countered by that security objective, OSPs enforced by that security objective, and assumptions upheld by that security objective. This illustrates that the security objectives counter all threats, the security objectives enforce all OSPs and the security objectives for the operational environment uphold all assumptions.

	O.ADMIN_AUTH	O.STRONG_CRYPTO	O.TRUSTED_COMM	O.STRONG_AUTHENTICATION_ENDPOINT	O.SECURE_UPDATES	O.ACTIVITY_AUDIT	O.CRYPTO_KEY_PROTECTION	O.PASSWORD_PROTECTION	O.SELF_TEST	O.BANNER	OE.PHYSICAL	OE.NO_THRU_TRAFFIC_PROTECTION	OE.TRUSTED_ADMIN	OE.UPDATES	OE.ADMIN_CREDENTIALS_SECURE	OE.RESIDUAL_INFORMATION
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	X															
T.WEAK_CRYPTOGRAPHY		X														
T.UNTRUSTED_COMMUNICATION_CHANNELS			X													
T.WEAK_AUTHENTICATION_ENDPOINTS				X												
T.UPDATE_COMPROMISE					X											
T.UNDETECTED_ACTIVITY						X										
T.CRYPTO_KEY_COMPROMISE							X									
T.ADMIN_PASSWORD_COMPROMISE								X								
T.SECURITY_FUNCTIONALITY_FAILURE									X							
P.ACCESS_BANNER										X						
A.PHYSICAL_PROTECTION											X					
A.NO_THRU_TRAFFIC_PROTECTION												X				
A.TRUSTED_ADMINISTRATOR													X			

	O.ADMIN_AUTH	O.STRONG_CRYPTO	O.TRUSTED_COMM	O.STRONG_AUTHENTICATION_ENDPOINT	O.SECURE_UPDATES	O.ACTIVITY_AUDIT	O.CRYPTO_KEY_PROTECTION	O.PASSWORD_PROTECTION	O.SELF_TEST	O.BANNER	OE.PHYSICAL	OE.NO_THRU_TRAFFIC_PROTECTION	OE.TRUSTED_ADMIN	OE.UPDATES	OE.ADMIN_CREDENTIALS_SECURE	OE.RESIDUAL_INFORMATION
A.REGULAR_UPDATES														X		
A.ADMIN_CREDENTIALS_SECURE															X	
A.RESIDUAL_INFORMATION																X

Table 5 *Security Objectives vs Security Problem Definition*

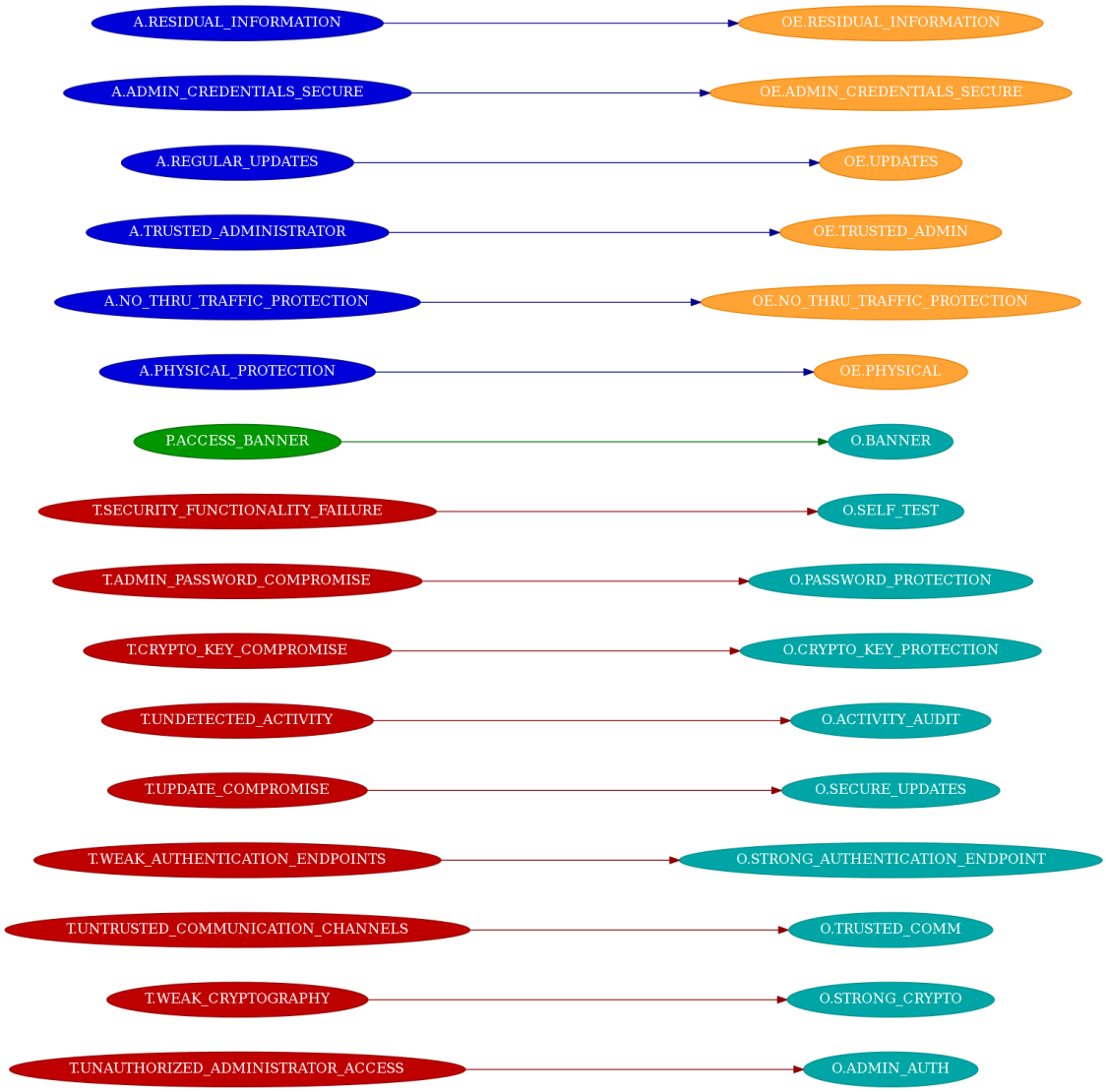


Figure 3 Mapping of Security Problem Definition to Security Objectives

4.3.1 Threats

T.UNAUTHORIZED_ADMINISTRATOR_ACCESS: This threat is countered by **O.ADMIN_AUTH** which requires identification and authentication of administrator before granting them access to the TOE and to management functions. It also enforces that the TOE requires identification and authentication of administrators before granting them access to the TOE management functions.

The TOE management capabilities available to administrators shall be defined and limited, and actions available prior to authentication shall be limited and constrained.

Administrators' authentication process shall consist in local authentication of the TOE using passwords through a secure communication channel, and authentication sessions will be closed by the TOE after a defined period of inactivity.

T.WEAK_CRYPTOGRAPHY: This threat is countered by **O.STRONG_CRYPTO** which requires usage of robust cryptographic algorithms, compliant to industry approved standards, in order to provide robust cryptographic protection of network communications.

T.UNTRUSTED_COMMUNICATION_CHANNELS: This threat is countered by **O.TRUSTED_COMM** which requires secure communication channels that use standardized tunneling protocols to protect the critical network traffic, ensuring protection of the communications between the TOE and trusted external entities in confidentiality, integrity, and protection against communication replays.

T.WEAK_AUTHENTICATION_ENDPOINTS: This threat is countered by **O.STRONG_AUTHENTICATION_ENDPOINT** which requires methods for strong authentication of trusted entities with the TOE, preventing attacks based on weak authentication methods.

T.UPDATE_COMPROMISE: This threat is countered by **O.SECURE_UPDATES** which requires verification of updates authenticity by administrators based on cryptographic digital signatures.

T.UNDETECTED_ACTIVITY: This threat is countered by **O.ACTIVITY_AUDIT** which requires the generation of audit records for relevant management actions carried by administrators, marked with precise timestamps, associated to user identity, and protected from unauthorized modification or deletion.

T.CRYPTO_KEY_COMPROMISE: This threat is countered by **O.CRYPTO_KEY_PROTECTION** which requires protection of stored cryptographic keys in order to prevent unauthorized access, restricting management of cryptographic keys to administrators, and enforcing secure key destruction methods.

T.ADMIN_PASSWORD_COMPROMISE: This threat is countered by **O.PASSWORD_PROTECTION** which requires the TOE to enforce password complexity and quality in passwords used by administrators for authentication, hence preventing successful attacks to weak passwords. The same objective also forbids plaintext storage of passwords in the TOE and prevents attacks based on massive authentication attempts or guessing passwords from feedback resulting from failed authentication attempts.

T.SECURITY_FUNCTIONALITY_FAILURE: This threat is countered by **O.SELF_TEST** which requires that The TOE carries outs TSF self-tests in order to detect potential failures during start-up or operation and prevent further situations that could lead to malfunction.

The following table maps the threats of the security problem established to the security objectives of the TOE and the security objectives of the operational environment.

Threats	Security Objectives
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	O.ADMIN_AUTH
T.WEAK_CRYPTOGRAPHY	O.STRONG_CRYPTO
T.UNTRUSTED_COMMUNICATION_CHANNELS	O.TRUSTED_COMM
T.WEAK_AUTHENTICATION_ENDPOINTS	O.STRONG_AUTHENTICATION_ENDPOINT
T.UPDATE_COMPROMISE	O.SECURE_UPDATES
T.UNDETECTED_ACTIVITY	O.ACTIVITY_AUDIT
T.CRYPTO_KEY_COMPROMISE	O.CRYPTO_KEY_PROTECTION
T.ADMIN_PASSWORD_COMPROMISE	O.PASSWORD_PROTECTION
T.SECURITY_FUNCTIONALITY_FAILURE	O.SELF_TEST

Table 6 Threats vs Security Objectives

4.3.2 Organizational Security Policies

P.ACCESS_BANNER: This policy is enforced by **O.BANNER** which requires that the TOE displays an advisory notice and consent about use of the TOE to administrators before establishing an administrative user session,

The following table maps the organizational security policies of the problem established to the security objectives of the TOE and the security objectives of the operational environment.

OSPs	Security Objectives
P.ACCESS_BANNER	O.BANNER

Table 7 OSPs vs Security Objectives

4.3.3 Assumptions

A.PHYSICAL_PROTECTION: This assumption is directly upheld by **OE.PHYSICAL**, which requires that physical protection to the TOE is provided by the operational environment.

A.NO_THRU_TRAFFIC_PROTECTION: This assumption is directly upheld by **OE.NO_THRU_TRAFFIC_PROTECTION**, which requires that the TOE does not provide any protection of traffic that traverses it, but such protection is covered by other security and assurance measures in the operational environment.

A.TRUSTED_ADMINISTRATOR: This assumption is directly upheld by **OE.TRUSTED_ADMIN**, which requires that TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

A.REGULAR_UPDATES: This assumption is directly upheld by **OE.UPDATES**, which requires that the TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

A.ADMIN_CREDENTIALS_SECURE: This assumption is directly upheld by **OE.ADMIN_CREDENTIALS_SECURE**, which requires that the administrator's credentials (private key) used to access the TOE are protected on any other platform on which they reside.

A.RESIDUAL_INFORMATION: This assumption is directly upheld by **OE.RESIDUAL_INFORMATION** which requires administrators to ensure that there is no unauthorized access possible for sensitive residual information on networking equipment when the equipment is discarded or removed from its operational environment.

The following table maps the assumptions of the problem established to the security objectives of the TOE and the security objectives of the operational environment.

Assumptions	Security Objectives
A.PHYSICAL_PROTECTION	OE.PHYSICAL
A.NO_THRU_TRAFFIC_PROTECTION	OE.NO_THRU_TRAFFIC_PROTECTION
A.TRUSTED_ADMINISTRATOR	OE.TRUSTED_ADMIN
A.REGULAR_UPDATES	OE.UPDATES
A.ADMIN_CREDENTIALS_SECURE	OE.ADMIN_CREDENTIALS_SECURE
A.RESIDUAL_INFORMATION	OE.RESIDUAL_INFORMATION

Table8 Assumptions vs Security Objectives for the Operational Environment

5 Extended Components Definition

5.1 Class FAU: Security audit

Security auditing involves recognising, recording, storing, and analysing information related to security relevant activities (i.e. activities controlled by the TSF). The resulting audit records can be examined to determine which security relevant activities took place and whom (which user) is responsible for them.

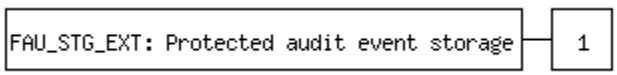
FAU class is extended in order to add the families FAU_STG_EXT. Both classes are defined to include new SFRs related to audit data generation features and audit storage features (respectively) that are not covered by FAU_GEN and FAU_STG classes. New families are added due to the meaningful differences between the extended components defined and those already existing in FAU_GEN and FAU_STG families.

5.1.1 Protected audit event storage (FAU_STG_EXT)

Family behavior

This component defines the requirements for the TSF to be able to securely transmit audit data between the TOE and an external IT entity.

Component levelling



Protected audit event storage requires the TSF to use a trusted channel implementing a secure protocol.

Management: FAU_STG_EXT.1

The following actions could be considered for the management functions in FMT:
- The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_STG_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- No audit necessary.

FAU_STG_EXT.1: Protected Audit Event Storage

Hierarchical to:

No other components.

Dependencies:

FAU_GEN.1

FTP_ITC.1

FAU_STG_EXT.1.1: ***The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.***

FAU_STG_EXT.1.2: ***The TSF shall be able to store generated audit data on the TOE itself. [selection: TOE shall consist of a single standalone component that stores audit data locally, The TOE shall be a distributed TOE that stores audit data on the following TOE components: [assignment: identification of TOE components], The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [assignment: list of TOE components that do not store audit data locally and the other TOE components to which they transmit their generated audit data]]***

FAU_STG_EXT.1.3: ***The TSF shall [selection: drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]] when the local storage space for audit data is full.***

5.2 Class FCS: Cryptographic support

The TSF may employ cryptographic functionality to help satisfy several high-level security objectives. These include (but are not limited to): identification and authentication, non-repudiation, trusted path, trusted channel and data separation. This class is used when the TOE implements cryptographic functions, the implementation of which could be in hardware, firmware and/or software.

The FCS class is composed of two families: FCS_CKM and FCS_COP. The FCS_CKM family addresses the management aspects of cryptographic keys, while the FCS_COP family is concerned with the operational use of those cryptographic keys.

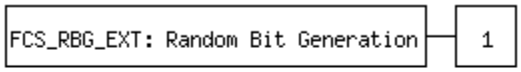
FCS class is extended in order to add the families FCS_TLSC_EXT, FCS_SSHS_EXT, FCS_SSHC_EXT and FCS_RGB_EXT.FCS_TLSC_EXT cover cryptographic requirements associated to TLS communications. FCS_SSHS_EXT includes cryptographic requirements related to SSH server implementation, while FCS_SSHC_EXT includes those related SSH client implementations. FCS_RGB_EXT includes requirements for random number generators, used for supporting other cryptographic operations. None of the requirements of those classes already exist in the Common Criteria standard.

5.2.1 Random Bit Generation (FCS_RBG_EXT)

Family behavior

Components in this family address the requirements for random bit/number generation. This is a new family defined for the FCS class.

Component levelling



Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source.

Management: FCS_RBG_EXT.1

The following actions could be considered for the management functions in FMT:
- There are no management activities foreseen

Audit: FCS_RBG_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- Minimal: failure of the randomization process

FCS_RBG_EXT.1: Random Bit Generation

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FCS_RBG_EXT.1.1: ***The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)]***

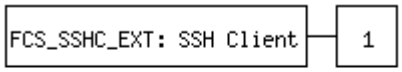
FCS_RBG_EXT.1.2: ***The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [selection: [assignment: number of software-based sources] software-based noise source, [assignment: number of hardware-based sources] hardware-based noise source] with a minimum of [selection: 128 bits, 192 bits, 256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.***

5.2.2 SSH Client (FCS_SSHC_EXT)

Family behavior

The component in this family addresses the ability for a client to use SSH to protect data between the client and a server using the SSH protocol.

Component levelling



SSH Client requires that the client side of SSH be implemented as specified.

Management: FCS_SSHC_EXT.1

The following actions could be considered for the management functions in FMT:
- There are no management activities foreseen.

Audit: FCS_SSHC_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:
- Failure of SSH session establishment
- SSH session establishment
- SSH session termination

FCS_SSHC_EXT.1: SSH Client

Hierarchical to:

No other components.

Dependencies:

FCS_CKM.1

FCS_CKM.2

FCS_COP.1/DataEncryption

FCS_COP.1/SigGen

FCS_COP.1/Hash

FCS_COP.1/KeyedHash

FCS_RBG_EXT.1

FCS_SSHC_EXT.1.1: ***The TSF shall implement the SSH protocol that complies with RFC(s) [selection: 4251, 4252, 4253, 4254, 5647, 5656, 6187, 6668, 8332]***

FCS_SSHC_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, [selection: password-based, no other method]***

FCS_SSHC_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.***

FCS_SSHC_EXT.1.4: ***The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [selection: aes128-cbc, aes256-***

cbc, aes128-ctr, aes256-ctr, AEAD_AES_128_GCM, AEAD_AES_256_GCM, aes128-gcm@openssh.com, aes256-gcm@openssh.com]

FCS_SSHC_EXT.1.5: ***The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256] as its public key algorithm(s) and rejects all other public key algorithms***

FCS_SSHC_EXT.1.6: ***The TSF shall ensure that the SSH transport implementation uses [selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM, implicit] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).***

FCS_SSHC_EXT.1.7: ***The TSF shall ensure that [selection: diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, ecdh-sha2-nistp256] and [selection: diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.***

FCS_SSHC_EXT.1.8: ***The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.***

FCS_SSHC_EXT.1.9: ***The TSF 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.***

5.2.3 SSH Server Protocol (FCS_SSHS_EXT)

Family behavior

The component in this family addresses the ability for a server to offer SSH to protect data between a client and the server using the SSH protocol.

Component levelling

SSH Server requires that the server side of SSH be implemented as specified.

Management: FCS_SSHS_EXT.1

The following actions could be considered for the management functions in FMT:
- There are no management activities foreseen.

Audit: FCS_SSHS_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:
- Failure of SSH session establishment
- SSH session establishment
- SSH session termination

FCS_SSHS_EXT.1: SSH Server Protocol

Hierarchical to:

No other components.

Dependencies:

FCS_CKM.1

FCS_CKM.2

FCS_COP.1/DataEncryption

FCS_COP.1/SigGen

FCS_COP.1/Hash

FCS_COP.1/KeyedHash

FCS_RBG_EXT.1

FCS_SSHS_EXT.1.1: ***The TSF shall implement the SSH protocol that complies with RFC(s) [selection: 4251, 4252, 4253, 4254, 5647, 5656, 6187, 6668, 8332]***

FCS_SSHS_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, [selection: password-based, no other method]***

FCS_SSHS_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.***

FCS_SSHS_EXT.1.4: ***The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [selection: aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr, AEAD_AES_128_GCM, AEAD_AES_256_GCM, aes128-gcm@openssh.com, aes256-gcm@openssh.com]***

FCS_SSHS_EXT.1.5: ***The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256] as its public key algorithm(s) and rejects all other public key algorithms.***

FCS_SSHS_EXT.1.6: ***The TSF shall ensure that the SSH transport implementation uses [selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM, implicit] as its MAC algorithm(s) and rejects all other MAC algorithm(s).***

FCS_SSHS_EXT.1.7: ***The TSF shall ensure that [selection: diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, ecdh-sha2-nistp256] and [selection: diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.***

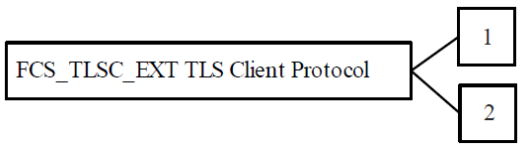
FCS_SSHS_EXT.1.8: ***The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed***

5.2.4 TLS Client Protocol (FCS_TLSC_EXT)

Family behavior

The component in this family addresses the ability for a client to use TLS to protect data between the client and a server using the TLS protocol.

Component levelling



TLS Client requires that the client side of TLS be implemented as specified.

TLS Client requires that the client side of the TLS implementation include mutual authentication.

Management: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

The following actions could be considered for the management functions in FMT:
- There are no management activities foreseen.

Audit: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:
- Failure of TLS session establishment
- TLS session establishment
- TLS session termination

FCS_TLSC_EXT.1: TLS Client Protocol

Hierarchical to:

No other components.

Dependencies:

FCS_CKM.1

FCS_CKM.2

FCS_COP.1/DataEncryption

FCS_COP.1/SigGen

FCS_COP.1/Hash

FCS_COP.1/KeyedHash

FCS_RBG_EXT.1

FCS_TLSC_EXT.1.1: ***The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites [selection: TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268, TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268, TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268, TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492, TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492, TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246, TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246, TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246, TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246, TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288, TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288, TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288, TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289, TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289, TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289]***

FCS_TLSC_EXT.1.2: ***The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.***

FCS_TLSC_EXT.1.3: ***When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [selection: Not implement any administrator override mechanism, require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate]***

FCS_TLSC_EXT.1.4: ***The TSF shall [selection: not present the Supported Elliptic Curves Extension, present the Supported Elliptic Curves Extension with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves] in the Client Hello.***

FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication

Hierarchical to:
No other components

Dependencies:
FCS_CKM.1

FCS_CKM.2

FCS_COP.1/DataEncryption

FCS_COP.1/SigGen

FCS_COP.1/Hash

FCS_COP.1/KeyedHash

FCS_RBG_EXT.1

FCS_TLSC_EXT.1

FIA_X509_EXT.1

FIA_X509_EXT.2

FCS_TLSC_EXT.2.1 The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

5.3 Class FIA: Identification and authentication

Families in this class address the requirements for functions to establish and verify a claimed user identity.

Identification and Authentication is required to ensure that users are associated with the proper security attributes (e.g. identity, groups, roles, security or integrity levels).

The unambiguous identification of authorised users and the correct association of security attributes with users and subjects is critical to the enforcement of the intended security policies. The families in this class deal with determining and verifying the identity of users, determining their authority to interact with the TOE, and with the correct association of security attributes for each authorised user. Other classes of requirements (e.g. User Data Protection, Security Audit) are dependent upon correct identification and authentication of users in order to be effective.

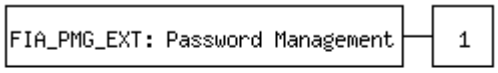
FIA class is extended in order to add the families FIA_X509_EXT, FIA_UIA_EXT, FIA_PGM_EXT, and FIA_UAU_EXT.
Those families are defined to include new SFRs related to identification based on X509 cryptography (FIA_X509_EXT), common requirements for identification and authentication (FIA_UIA_EXT), password management requirements (FIA_PGM_EXT) and additional user authentication requirements (FIA_UAU_EXT). In the case of FIA_UAU_EXT, this new family is added due to the meaningful differences between the extended components defined and those already existing in FIA_UAU family.

5.3.1 Password Management (FIA_PMG_EXT)

Family behavior

The TOE defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

Component levelling



Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

Management: FIA_PMG_EXT.1

No management functions.

Audit: FIA_PMG_EXT.1

No specific audit requirements.

FIA_PMG_EXT.1: Password Management

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FIA_PMG_EXT.1.1: ***The TSF shall provide the following password management capabilities for administrative passwords:***

a) ***Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [selection: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: other characters]]***

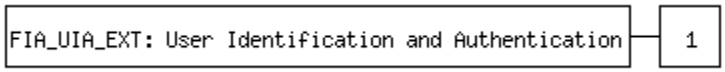
b) ***Minimum password length shall be configurable to between [assignment: minimum number of characters supported by the TOE] and [assignment: number of characters greater than or equal to 15] characters.***

5.3.2 User Identification and Authentication (FIA_UIA_EXT)

Family behavior

The TSF allows certain specified actions before the non-TOE entity goes through the identification and authentication process.

Component levelling



User Identification and Authentication requires Administrators (including remote Administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions

Management: FIA_UIA_EXT.1

The following actions could be considered for the management functions in FMT:
- Ability to configure the list of TOE services available before an entity is identified and authenticated

Audit: FIA_UIA_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- All use of the identification and authentication mechanism
- Provided user identity, origin of the attempt (e.g. IP address)

FIA_UIA_EXT.1: User Identification and Authentication

Hierarchical to:

No other components.

Dependencies:

FTA_TAB.1

FIA_UIA_EXT.1.1: ***The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:***

a) ***Display the warning banner in accordance with FTA_TAB.1***

b) ***[selection: no other actions, automated generation of cryptographic keys, [assignment: list of services, actions performed by the TSF in response to non-TOE requests]]***

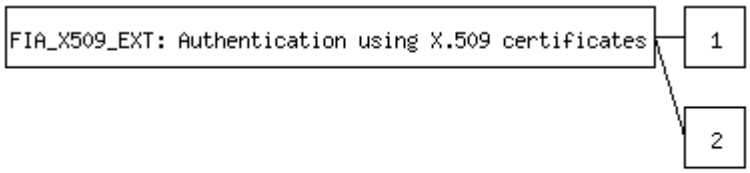
FIA_UIA_EXT.1.2: ***The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.***

5.3.3 Authentication using X.509 certificates (FIA_X509_EXT)

Family behavior

This family defines the behaviour, management, and use of X.509 certificates for functions to be performed by the TSF. Components in this family require validation of certificates according to a specified set of rules, use of certificates for authentication for protocols and integrity verification, and the generation of certificate requests.

Component levelling



X509 Certificate Validation, requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component.

X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates.

Management: FIA_X509_EXT.1

The following actions could be considered for the management functions in FMT:
- Remove imported X.509v3 certificates
- Approve import and removal of X.509v3 certificates
- Initiate certificate requests

Management: FIA_X509_EXT.2

The following actions could be considered for the management functions in FMT:
- Remove imported X.509v3 certificates
- Approve import and removal of X.509v3 certificates
- Initiate certificate requests

Audit: FIA_X509_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- Minimal: No specific audit requirements are specified.

Audit: FIA_X509_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- Minimal: No specific audit requirements are specified.

FIA_X509_EXT.1: X.509 Certificate Validation

Hierarchical to:

No other components.

Dependencies:

FIA_X509_EXT.2 X.509 Certificate Authentication.

FIA_X509_EXT.1.1: ***The TSF shall validate certificates in accordance with the following rules:***

- ***RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.***

- ***The certification path must terminate with a trusted CA certificate designated as a trust anchor.***

- ***The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.***

- ***The TSF shall validate the revocation status of the certificate using [selection: the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5]***

- ***The TSF shall validate the extendedKeyUsage field according to the following rules:***

    - ***Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.***

    - ***Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.***

- ***Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.***

- ***OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.***

FIA_X509_EXT.1.2: ***The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.***

FIA_X509_EXT.2: X509 Certificate Authentication

Hierarchical to:

No other components.

Dependencies:

FIA_X509_EXT.1

FIA_X509_EXT.2.1: ***The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection: DTLS, HTTPS, IPsec, TLS, SSH, [assignment: other protocols], no protocols], and [selection: code signing for system software updates, code signing for integrity verification, [assignment: other uses], no additional uses]***

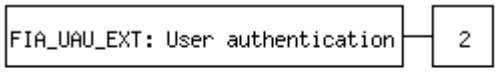
FIA_X509_EXT.2.2: ***When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: allow the Administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate]***

5.3.4 User authentication (FIA_UAU_EXT)

Family behavior

Provides for a locally based administrative user authentication mechanism.

Component levelling



The password-based authentication mechanism provides administrative users a locally based authentication mechanism.

Management: FIA_UAU_EXT.2

The following actions could be considered for the management functions in FMT:
- None

Audit: FIA_UAU_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- Minimal: All use of the authentication mechanism

FIA_UAU_EXT.2: Password-based Authentication Mechanism

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FIA_UAU_EXT.2.1: The TSF shall provide a local ***[selection: password-based, SSH public key-based, certificate-based, [assignment: other authentication mechanism(s)]*** authentication mechanism to perform local administrative user authentication.

5.4 Class FPT: Protection of the TSF

This class contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data. In some sense, families in this class may appear to duplicate components in the FDP class; they may even be implemented using the same mechanisms. However, FDP focuses on user data protection, while FPT focuses on TSF data protection. In fact, components from the FPT class are necessary to provide requirements that the SFPs in the TOE cannot be tampered with or bypassed.

From the point of view of this class, regarding to the TSF there are three significant elements:

- The TSF's implementation, which executes and implements the mechanisms that enforce the SFRs.
- The TSF's data, which are the administrative databases that guide the enforcement of the SFRs.
- The external entities that the TSF may interact with in order to enforce the SFRs.

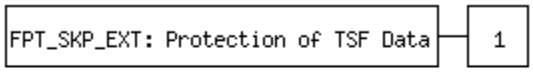
FPT class is extended in order to add the families FPT_TUD_EXT, FPT_TST_EXT, FPT_STM_EXT, FPT_SKP_EXT and FPT_APW_EXT. In the case of FPT_TST_EXT, a self-test requirement for the TOE is added that presents meaningful differences with those existing in FPT_TST. The rest of the mentioned families include new requirements related to TOE self-protection that are not covered in any of the existing FPT families of the Common Criteria standard.

5.4.1 Protection of TSF Data (FPT_SKP_EXT)

Family behavior

Components in this family address the requirements for managing and protecting TSF data, such as cryptographic keys. This is a new family modelled after the FPT_PTD Class.

Component levelling



Protection of TSF Data (for reading all symmetric keys), requires preventing symmetric keys from being read by any user or subject. It is the only component of this family.

Management: FPT_SKP_EXT.1

The following actions could be considered for the management functions in FMT:
- There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- There are no auditable events foreseen.

FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)

Hierarchical to:

No other components.

Dependencies:

No dependencies.

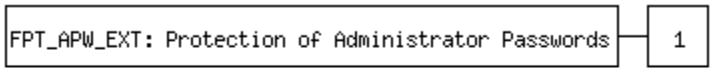
FPT_SKP_EXT.1.1: ***The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.***

5.4.2 Protection of Administrator Passwords (FPT_APW_EXT)

Family behavior

Components in this family ensure that the TSF will protect plaintext credential data such as passwords from unauthorized disclosure.

Component levelling



Protection of Administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.

Management: FPT_APW_EXT.1

The following actions could be considered for the management functions in FMT:
- No management functions.

Audit: FPT_APW_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- No audit necessary.

FPT_APW_EXT.1: Protection of Administrator Passwords

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPT_APW_EXT.1.1: ***The TSF shall store passwords in non-plaintext form.***

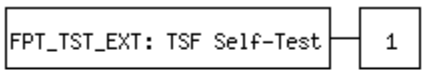
FPT_APW_EXT.1.2: ***The TSF shall prevent the reading of plaintext passwords.***

5.4.3 TSF Self-Test (FPT_TST_EXT)

Family behavior

Components in this family address the requirements for self-testing the TSF for selected correct operation.

Component levelling



Self-Test requires a suite of self-tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

Management: FPT_TST_EXT.1

The following actions could be considered for the management functions in FMT:
- No management functions.

Audit: FPT_TST_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:
- Indication that TSF self-test was completed
- Failure of self-test

FPT_TST_EXT.1: TSF Testing

Hierarchical to:

No other components.

Dependencies:

No dependencies.

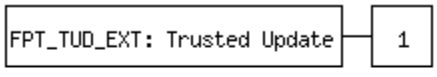
FPT_TST_EXT.1.1: ***The TSF shall run a suite of the following self-tests [selection: during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]] to demonstrate the correct operation of the TSF: [assignment: list of self-tests run by the TSF]***

5.4.4 Trusted Update (FPT_TUD_EXT)

Family behavior

Components in this family address the requirements for updating the TOE firmware and/or software.

Component levelling



Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

Management: FPT_TUD_EXT.1

The following actions could be considered for the management functions in FMT:
- Ability to update the TOE and to verify the updates

- Ability to update the TOE and to verify the updates using the digital signature capability (FCS_COP.1/SigGen)
- Ability to update the TOE and to verify the updates using the digital signature capability prior to installing those updates

Audit: FPT_TUD_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- Initiation of the update process.
- Any failure to verify the integrity of the update

FPT_TUD_EXT.1: Trusted Update

Hierarchical to:

No other components.

Dependencies:

FCS_COP.1/SigGen or FCS_COP.1/Hash

FPT_TUD_EXT.1.1: ***The TSF shall provide [assignment: Administrators] the ability to query the currently executing version of the TOE firmware/software and [selection: the most recently installed version of the TOE firmware/software, no other TOE firmware/software version]***

FPT_TUD_EXT.1.2: ***The TSF shall provide [assignment: Administrators] the ability to manually initiate updates to TOE firmware/software and [selection: support automatic checking for updates, support automatic updates, no other update mechanism]***

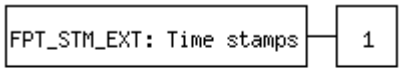
FPT_TUD_EXT.1.3: ***The TSF shall provide means to authenticate firmware/software updates to the TOE using a [selection: digital signature mechanism, published hash] prior to installing those updates.***

5.4.5 Time stamps (FPT_STM_EXT)

Family behavior

Components in this family extend FPT_STM requirements by describing the source of time used in timestamps.

Component levelling



Reliable Time Stamps is hierarchic to FPT_STM.1: it requires that the TSF provide reliable time stamps for TSF and identifies the source of the time used in those timestamps.

Management: FPT_STM_EXT.1

The following actions could be considered for the management functions in FMT:
- Management of the time
- Administrator setting of the time.

Audit: FPT_STM_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- Discontinuous changes to the time.

FPT_STM_EXT.1: Reliable Time Stamps

Hierarchical to:

No other components.

Dependencies:

No dependencies.

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

FPT_STM_EXT.1.2: ***The TSF shall [selection: allow the Security Administrator to set the time, synchronise time with an NTP server]***

5.5 Class FTA: TOE access

This family specifies functional requirements for controlling the establishment of a user's session.

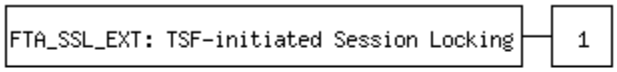
FTA class is extended in order to add the family FTA_SSL_EXT. This class includes requirements for authenticated sessions, considering different requirements for local and remote user sessions.

5.5.1 TSF-initiated Session Locking (FTA_SSL_EXT)

Family behavior

Components in this family address the requirements for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.
The extended FTA_SSL_EXT family is based on the FTA_SSL family.

Component levelling



TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.

Management: FTA_SSL_EXT.1

The following actions could be considered for the management functions in FMT:
- Specification of the time of user inactivity after which lock-out occurs for an individual user.

Audit: FTA_SSL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
- Any attempts at unlocking an interactive session.

FTA_SSL_EXT.1: TSF-initiated Session Locking

Hierarchical to:

No other components.

Dependencies:

FIA_UAU.1

FTA_SSL_EXT.1.1: ***The TSF shall, for local interactive sessions [selection: lock the session - disable any activity of the Administrator's data access/display devices other than unlocking the session, and requiring that the Administrator re-authenticate to the TSF prior to unlocking the session, terminate the session] after a Security Administrator-specified time period of inactivity.***

6 Security Requirements

This section defines the Security functional requirements (SFRs) and the Security assurance requirements (SARs) that fulfill the TOE. Assignment, selection, iteration and refinement operations have been made, adhering to the following conventions:

- Assignments. They appear between square brackets. The word “assignment” is maintained and the resolution is presented in ***boldface, italic and blue color.***
- Selections. They appear between square brackets. The word “selection” is maintained and the resolution is presented in ***boldface, italic and blue color.***
- Iterations. It includes “/” and an “identifier” following requirement identifier that allows to distinguish the iterations of the requirement. Example: FCS_COP.1/XXX.
- Refinements: the text where the refinement has been done is shown ***bold, italic, and light red color.*** Where part of the content of a SFR component has been removed, the removed text is shown in ~~***bold, italic, light red color and crossed out.***~~

6.1 Security Functional Requirements

6.1.1 FAU: Security audit

6.1.1.1 FAU_GEN.1: Audit data generation

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

a) Start-up and shutdown of the audit functions;

b) All auditable events for the ***[selection: not specified]*** level of audit; and

c) ***[assignment: All administrative actions comprising:***
- Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- Starting and stopping services.

Specifically defined auditable events listed in the table presented in the next application note.] .

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, ***[assignment: information specified in column three of the table presented in the next application note.]*** .

Application Note

The following table presents the list of auditable events.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.

FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of the identification and authentication mechanism	Origin of the attempt (e.g. IP address).
FIA_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g. IP address).
FIA_UAU.7	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	All management activities of TSF data.	None.

FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on **FPT_STM_EXT.1**)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g. IP address).
FTA_SSL_EXT.1	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the	None.

	session locking mechanism.	
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failures of the trusted path functions.	None.
FAU_STG.1	None.	None.
FAU_STG.3	Low storage space for audit events.	None.
FCS_SSHC_EXT.1	Failure to establish an SSH session.	Reason for failure.
FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
FCS_TLSC_EXT.1		Reason for failure.

	Failure to establish a TLS Session.	
FCS_TLSC_EXT.2	Failure to establish a TLS Session.	Reason for failure.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate.	Reason for failure.
FMT_MOF.1/Services	Starting and stopping of services.	None.
FMT_MOF.1/Functions	Modification of the behaviour of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full.	None.
FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	None.

Table 9 Security Functional Requirements and Auditable Events

Application Note

Audit functionality is enabled by default. The auditing functionality cannot be disabled.

Application Note

The TOE does not support using reset command to reset password directly, but it can modify password in the following way: re-create local-user or change local-user password.

6.1.1.2 FAU_GEN.2: User identity association

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.1.1.3 FAU_STG.1: Protected audit trail storage

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to ***[selection: prevent]*** unauthorised modifications to the stored audit records in the audit trail.

6.1.1.4 FAU_STG.3: Action in case of possible audit data loss

FAU_STG.3.1 The TSF shall ***[assignment: generate a warning to inform the Administrator]*** if the audit trail exceeds ***[assignment: the local audit trail storage capacity]*** .

Application Note

The audit data is stored in the CF card of the non-TOE hardware platform.

6.1.1.5 FAU_STG_EXT.1: Protected Audit Event Storage

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself. In addition ***[selection: TOE shall consist of a single standalone component that stores audit data locally].***

Application Note

The TOE consists only in software. The storage is done by the TOE software in the local non-TOE hardware platform in which the TOE runs.

FAU_STG_EXT.1.3 The TSF shall ***[selection: overwrite previous audit records according to the following rule: [assignment: overwrite the oldest log information always]]*** when the local storage space for audit data is full.

6.1.2 FCS: Cryptographic support

6.1.2.1 FCS_CKM.1: Cryptographic key generation

FCS_CKM.1.1 *The TSF shall generate* ***asymmetric*** *cryptographic keys in accordance with a specified cryptographic key generation algorithm* ***[assignment: RSA scheme and ECC schemes]*** and specified cryptographic key sizes ***[assignment: 2048 bits or greater (RSA) and P-256, P-384 and P-521 (ECC)]*** that meet the following: ***[assignment: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 and FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 (ECC)]***.

Application Note

The compliance to FIPS PUB 186-4 is limited to Appendix B.3, in particular:

- Generation of random primes (B.3.2 and B.3.3).
- Generation of random primes with conditions (B.3.4, B.3.5 and B.3.6).

6.1.2.2 FCS_CKM.2: Cryptographic key distribution

FCS_CKM.2.1 *The TSF shall* ~~***distribute cryptographic keys***~~ ***perform cryptographic key establishment*** *in accordance with a specified cryptographic key* ~~***distribution***~~ ***establishment*** *method* ***: [assignment:***

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";] .

~~***that meets the following: [assignment: list of standards].***~~

6.1.2.3 FCS_CKM.4: Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method ***[assignment: For plaintext keys in volatile storage, the destruction shall be executed by a single overwrite consisting of zeroes;***

- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that logically addresses the storage location of the key and performs a single, overwrite consisting of a new value of the key.] that meets the following: ***[assignment: No standard]***.

6.1.2.4 FCS_COP.1/DataEncryption: Cryptographic operation

FCS_COP.1.1/DATAENCRYPTION The TSF shall perform ***[assignment: encryption/decryption]*** in accordance with a specified cryptographic algorithm ***[assignment: AES used in GCM mode]*** and cryptographic key sizes ***[assignment: 128 bits, 256 bits]*** that meet the following: ***[assignment: AES as specified in ISO 18033-3, GCM as specified in ISO 19772]***

Application Note

The compliance to ISO/IEC 18033-3 is limited to section 5.2

Application Note

The compliance to ISO/IEC 19772 is limited to Annex A, section A.8.

6.1.2.5 FCS_COP.1/SigGen: Cryptographic operation

FCS_COP.1.1/SIGGEN The TSF shall perform ***[assignment: cryptographic signature services (generation and verification)]*** in accordance with a specified cryptographic algorithm ***[assignment: RSA Digital Signature Algorithm]*** and cryptographic key sizes ***[assignment: 3072 bits, 4096 bits]*** that meet the following: ***[assignment: For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3]***.

Application Note

The compliance to ISO/IEC 9796-2 is limited to:

- Digital Signature Scheme 2 (section 9 of the standard).
- Digital Signature Scheme 3 (section 10 of the standard).

6.1.2.6 FCS_COP.1/Hash: Cryptographic operation

FCS_COP.1.1/HASH The TSF shall perform ***[assignment: cryptographic hashing service]*** in accordance with a specified cryptographic algorithm ***[assignment: SHA-256, SHA-384, SHA-512]*** *and* ~~***cryptographic***~~ ~~***key***~~ ***message digest*** *sizes* ***[assignment: 256, 384, 512] bits*** *that meet the following:* ***[assignment: ISO/IEC 10118-3:2004]***.

Application Note

The compliance to ISO/IEC 10118-3:2004 is limited to:

- Section 10 Dedicated Hash-Function 4, which defines SHA-256.
- Section 12 Dedicated Hash-Function 6, which defines SHA-384.

6.1.2.7 FCS_COP.1/KeyedHash: Cryptographic operation

FCS_COP.1.1/KEYEDHASH The TSF shall perform ***[assignment: keyed-hash message authentication]*** in accordance with a specified cryptographic algorithm ***[assignment: HMAC-SHA-256]*** *and* ~~***cryptographic***~~ ~~***key***~~ ~~***sizes***~~ ***message digest size*** ***[assignment: 256] bits*** *that meet the following:* ***[assignment: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"]***.

Application Note

The compliance to ISO/IEC 9797-2 2011, Section 7 is limited to the definition of the algorithm HMAC-SHA-256.

6.1.2.8 FCS_RBG_EXT.1: Random Bit Generation

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using ***[selection: Hash_DRBG (any)]***

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from ***[selection: [assignment: 1] platform-based noise source]*** with a minimum of ***[selection: 256 bits]*** of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

Application Note

The compliance to ISO/IEC 18031:2011 is limited to:

- Section 7.2 Deterministic Random Bit Generator.
- Section 9. Overview and requirements for a deterministic random bit generator.
- C.2.1 Hash-function DRBG(Hash_DRBG)

6.1.2.9 FCS_SSHC_EXT.1: SSH Client

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, ***[selection: 6668]***

Application Note

The compliance to the RFC 4251 affecting the evaluated configuration includes only the general architecture of the SSH protocol, i.e. transport, confidentiality, data integrity, key exchange and authentication protocols.

Application Note

The compliance to the RFC 4252 is limited to public-key based and password-based authentication (as per FCS_SSHC_EXT.1.2).

Application Note

All required algorithms required by RFC 4253 are supported.

In addition, the parts of this RFC that are relevant to the scope of the TSF in terms of evaluated configuration are the following:

- Maximum packet length (as per FCS_SSHC_EXT.1.3).
- Encryption algorithms used during transport (as per FCS_SSHC_EXT.1.4).
- Supported public-key algorithms (as per FCS_SSHC_EXT.1.5).
- Transport data integrity algorithms (as per FCS_SSHC_EXT.1.6).
- Key exchange methods (as per FCS_SSHC_EXT.1.7).
- Session re-key requirements (as per FCS_SSHC_EXT.1.8).
- Host Keys (as per FCS_SSHC_EXT.1.9).

FCS_SSHC_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, ***[selection: password-based]***

FCS_SSHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than ***[assignment: 262144]*** bytes in an SSH transport connection are dropped.

Application Note

The TOE drops packets greater than 256 KB in an SSH transport connection. Packets of size greater than 35000 bytes and smaller than 256 KB are not dropped because of that the TOE may support uncompressed big certificates.

FCS_SSHC_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: ***[selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM]***

FCS_SSHC_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses ***[selection: rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521]*** as its public key algorithm(s) and rejects all other public key algorithms

Application Note

The SSH transport implementation uses rsa-sha2-256 and rsa-sha2-512 as its public key algorithm. And it doesn't support x509v3.

FCS_SSHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses ***[selection: hmac-sha2-256]*** as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7 The TSF shall ensure that ***[selection: ecdh-sha2-nistp256]*** and ***[selection: ecdh-sha2-nistp384, ecdh-sha2-nistp521]*** are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

FCS_SSHC_EXT.1.9 The TSF 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 and ***[selection: no other methods]*** as described in RFC 4251 section 4.1.

6.1.2.10 FCS_SSHS_EXT.1: SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, ***[selection: 6668]***

Application Note

The compliance to the RFC 4251 affecting the evaluated configuration includes only the general architecture of the SSH protocol, i.e. transport, confidentiality, data integrity, key exchange and authentication.

Application Note

The compliance to the RFC 4252 is limited to public-key based and password-based authentication (as per FCS_SSHS_EXT.1.2).

Application Note

The parts of the RFC 4253 that are relevant to the scope of the TSF in terms of evaluated configuration are the following:

- Maximum packet length (as per FCS_SSHS_EXT.1.3)
- Encryption algorithms used during transport (as per FCS_SSHS_EXT.1.4)
- Supported public-key based authentication algorithms (as per FCS_SSHS_EXT.1.5).
- Transport data integrity algorithms (as per FCS_SSHS_EXT.1.6).
- Key exchange methods (as per FCS_SSHS_EXT.1.7).
- Session re-key requirements (as per FCS_SSHS_EXT.1.8).

FCS_SSHS_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, ***[selection: password-based]***

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than ***[assignment: 262144]*** bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: ***[selection: AEAD_AES_128_GCM, AEAD_AES_256_GCM]***

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses ***[selection: rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521]*** as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses ***[selection: hmac-sha2-256]*** as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that ***[selection: ecdh-sha2-nistp256]*** and ***[selection: ecdh-sha2-nistp384, ecdh-sha2-nistp521]*** are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed

6.1.2.11 FCS_TLSC_EXT.1: TLS Client Protocol

FCS_TLSC_EXT.1.1 The TSF shall implement ***[selection: TLS 1.2 (RFC 5246)]*** and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites ***[selection: TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289, TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289]***

Application Note

The compliance of the TSF with RFC 5246 in terms of evaluated configuration includes support of only the ciphersuites defined in FCS_TLSC_EXT.1.1. The compliance with this RFC is also limited by the TSF defined in FCS_TLSC_EXT.1.2, FCS_TLSC.1.3 and FCS_TLSC.1.4.

Application Note

The compliance to the RFC 5288 is limited to the ciphersuites supported by the TOE. The ciphersuites included in the evaluated configuration are limited by FCS_TLSC_EXT.1.1.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier ***matches [selection: the reference identifier per RFC 6125 section 6]***.

Application Note

The reference identifier is established by the user and by an application (a parameter of an API). Based on a singular reference identifier's source domain and application service type, the client establishes all reference identifiers including a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

FCS_TLSC_EXT.1.3 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also ***[selection: Not implement any administrator override mechanism]***

FCS_TLSC_EXT.1.4 The TSF shall ***[selection: not present the Supported Elliptic Curves Extension]*** in the Client Hello.

Application Note

The ciphersuites with elliptic curves were not selected in FCS_TLSC_EXT.1.1. The TSF doesn't support this ciphersuite.

6.1.2.12 FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication

FCS_TLSC_EXT.2.1 The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.
Application Note 55

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.

6.1.3 FIA: Identification and authentication

6.1.3.1 FIA_AFL.1: Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when ***[selection: an administrator configurable positive integer within [assignment: 3 to 5]]*** unsuccessful authentication attempts occur related to ***[assignment: Administrators attempting to authenticate remotely]*** .

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been ***[selection: met]*** , the TSF shall ***[assignment: prevent the offending remote Administrator from successfully authenticating until unlock is taken by a local Administrator; prevent the offending remote Administrator from successfully authenticating until an Administrator defined time period has elapsed]*** .

6.1.3.2 FIA_UAU.7: Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only ***[assignment: obscured feedback]*** *to the user while the authentication is in* ~~***progress.***~~ ***progress at the local console.***

6.1.3.3 FIA_PMG_EXT.1: Password Management

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: ***[selection: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: "-", "+", "=", "[", "]", "{", "}", "|", "\", ",", ".", "/", "<", ">", ";", "'", ":", """]]***

b) Minimum password length shall be configurable to between ***[assignment: 8]*** and ***[assignment: 128]*** characters.

6.1.3.4 FIA_UIA_EXT.1: User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

a) Display the warning banner in accordance with FTA_TAB.1

b) ***[selection: no other actions]***

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

Application Note

Only a banner will show to the user or IT entity and no services are available before authentication.

6.1.3.5 FIA_X509_EXT.1/Rev: X.509 Certificate Validation

FIA_X509_EXT.1.1/Rev The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using ***[selection: a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3]***
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

Application Note

The aspects of the RFC 5280 that are relevant to the TSF are the following:

- Validation of trusted certification chain.
- Verification of fail in validation of incomplete certificate chain.
- Handling of revoked certificates.
- Revocation of peer certificates and peer intermediate certificates.
- Validation of the KeyUsage cRLSign field when employing CRLs.

FIA_X509_EXT.1.2/Rev The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

Application Note

Revocation status is verified using CRLs. TLS requires that certificates are used and this use requires that the extendedKeyUsage rules are verified. The validation is expected to end in a trusted root CA certificate in a root store managed by the platform. The certificate path must end in a trusted root CA certificate otherwise it will be judged invalid.

6.1.3.6 FIA_X509_EXT.2: X509 Certificate Authentication

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for ***[selection: TLS]*** , and ***[selection: code signing for system software updates]***

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall ***[selection: not accept the certificate]***

6.1.3.7 FIA_UAU_EXT.2: Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local ***[selection: password-based [assignment: other authentication mechanism(s)]]*** authentication mechanism to perform local administrative user authentication.

6.1.4 FMT: Security management

6.1.4.1 FMT_MOF.1/ManualUpdate: Management of security functions behaviour

FMT_MOF.1.1/MANUALUPDATE The TSF shall restrict the ability to ***[selection: enable]*** the functions ***[assignment: to perform manual updates]*** to ***[assignment: Security Administrators]***.

6.1.4.2 FMT_MOF.1/Functions: Management of security functions behaviour

FMT_MOF.1.1/FUNCTIONS The TSF shall restrict the ability to ***[selection: determine the behaviour of]*** the functions ***[assignment: transmission of audit data to an external IT entity]*** to ***[assignment: Security Administrators]***.

6.1.4.3 FMT_MOF.1/Services: Management of security functions behaviour

FMT_MOF.1.1/SERVICES The TSF shall restrict the ability to ***[selection: disable, enable]*** ~~***the functions***~~ ***services [assignment: Start and stop]*** to ***[assignment: Security Administrators]***.

Application Note

The following security services can be enabled and disabled by Security Administrators: SSH, TLS, NTP and SYSLOG.

6.1.4.4 FMT_MTD.1/CoreData: Management of TSF data

FMT_MTD.1.1/COREDATA The TSF shall restrict the ability to ***[selection: [assignment: manage]]*** the ***[assignment: TSF data]*** to ***[assignment: Security Administrators]***.

6.1.4.5 FMT_MTD.1/CryptoKeys: Management of TSF data

FMT_MTD.1.1/CRYPTOKEYS The TSF shall restrict the ability to ***[selection: [assignment: manage]]*** the ***[assignment: cryptographic keys]*** to ***[assignment: Security Administrators]***.

6.1.4.6 FMT_SMF.1: Specification of Management Functions

- **FMT_SMF.1.1** The TSF shall be capable of performing the following management functions: ***[assignment: - Ability to administer the TOE locally and remotely;***
  - Ability to configure the access banner;
  - Ability to configure the session inactivity time before session termination or locking;
  - Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
  - Ability to configure the authentication failure parameters for FIA_AFL.1;

  - Ability to start and stop services

  - Ability to configure audit behavior; (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);

  - Ability to modify the behaviour of the transmission of audit data to an external IT entity;

  - Ability to manage the cryptographic keys;

  - Ability to configure the cryptographic functionality;

  - Ability to configure thresholds for SSH rekeying ;

  - Ability to re-enable an Administrator account;

  - Ability to set the time which is used for time-stamps;

  - Ability to manage the TOE's trust store and designate X.509.v3 certificates as trust anchors;].

6.1.4.7 FMT_SMR.2: Restrictions on security roles

FMT_SMR.2.1 The TSF shall maintain the roles: ***[assignment: Security Administrator]***.

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions ***[assignment: The Security Administrator role shall be able to administer the TOE locally; The Security Administrator role shall be able to administer the TOE remotely]*** are satisfied.

6.1.5 FTA: TOE access

6.1.5.1 FTA_SSL.3: TSF-initiated termination

FTA_SSL.3.1 *The TS shall terminate ~~an~~ **a remote** interactive session after a **[assignment: Security Administrator-configurable time interval of session inactivity]**.*

6.1.5.2 FTA_SSL.4: User-initiated termination

FTA_SSL.4.1 *The TSF shall allow ~~user-initiated~~ **Administrator-initiated** termination of the ~~user's~~ **Administrator's** own interactive session.*

6.1.5.3 FTA_TAB.1: Default TOE access banners

FTA_TAB.1.1 *Before establishing an **administrative** user session, the TSF shall display a **Security Administrator specified** advisory **notice and consent** ~~warning message~~ regarding ~~unauthorised~~ use of the TOE.*

6.1.5.4 FTA_SSL_EXT.1: TSF-initiated Session Locking

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions ***[selection: terminate the session]*** after a Security Administrator-specified time period of inactivity.

6.1.6 FTP: Trusted path/channels

6.1.6.1 FTP_ITC.1: Inter-TSF trusted channel

FTP_ITC.1.1 *The TSF shall **be capable of using TLS to** provide a communication channel between itself and ~~another~~ ~~trusted~~ **authorized** IT ~~product~~ **entities supporting the following capabilities: audit server;** 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 ~~or disclosure.~~ **of the channel data.***

FTP_ITC.1.2 The TSF shall permit ***[selection: the TSF]*** *to initiate communication via the trusted channel.*

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for ***[assignment: audit service]***.

6.1.6.2 FTP_TRP.1: Trusted path

FTP_TRP.1.1 *The TSF shall **be capable of using SSH to** provide a communication path between itself and **authorized** **[selection: remote]** ~~users~~ **administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **[selection: disclosure [assignment: and provides detection of modification of the channel data]]**.*

FTP_TRP.1.2 The TSF shall permit ***[selection: remote administrators]*** to initiate communication via the trusted path.

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

6.1.7 FPT: Protection of the TSF

6.1.7.1 FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

6.1.7.2 FPT_APW_EXT.1: Protection of Administrator Passwords

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

6.1.7.3 FPT_TST_EXT.1: TSF Testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests ***[selection: during initial start-up (on power on)]*** to demonstrate the correct operation of the TSF: ***[assignment: integrity of the firmware and software (software digital signature), the correct operation of cryptographic functions]***

6.1.7.4 FPT_TUD_EXT.1: Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide ***[assignment: Security Administrators]*** the ability to query the currently executing version of the TOE firmware/software and ***[selection: the most recently installed version of the TOE firmware/software]***

FPT_TUD_EXT.1.2 The TSF shall provide ***[assignment: Security Administrators]*** the ability to manually initiate updates to TOE firmware/software and ***[selection: no other update mechanism]***

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a ***[selection: digital signature]*** prior to installing those updates.

6.1.7.5 FPT_STM_EXT.1: Reliable Time Stamps

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

FPT_STM_EXT.1.2 The TSF shall ***[selection: synchronise time with an NTP server]***

6.2 Security Assurance Requirements

The development and the evaluation of the TOE shall be done in accordance to the following security assurance requirements: **EAL2 augmented with ALC_FLR.2**

The following table shows the assurance requirements by reference the individual components in [CC31R5P3]

Assurance Class	Assurance Components
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims ASE_ECD.1 Extended components definition

Assurance Class	Assurance Components
	ASE_INT.1 ST introduction ASE_OBJ.2 Security objectives ASE_REQ.2 Derived security requirements ASE_SPD.1 Security problem definition ASE_TSS.1 TOE summary specification
ALC: Life-cycle support	ALC_CMC.2 Use of a CM system ALC_CMS.2 Parts of the TOE CM coverage ALC_DEL.1 Delivery procedures ALC_FLR.2 Flaw reporting procedures
ADV: Development	ADV_ARC.1 Security architecture description ADV_FSP.2 Security-enforcing functional specification ADV_TDS.1 Basic design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance AGD_PRE.1: Preparative procedures
ATE: Tests	ATE_COV.1 Evidence of coverage ATE_FUN.1 Functional testing ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

Table 10 *Security Assurance Requirements*

6.3 Security Requirements Rationale

6.3.1 Necessity and sufficiency analysis

SFR / TOE Security Objective	O.ADMIN_AUTH	O.STRONG_CRYPTO	O.TRUSTED_COMM	O.STRONG_AUTHENTICATION_ENDPOINT	O.SECURE_UPDATES	O.ACTIVITY_AUDIT	O.CRYPTO_KEY_PROTECTION	O.PASSWORD_PROTECTION	O.SELF_TEST	O.BANNER
FAU_GEN.1						X				

SFR / TOE Security Objective	O.ADMIN_AUTH	O.STRONG_CRYPTO	O.TRUSTED_COMM	O.STRONG_AUTHENTICATION_ENDPOINT	O.SECURE_UPDATES	O.ACTIVITY_AUDIT	O.CRYPTO_KEY_PROTECTION	O.PASSWORD_PROTECTION	O.SELF_TEST	O.BANNER
FAU_GEN.2						X				
FAU_STG_EXT.1						X				
FAU_STG.3						X				
FAU_STG.1						X				
FCS_CKM.1		X								
FCS_CKM.2		X								
FCS_CKM.4							X			
FCS_COP.1/DataEncryption		X								
FCS_COP.1/SigGen		X								
FCS_COP.1/Hash		X								
FCS_COP.1/KeyedHash		X								
FCS_SSHC_EXT.1			X							
FCS_SSHS_EXT.1			X							
FIA_AFL.1								X		
FIA_PMG_EXT.1								X		
FIA_UIA_EXT.1	X									
FIA_UAU_EXT.2	X									

SFR / TOE Security Objective	O.ADMIN_AUTH	O.STRONG_CRYPTO	O.TRUSTED_COMM	O.STRONG_AUTHENTICATION_ENDPOINT	O.SECURE_UPDATES	O.ACTIVITY_AUDIT	O.CRYPTO_KEY_PROTECTION	O.PASSWORD_PROTECTION	O.SELF_TEST	O.BANNER
FIA_UAU.7								X		
FIA_X509_EXT.1/Rev			X		X					
FIA_X509_EXT.2			X		X					
FMT_MOF.1/ManualUpdate					X					
FMT_MOF.1/Functions	X					X				
FMT_MOF.1/Services	X									
FMT_MTD.1/CoreData	X									
FMT_MTD.1/CryptoKeys							X			
FMT_SMF.1	X	X			X	X	X			
FMT_SMR.2	X									
FPT_SKP_EXT.1							X			
FPT_APW_EXT.1								X		
FPT_TUD_EXT.1					X					
FPT_STM_EXT.1						X				
FTA_SSL_EXT.1	X									
FTA_SSL.3	X									
FTA_SSL.4	X									

SFR / TOE Security Objective	O.ADMIN_AUTH	O.STRONG_CRYPTO	O.TRUSTED_COMM	O.STRONG_AUTHENTICATION_ENDPOINT	O.SECURE_UPDATES	O.ACTIVITY_AUDIT	O.CRYPTO_KEY_PROTECTION	O.PASSWORD_PROTECTION	O.SELF_TEST	O.BANNER
FTA_TAB.1	X									X
FTP_ITC.1			X	X						
FTP_TRP.1	X		X	X						
FCS_RBG_EXT.1		X								
FPT_TST_EXT.1									X	
FCS_TLSC_EXT.1			X							
FCS_TLSC_EXT.2			X							

Table 3 SFRs / TOE Security Objectives coverage

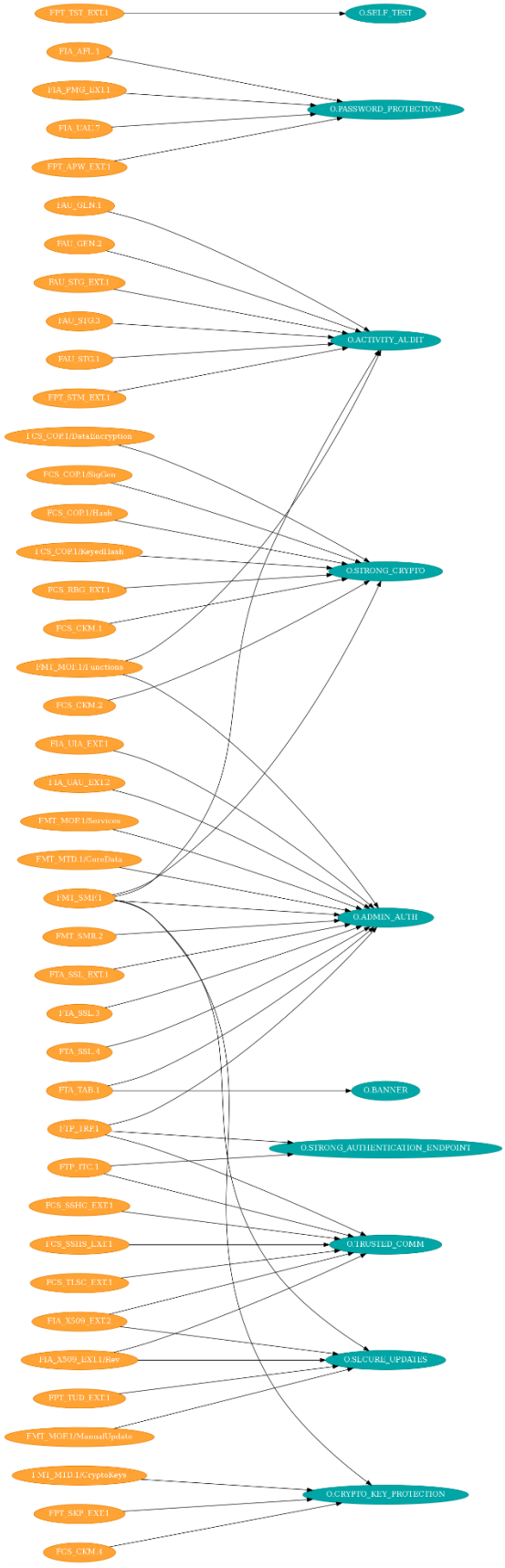


Figure 4 Mapping of SFRs to TOE Security Objectives

6.3.2 Security Requirement Sufficiency

O.ADMIN_AUTH: The Administrator role is defined in **FMT_SMR.2** and along with **FMT_SMF.1**, they support this security objective by defining the TOE management capabilities available to administrators.

FMT_MTD.1/CoreData restricts the ability to manage the TSF to security administrators.

FMT_MOF.1/Functions restricts the ability to determine the feature of the transmission of audit data to an external IT entity. On the other hand, **FMT_MOF.1/Services** restricts the ability to disable or enable the services of the TOE to the security administrators.

FIA_UIA_EXT.1 supports this security objective by preventing any action before the identification and authentication process. **FTA_TAB.1** supports this security objective by showing an advisory notice and consent warning message is the only action allowed before the identification and authentication process occurs.

The administrators' authentication process is defined in **FIA_UAU_EXT.2** and consists on providing a local password-based authentication mechanism to perform local administrative user authentication.

This security objective is intended to perform the closure of authentication sessions after a defined period of inactivity. **FTA_SSL_EXT.1** ensures this feature for local sessions, **FTA_SSL.4** for all interactive sessions, and by **FTA_SSL.3** for remote sessions.

Another goal of this security objective is that the authentication of the TOE using passwords was through a secure communication channel. This task is covered by **FTP_TRP.1**.

O.STRONG_CRYPTO: This security objective is supported by the following security requirements, that provide use of robust cryptographic algorithms, compliant to industry approved standards:

- Requirements for key generation and key distribution are set in **FCS_CKM.1** and **FCS_CKM.2**, respectively.

- Requirements for use of cryptographic schemes are set in **FCS_COP.1/DataEncryption**, **FCS_COP.1/SigGen**, **FCS_COP.1/Hash**, and **FCS_COP.1/KeyedHash**.

- Requirements for random bit generation to support key generation and secure protocols are set in **FCS_RBG_EXT.1**.

- **FMT_SMF.1** is intended to establish the management of cryptographic functions by a security administrator.

O.TRUSTED_COMM: This security objective is intended to implement secure channels that use standardized tunneling protocols to protect the critical network traffic. The following security requirements support this security objective:

- **FTP_ITC.1** supports this security objective by providing a communication channel between itself and authorized IT entities by using the TLS protocol. **FTP_TRP.1** is able to provide this channel by using SSH.

- **FCS_SSHC_EXT.1**, **FCS_SSHS_EXT.1**, **FCS_TLSC_EXT.1** and support this security objective for the use of secure communication protocols by using SSH Client, SSH Server Protocol, and TLS Client Protocols.

- Requirements for the use of public-key certificates to support secure protocols are supported by **FIA_X509_EXT.1/Rev**, **FIA_X509_EXT.2**.

All the above-mentioned security requirements support this security objective by providing communication security in terms of confidentiality, integrity, and protection.

O.STRONG_AUTHENTICATION_ENDPOINT: **FTP_ITC.1** and **FTP_TRP.1** support this security objective by providing assured identification of its endpoints (using TLS and SSH respectively) and protection of the channel data from disclosure and detection of modification of the channel data.

O.SECURE_UPDATES: **FPT_TUD_EXT.1** supports this security objective by allowing administrators to query the current TOE version of the TOE firmware/software and manually initiate the installation of updates, that need to be authenticated using digital signature before installing.

FIA_X509_EXT.1/Rev and **FIA_X509_EXT.2** support this security objective by defining a set of rules to validate the authenticity of the certificate using secure and robust cryptographic algorithms.

FMT_SMF.1 supports this security objective by providing a set of management functions, which includes the ability to update the TOE and to verify the updates using digital signature capabilities. **FMT_MOF.1/ManualUpdate** likewise supports this security objective by restricting the ability of this feature to security administrators.

O.ACTIVITY_AUDIT: **FAU_GEN.1** supports this security objective by generating audit records of a set of auditable events. Each audit event will be marked with precise timestamps, associated subject identity, and the outcome (success or failure) of the event. **FAU_GEN.2** supports this security objective by providing a relationship between the audit event and the identity of the user that causes the event. **FPT_STM_EXT.1** is intended to provide reliable time stamps for the own TOE's use.

FAU_STG.1 supports this security target by protecting the stored audit records from unauthorized deletion. This security requirement is also intended to prevent unauthorized modifications to the stored audit records in the audit trail.

FAU_STG_EXT.1 is intended to provide the ability to transmit the generated audit data to an external IT entity using a trusted channel by using secure communication protocols. This security requirement also provides a mechanism to overwrite the oldest log information always when the local storage space for audit data is full.

FAU_STG.3 provides a mechanism to generate a warning to inform the TOE's administrator if the audit trail exceeds the local audit trail storage capacity. This mechanism is intended to deal with the potential loss of locally stored audit records.

FMT_SMF.1 supports this security objective by providing a set of management functions, which includes the ability to configure audit behavior.

FMT_MOF.1/Functions likewise supports this security objective by restricting the ability to enable the transmission of audit data to an external IT entity to a security administrator.

O.CRYPTO_KEY_PROTECTION: **FPT_SKP_EXT.1** supports the protection of secret/private keys against compromise.

FCS_CKM.4 provides a mechanism to perform the key's destruction in a secure way that prevents key recovery from residual information.

FMT_SMF.1 supports this security objective by providing a set of management functions, which includes the ability to configure thresholds for SSH rekeying. **FMT_MTD.1/CryptoKeys** is intended to restrict the ability to manage cryptographic keys to security administrators.

O.PASSWORD_PROTECTION: **FIA_PMG_EXT.1** supports this security objective by providing the ability to establish password management capabilities based on password complexity (composed of any combination of upper and lower case letters numbers, and special characters) and minimum password length restriction.

Protection of password entry by providing obscured feedback is specified in **FIA_UAU.7**

FIA_AFL.1 supports this security objective by providing the ability to detect unsuccessful authentication attempts when administrators are attempting to authenticate remotely. This security requirement meets this security objective by locking the authentication of the user, which reaches the defined number of authentication attempts established.

FPT_APW_EXT.1 prevents the plaintext storage of passwords. Therefore, this security requirement also prevents the reading of plaintext passwords.

O.SELF_TEST: **FPT_TST_EXT.1** provides the necessary mechanisms to support this security objective by run a suite of tests during the initial start-up of the TOE to demonstrate its correct operation.

O.BANNER: This security objective is supported by **FTA_TAB.1** which requires that the TOE displays an advisory notice and consent about the use of the TOE to administrators before establishing an administrative user session.

6.3.3 SFR Dependency Rationale

6.3.3.1 Table of SFR dependencies

The following table lists the dependencies for each requirement, indicating how they have been satisfied. The abbreviation "h.a." indicates that the dependency has been satisfied by a SFR that is hierarchically above the required dependency.

SFR	Required	Fulfilled	Missing
FAU_GEN.1	FPT_STM.1	None	FPT_STM.1
FAU_GEN.2	FAU_GEN.1, FIA_UID.1	FAU_GEN.1	FIA_UID.1
FAU_STG_EXT.1	FAU_GEN.1, FTP_ITC.1	FAU_GEN.1, FTP_ITC.1	None
FAU_STG.3	FAU_STG.1	FAU_STG.1	None
FAU_STG.1	FAU_GEN.1	FAU_GEN.1	None
FCS_CKM.1	FCS_CKM.4, [FCS_CKM.2 or FCS_COP.1]	FCS_CKM.4, FCS_CKM.2	None

SFR	Required	Fulfilled	Missing
FCS_CKM.2	FCS_CKM.4, [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.4, FCS_CKM.1	None
FCS_CKM.4	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1	None
FCS_COP.1/DataEncryption	FCS_CKM.4, [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.4, FCS_CKM.1	None
FCS_COP.1/SigGen	FCS_CKM.4, [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.4, FCS_CKM.1	None
FCS_COP.1/Hash	FCS_CKM.4, [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.4, FCS_CKM.1	None
FCS_COP.1/KeyedHash	FCS_CKM.4, [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.4, FCS_CKM.1	None
FCS_SSHC_EXT.1	FCS_CKM.1, FCS_CKM.2, FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1	FCS_CKM.1, FCS_CKM.2, FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1	None
FCS_SSHS_EXT.1	FCS_CKM.1, FCS_CKM.2, FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1	FCS_CKM.1, FCS_CKM.2, FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1	None
FIA_AFL.1	FIA_UAU.1	None	FIA_UAU.1
FIA_PMG_EXT.1	None	None	None
FIA_UIA_EXT.1	FTA_TAB.1	FTA_TAB.1	None
FIA_UAU_EXT.2	None	None	None
FIA_UAU.7	FIA_UAU.1	None	FIA_UAU.1
FIA_X509_EXT.1/Rev	FIA_X509_EXT.2	FIA_X509_EXT.2	None
FIA_X509_EXT.2	FIA_X509_EXT.1	FIA_X509_EXT.1/Rev	None
FMT_MOF.1/ManualUpdate	FMT_SMR.1, FMT_SMF.1	FMT_SMR.2 (h.a. FMT_SMR.1), FMT_SMF.1	None

SFR	Required	Fulfilled	Missing
FMT_MOF.1/Functions	FMT_SMR.1, FMT_SMF.1	FMT_SMR.2 (h.a. FMT_SMR.1), FMT_SMF.1	None
FMT_MOF.1/Services	FMT_SMR.1, FMT_SMF.1	FMT_SMR.2 (h.a. FMT_SMR.1), FMT_SMF.1	None
FMT_MTD.1/CoreData	FMT_SMR.1, FMT_SMF.1	FMT_SMR.2 (h.a. FMT_SMR.1), FMT_SMF.1	None
FMT_MTD.1/CryptoKeys	FMT_SMR.1, FMT_SMF.1	FMT_SMR.2 (h.a. FMT_SMR.1), FMT_SMF.1	None
FMT_SMF.1	None	None	None
FMT_SMR.2	FIA_UID.1	None	FIA_UID.1
FPT_SKP_EXT.1	None	None	None
FPT_APW_EXT.1	None	None	None
FPT_TUD_EXT.1	[FCS_COP.1/SigGen or FCS_COP.1/Hash]	FCS_COP.1/SigGen, FCS_COP.1/Hash	None
FPT_STM_EXT.1	None	None	None
FTA_SSL_EXT.1	FIA_UAU.1	None	FIA_UAU.1
FTA_SSL.3	None	None	None
FTA_SSL.4	None	None	None
FTA_TAB.1	None	None	None
FTP_ITC.1	None	None	None
FTP_TRP.1	None	None	None
FCS_RBG_EXT.1	None	None	None
FPT_TST_EXT.1	None	None	None
FCS_TLSC_EXT.1	FCS_CKM.1, FCS_CKM.2, FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1 FIA_X509_EXT.1 FIA_X509_EXT.2	FCS_CKM.1, FCS_CKM.2, FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1 FIA_X509_EXT.1 FIA_X509_EXT.2	None
FCS_TLSC_EXT.2	FCS_CKM.1, FCS_CKM.2, FCS_COP.1/DataEncryption,	FCS_CKM.1, FCS_CKM.2,	None

SFR	Required	Fulfilled	Missing
	FCS_COP.1 /SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1, FCS_TLSC_EXT.1, FIA_X509_EXT.1, FIA_X509_EXT.2	FCS_COP.1/DataEncryption, FCS_COP.1 /SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1, FCS_TLSC_EXT.1, FIA_X509_EXT.1, FIA_X509_EXT.2	

Table 4 SFR Dependencies

6.3.3.2 Justification for missing dependencies

FAU_GEN.1 dependency on FPT_STM.1

FPT_STM_EXT.1 ensures the TOE capability to provide reliable time stamps for its own use. Therefore, FPT_STM.1 is not required.

FAU_GEN.2 dependency on FIA_UID.1

FIA_UIA_EXT.1 defines the list of actions allowed prior to requiring identification and requires administrative users to be identified before allowing other TSF-mediated actions. Because that SFR behaves in a way similar to FIA_UID.1 for this TOE, hence FIA_UID.1 is not required.

FIA_AFL.1 dependency on FIA_UAU.1

FIA_UIA_EXT.1 defines the list of actions allowed prior to requiring authentication and requires administrative users to be authenticated before allowing other TSF-mediated actions. Because that SFR behaves in a way similar to FIA_UAU.1 for this TOE, hence FIA_UAU.1 is not required.

FIA_UAU.7 dependency on FIA_UAU.1

FIA_UIA_EXT.1 defines the list of actions allowed prior to requiring authentication and requires administrative users to be authenticated before allowing other TSF-mediated actions. Because that SFR behaves in a way similar to FIA_UAU.1 for this TOE, hence FIA_UAU.1 is not required.

FMT_SMR.2 dependency on FIA_UID.1

FIA_UIA_EXT.1 defines the list of actions allowed prior to requiring identification and requires administrative users to be identified before allowing other TSF-mediated actions. Because that SFR behaves in a way similar to FIA_UID.1 for this TOE, hence FIA_UID.1 is not required.

FTA_SSL_EXT.1 dependency on FIA_UAU.1

FIA_UIA_EXT.1 defines the list of actions allowed prior to requiring authentication and requires administrative users to be authenticated before allowing other TSF-mediated actions. Because that SFR behaves in a way similar to FIA_UAU.1 for this TOE, hence FIA_UAU.1 is not required.

6.3.4 SAR Rationale

The TOE claims compliance to EAL2 augmented with ALC_FLR.2. The assurance level EAL2 indicates that the product is methodically designed, tested, and reviewed.

6.3.5 SAR Dependency Rationale

6.3.5.1 Table of SAR dependencies

SAR	Required	Fulfilled	Missing
ASE_CCL.1	ASE_INT.1, ASE_ECD.1, ASE_REQ.1	ASE_INT.1, ASE_ECD.1, ASE_REQ.2 (hierarchically above ASE_REQ.1)	None
ASE_ECD.1	None	None	None
ASE_INT.1	None	None	None
ASE_OBJ.2	ASE_SPD.1	ASE_SPD.1	None
ASE_REQ.2	ASE_OBJ.2, ASE_ECD.1	ASE_OBJ.2, ASE_ECD.1	None
ASE_TSS.1	ASE_INT.1, ASE_REQ.1, ADV_FSP.1	ASE_INT.1, ASE_REQ.2 (hierarchically above ASE_REQ.1), ADV_FSP.2 (hierarchically above ADV_FSP.1)	None
ALC_CMC.2	ALC_CMS.1	ALC_CMS.2 (hierarchically above ALC_CMS.1)	None
ALC_CMS.2	None	None	None
ALC_FLR.2	None	None	None
ADV_FSP.2	ADV_TDS.1	ADV_TDS.1	None
AGD_OPE.1	ADV_FSP.1	ADV_FSP.2 (hierarchically above ADV_FSP.1)	None
AGD_PRE.1	None	None	None
ATE_IND.2	ADV_FSP.2, AGD_OPE.1, AGD_PRE.1, ATE_COV.1, ATE_FUN.1	ADV_FSP.2, AGD_OPE.1, AGD_PRE.1, ATE_COV.1, ATE_FUN.1	None
AVA_VAN.2	ADV_ARC.1, ADV_FSP.2, ADV_TDS.1, AGD_OPE.1, AGD_PRE.1	ADV_ARC.1, ADV_FSP.2, ADV_TDS.1, AGD_OPE.1, AGD_PRE.1	None
ASE_SPD.1	None	None	None
ALC_DEL.1	None	None	None
ADV_ARC.1	ADV_FSP.1, ADV_TDS.1	ADV_FSP.2 (hierarchically above ADV_FSP.1), ADV_TDS.1	None

SAR	Required	Fulfilled	Missing
ADV_TDS.1	ADV_FSP.2	ADV_FSP.2	None
ATE_COV.1	ADV_FSP.2, ATE_FUN.1	ADV_FSP.2, ATE_FUN.1	None
ATE_FUN.1	ATE_COV.1	ATE_COV.1	None

Table 5 SAR dependencies

7 TOE Summary Specification

7.1 Security Audit

This section describes how the TOE meets each security functional requirement that belong to class FAU (defined in **[CC31R5P2]**) and that is listed in Section 6 of this ST.

Audit data generation TSF is defined by **FAU_GEN.1**. The TOE meets this SFR as described below:

- The TOE generates an audit record whenever an audited event occurs. The types of events that cause audit records to be generated include identification and authentication related events, and administrative events (the specific events and the contents of each audit record are listed in the table within the **FAU_GEN.1** SFR, (in the table of auditable events presented in the application note next to that SFR). Each of the events specified in the audit record is in enough detail to identify the user for which the event is associated (e.g. user identity, MAC address, IP address), when the event occurred, where the event occurred, the outcome of the event, and the type of event that occurred.
- The audit trail consists of the individual audit records; one audit record for each event that occurred. The audit record contains a lot of information, such as the type of event that occurred, and two percent sign (%%), which follows the device name. As noted above, the information includes at least all of the required information. Additional information can be configured and included if desired.
- Administrators have the ability to execute CLI command to generate, import or delete cryptographic keys, each command will generate a log and will be stored in log file (in the hardware platform). Starting and stopping the referred service will generate a log for audit.
- Date and time of the event, type of event, subject identity, and the outcome of the event are included in audit log.

TSF related to association of user identities to audit data is defined by **FAU_GEN.2**. The TOE meets this SFR as follows: each auditable event is associated with the user that triggered the event and as a result, they are traceable to a specific user. For example, a human user, user identity or related session ID would be included in the audit record. For the IT entity or device where the authentication event came from, the IP address, MAC address, host name, or other configured identification is presented. The security log of user account management shall include user name.

Protection of audit trail storage TSF is defined by **FAU_STG.1** which is met by the TOE as follows:

- Only the authorized administrators can monitor the logfile record, and operate the log files. The unauthorized users have no access to do those actions. And the actions of the authorized administrators will be logged.
- An administrator cannot alter audit records but can delete audit information records as a whole.

Protected audit event storage TSF is defined by **FAU_STG_EXT.1** and the TOE meets this SFR as described below:

- The TOE supports to export syslog records to a specified, external syslog server. The TOE protects communications with an external syslog server via TLS. The TOE stores audit records on a CF card (located in the non-TOE hardware platform) whenever it is connected with syslog server or not. The transmission of audit information to an external syslog server can be done in real-time.
- The size of an information file is configurable by the administrator with value 4M/8M/16M/32M bytes. The default maximum size of each information file is 8 MB. When the size of an information file exceeds the configured maximum size, the information file is compressed into a smaller file in standard log_slot ID_time.log.zip format. The maximum quantity of compressed files is configurable by the administrator with a value ranging from 3 to 500. A maximum of 200 files can be stored on a device by default. The unauthorized users are disallowed to handle the audit records.
- The logs are saved to flash memory (internal CF card in the non-TOE platform) so records can't be lost in case of failures or restarts. The log buffer is circular, so newer messages overwrite older messages after the buffer is full. Administrators are instructed to monitor the log buffer using the "show logging privileged CLI" command to view the audit records. The first message displayed is the newest message in the buffer. There are other associated commands to clear the buffer, to reset log buffer, etc. The size of the log buffer can be configured by users with sufficient privileges.
- When the local audit data store in CF card exceeds the maximum allowed size of log file storage, the system deletes oldest compressed files to save the latest log file.
- An administrator cannot alter audit records but can delete audit records as a whole.

TSF related to action in case of possible audit data loss is defined by **FAU_STG.3**. The TOE meets this SFR as follows:

- If the log files have already occupied more than 80% of the total audit storage in CF card (of the non-TOE hardware platform), or the oldest compressed files are deleted to save the latest log file, an event will be generated and sent to management server to notice the clients of the warning information.
- If the number of compressed log files generated in the system exceeded 90% of the maximum number of compressed files, an event will also be generated to notice management server the warning information.
- If the number of recorded compressed files reaches the maximum number that the security administrator has configured, or the storage of audit events reaches the configured storage size, another event will be generated to notice management server.

7.2 Cryptographic Support (FCS)

This section describes how the TOE meets each security functional requirement that belong to class FCS (defined in **[CC31R5P2]**) and that is listed in Section 6 of this ST.

7.2.1 Cryptographic Key Management

The TSF implements generation of cryptographic keys defined by **FCS_CKM.1** and cryptographic key establishment defined by **FCS_CKM.2**. In order to meet these SFRs, the TOE's DRBG is used to generate RSA keys with key sizes between 3072 to 4096 bits and ECC key pairs with 256, 384 and 521 curves used for ECDSA public key authentication. The generated keys are used for device authentication. The TOE implements Elliptic-curve based Diffie-Hellman algorithms for SSH and TLS key establishment.

The TOE acts as an SSH server to receive the communications for remote administration, and as a client for TLS communications to transmit log data to a remote Syslog server.

Scheme	SFR	Service
RSA Key generation	FCS_TLSC_EXT.1	Transmit generated audit data to an external IT entity
	FCS_SSHS_EXT.1	SSH remote administration
Elliptic-Curve Deffie-Hellman Key establishment	FCS_TLSC_EXT.1	Transmit generated audit data to an external IT entity
Elliptic-Curve Deffie-Hellman Key establishment	FCS_SSHS_EXT.1	SSH remote administration
	FCS_SSHC_EXT.1	SSH remote administration
ECC key generation and key establishment	FCS_SSHS_EXT.1	SSH remote administration
	FCS_SSHC_EXT.1	SSH remote administration

Table 6 Key generation and establishment methods and the corresponding services

Cryptographic key destruction TSF is defined by **FCS_CKM.4**. The table below describes how the TOE meets this SFR, by specifying how each type of cryptographic key is securely destroyed from each applicable storage media of the non-TOE hardware platform.

Name	Description of Key	Storage	Zeroization

Diffie-Hellman Shared Secret	This is the shared secret used as part of the Diffie-Hellman key exchange.	SDRAM (plaintext)	Overwritten with: zeros. Automatically after completion of DH exchange.
Diffie-Hellman private exponent	This is the private exponent used as part of the Diffie-Hellman key exchange.	SDRAM (plaintext)	Overwritten with: zeros. Automatically after completion of DH exchange.
SSH/TLS session key	The key is used for encrypting/decrypting the traffic in a secure connection.	SDRAM (plaintext)	Overwritten with: zeros. Automatically after session terminated.
SSH private host key	The key for authentication.	Internal flash (plaintext)	Overwritten with: a new value of the key. Overwritten by a command.
TLS private key	The key is used for signature and authentication.	Internal flash (plaintext)	Overwritten with: a new value of the key. Overwritten by a command.
RSA key pair	The key pair is used for digital signature and key establishment.	SDRAM (plaintext)	Overwritten with: zeros. Automatically after completion of use of the key.
RSA key pair	The key pair is used for digital signature and key establishment.	CF card (AES256 cipher)	Overwritten with: zeros Zeroized using “undo rsa key-pair” command.

Table 7 Key destruction methods

7.2.2 Cryptographic operations

TSF related to cryptographic operations for data encryption and decryption are supported by the TOE as follows.

AES encryption and decryption are provided by **FCS_COP.1/DataEncryption**. The TOE provides symmetric encryption and decryption capabilities using AES algorithm with key size 128 bits, 256 bits in GCM mode as specified in **[ISO 19772]**.

- AES128 GCM, AES256 GCM are supported by TLS.
- AES128 GCM, AES256 GCM are supported by SSH.

Cryptographic signature verification and generation operations are supported by **FCS_COP.1/SigGen**. The TOE provides cryptographic signature services using RSA with key sizes between 3072 and 4096 bits as specified in **[FIPS 186-4]** "Digital Signature Standard (DSS)":

- The RSA with key size 3072 is used for signature generation and verification of SSH.
- The RSA with key size of 3072 to 4096 is used for signature generation and verification of TLS.

Cryptographic operations for SHA algorithms are supported by **FCS_COP.1/Hash**. The TOE provides cryptographic hashing services using SHA-256, SHA-384 and SHA-512 as specified in **[FIPS Pub 180-3]**, it also meet the **[ISO/IEC 10118-3:2004]**. The association of the hash function with other TSF cryptographic functions is described in the table below:

Cryptographic Functions	Hash Function
HMAC-SHA-256	SHA-256
TLS Digital signature verification	SHA-256 SHA-384
SSH Digital signature verification	SHA-256 SHA-512
Hash_DRBG	SHA-256

Table 8 Association of the hash function with other crypto functions

Cryptographic operations for keyed-hash algorithms are supported by **FCS_COP.1/KeyedHash**. The TOE provides cryptographic keyed hash services using HMAC-SHA-256 according to **[RFC 2104]**: HMAC, it also complies with the **[ISO/IEC 9797-2:2011]**, Section 7 "MAC Algorithm 2". SSH and TLS performs keyed-hash message authentication in accordance with the specified cryptographic algorithm: HMAC-SHA-256. The table below defines the parameters used for HMAC cryptographic function:

HMAC functions	Key length (bits)	Hash function	Block size (bits)	Output MAC length (bits)
HMAC-SHA-256	256	SHA-256	512	256

Table 9 HMAC parameters

7.2.3 Random Number Generation

Generation of random numbers is provided by **FCS_RBG_EXT.1**. The TOE implements a deterministic random bit generator (DRBG) which is conformant to **[ISO/IEC 18031:2011]** using the DRBG mechanism Hash_DRBG as specified in **[SP800-90A]**, chap. 10.1.1. The entropy source is based on platform non-TOE hardware (one internal noise source). Random numbers from the internal noise source are only used for seeding the DRBG. The TOE sets a new seed using at least 256 bits entropy before generating random bits as cryptographic key. TSF uses Hash_DRBG to perform deterministic random bit generation. The identified hash functions (SHA-256) are allowed for Hash_DRBG.

7.2.4 SSH protocol cryptography

SSH protocol related cryptography for SSH client is provided by **FCS_SSHC_EXT.1**. The TOE meets this SFR as follows:

- The TOE implements the SSH protocol that comply with **[RFC 4251]**, **[RFC 4252]**, **[RFC 4253]**, **[RFC 4254]**, **[RFC 6668]**. All required algorithms required by **[RFC 4253]** are supported. For password-based authentication the guidance documentation will specify a minimum length for the password to ensure a minimum security strength of 100 bit (~16-20 characters, to be checked).
- Both public key and password authentication modes are supported by SSH client function. Users can use any or both of those modes to login external SSH server successfully. The supported public key algorithms for authentication include RSA with cryptographic key size of 3072-bit or greater. For password-based authentication the guidance documentation will specify a minimum length for the password to ensure a minimum security strength of 100 bit (~16-20 characters, to be checked).
- The TOE drops packets greater than 256 KB in an SSH transport connection Packets of size greater than 35000 bytes and smaller than 256 KB are not dropped because of that the TOE may support uncompressed big certificates.
- The SSH client supports the encryption algorithms of aes128-gcm and aes256-gcm. When SSH Client establishes a connection, it will send a list of encryption algorithms to SSH server. SSH Server will check each algorithm in the list one by one. If it finds one algorithm in the list that is also supported by it, this algorithm will be chosen as the encryption algorithm between client and server. If no algorithm in the list is supported by SSH server, the connection will be terminated. After the encryption algorithm is selected, Server and Client will create a random number and exchange. Client and Server will use own random number to create an encryption

key. Then SSH Client will use its own encryption key to encrypt packet, and use SSH Server's encryption key to decrypt packet.

- SSH client function supports the public key algorithm of rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384 and ecdsa-sha2-nistp521. Before SSHC and SSHS build a connection, they both need to configure a Local Key-pair what is used for authentication. In Huawei device, this local key-pair is used for SSH server and SSH client. When Client authenticates Server, first step is to consult public key algorithms. Client will send a list of public key algorithms to SSH server. SSH Server will check each algorithm in the list one by one. If it finds one algorithm in the list that is also supported by it, this algorithm will be chosen as the public key algorithm between client and server. If no algorithm in the list is supported by SSH server, the connection will be terminated. The SSH transport implementation doesn't support x509v3.
- SSH client supports the data integrity algorithms of hmac-sha2-256
- SSH client supports the following key exchange algorithm of ecdh-sha2-nistp256, ecdh-sha2-nistp384 and ecdh-sha2-nistp521.
- The SSH connection will be rekeyed after one hour of session time or one gigabyte of transmitted data using that key which ever goes first. The SSH allows either side to force another run of the key-exchange phase, changing the encryption and integrity keys for the session. The idea is to do this periodically, after one hour of session time or one gigabyte of transmitted data using that key which ever goes first.
- The SSH client will authenticate the identity of the SSH server using a local database associating each host name with its corresponding public key.

Cryptography for SSH protocol in server implementation is provided by **FCS_SSHS_EXT.1**. The TOE meets this SFR as follows:

- The TOE implements the SSH protocol that complies with **[RFC 4251]**, **[RFC 4252]**, **[RFC 4253]**, **[RFC 4254]**, **[RFC 6668]**. Both public key and password authentication modes are supported by SSH server function. The TOE implements the public key algorithms of rsa-sha2-256 and rsa-sha2-512. SSH users can be authenticated in eight modes: RSA, password, password-RSA, and All (any authentication mode of RSA or password is allowed with "ALL" mode). The SSH user that created by administrators shall configured one of mode. Then the external SSH client can login SSH server successfully via the configured SSH user and authentication mode.
- The TOE drops packets greater than 256 KB in an SSH transport connection. Packets of size greater than 35000 bytes and smaller than 256 KB are not dropped because of that the TOE may support uncompressed big certificates.
- SSH server function supports the encryption algorithms of aes128-gcm and aes256-gcm. When SSH Client establishes a connection, it will send a list of encryption algorithms to SSH server. SSH Server will check each algorithm in the list one by one. If it finds one algorithm in the list that is also supported by it, this algorithm will be chosen as the encryption algorithm between client and server. If no algorithm in the list is supported by SSH server, the connection will be terminated. After the encryption algorithm is selected, Server and Client will create a random number and exchange. Client and Server will use own random number

to create an encryption key. Then SSH server will use its own encryption key to encrypt packet, and use SSH client's encryption key to decrypt packet.

- SSH server function supports the public key algorithm of rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384 and ecdsa-sha2-nistp521. Before SSH Client and SSH Server build a connection, they both need to configure a Local Key-pair what is used for authentication. In the TOE, this local key-pair is used for SSH server and SSH client. When Client authenticates Server, first step is to consult public key algorithms. Client will send a list of public key algorithms to SSH server. SSH Server will check each algorithm in the list one by one. If it finds one algorithm in the list that is also supported by it, this algorithm will be chosen as the public key algorithm between client and server. If no algorithm in the list is supported by SSH server, the connection will be terminated. The SSH transport implementation doesn't support x509v3.
- SSH server function supports the data integrity algorithms of hmac-sha2-256.
- SSH server supports the following key exchange algorithm: ecdh-sha2-nistp256, ecdh-sha2-nistp384 and ecdh-sha2-nistp521.
- The SSH connection will be rekeyed after one hour of session time or one gigabyte of transmitted data using that key which ever goes first. The SSH allows either side to force another run of the key-exchange phase, changing the encryption and integrity keys for the session. The idea is to do this periodically, after one hour of session time or one gigabyte of transmitted data using that key which ever goes first.

7.2.5 Transport Layer Security Cryptography

TLS protocol-related cryptography is provided by **FCS_TLSC_EXT.1**. The TOE meets this SFR as follows:

- The TLS client supports the following ciphersuites:
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in **[RFC 5289]**.
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in **[RFC 5289]**.
- The TOE supports configuring reference identifier and matching this identifier with server certificate. The reference identifier is established by the user and by an application (a parameter of an API). Based on a singular reference identifier's source domain and application service type (e.g. HTTP, FTP), the client establishes all reference identifiers including a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.
- Only when the peer certificate is valid the TLS trusted channel can be established. If the peer certificate is invalid, the connection will be rejected.
- TLS doesn't support EC Extension in the Client Hello.

TLS communication Authentication is provided by **FCS_TLSC_EXT.2**. The TOE act as TLS client communicates to the audit (syslog) server with authentication using X.509v3 certificates.

7.3 Identification and Authentication

This section describes how the TOE meets each security functional requirement that belongs to class FIA (defined in **[CC31R5P2]**) and that is listed in Section 6 of this ST.

Authentication Failure Management TSF is provided by **FIA_AFL.1**. In order to meet this SFR, the TOE can be configured within 3 to 5 unsuccessful authentication attempts by Administrators. When the defined number of unsuccessful authentication attempts has been met, the TOE will prevent the offending remote Administrator from successfully authenticating until unlock is taken by a local Administrator or prevent the offending remote Administrator from successfully authenticating until an Administrator defined time period has elapsed.

Password Management TSF is provided by **FIA_PMG_EXT.1**. In order to meet this SFR, the TOE supports the local definition of users with corresponding passwords which are used for security administrators' authentication of local or remote administration connections. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters (not including spaces or question marks)". Minimum password length is configurable by the Authorized Administrator, and support passwords of 15 characters or greater. Password composition rules specifying the types and number of required characters that comprise the password are settable by the Authorized Administrator. Passwords have a maximum lifetime, configurable by the Authorized Administrator. The administrative passwords at local console or over protocols support the same set of special characters that listed in FIA_PGM_EXT.1.1.

User Identification and Authentication TSF is provided by **FIA_UIA_EXT.1**. The TOE meets this SFR as follows:

- The TOE requires all users to be successfully identified and authenticated before allowing execution of any TSF mediated action except display of the banner.
- The TOE supports user login over console or remote interface. Any login method need authentication before successfully logon.
- Local access is achieved by console port. The console interface supports user-based AAA authentication.
- Remote access is achieved by SSH. Users can initiate a SSH session to login to a remote interface by user-based AAA authentication. The TOE supports public-key of RSA or username/password for identity authentication. It also supports associated identity authentication of password and public-key. Users can also login with any of the identity authentication modes of password, and RSA when their login mode are configured to be 'ALL'.
- No services are available to users before authentication.

Password-based Authentication Mechanism is provided by **FIA_UAU_EXT.2**. In order to meet this SFR, the TOE can be configured to require local authentication (using console interface) or remote authentication (using SSH) as defined in the authentication policy for interactive (human) users. Administrators are authenticated against a local user database. If the interactive (human) users (Administrators) password is expired, the user is required to create a new password after correctly entering the expired password.

Protected Authentication Feedback is given by **FIA_UAU.7**. When a user inputs their password at the local console, the console will not display the input so that the user password is obscured. For remote session authentication, the TOE does not echo any characters as they are entered. The TOE does not provide any additional information to the user that would give any indication about the authentication data.

Validation of X.509 certificates is provided by **FIA_X509_EXT.1/Rev**. This requirement is met by the TOE as follows:

- The TOE supports to verify the certificate and the certificate path by the rules specified in **[RFC 5280]**, using algorithm RSA.
- The TOE supports to verify the revocation status by CRLs as specified in **[RFC 5280]**. Revocation status is verified using CRLs. TLS requires that certificates are used and this use requires that the extendedKeyUsage rules are verified.
- The TOE validates the certificate by steps as below:
    - Validate basic certificate fields and the extendedKeyUsage field.
    - Validate the revocation status using CRL as specified in **[RFC 5759]**.
    - Validate certificate path as specified in **[RFC 5280]**, do step 1 and 2 for every certificate in the certificate chain.
    - Validate the end of the certificate chain, it should be trusted root certificate.
- TLS requires that certificates are used and this use requires that the extendedKeyUsage rules are verified. The validation is expected to end in a trusted root CA certificate in a root store managed by the platform.
- The certificate path must end in a trusted root CA certificate otherwise it will be judged invalid.

Authentication with X.509 certificates functionality is provided by **FIA_X509_EXT.2**. Such functionality is used in TLS connections and code signing for system software updates. The TOE meets this SFR as follows:

- The certificate used by TLS authentication is sent by TLS server. The CRL should be loaded for certificate validation.
- The TOE will send a security log when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. TLS only supports RSA certificate.

The check of validity of the certificates takes place at authentication of TLS connection and verification of code signing for system software updates. When the certificate is valid, we can trust the peer identity and use the certificate to verify the integrity of the message. **FIA_X509_EXT.2** and **FTP_ITC.1.1** both use TLS for secure communications. If the TSF can't establish a connection to determine the validity of a certificate, then the certificate is not accepted by the TSF.

7.4 TOE Management Functions

This section describes how the TOE meets each security functional requirement that belong to class FMT (defined in **[CC31R5P2]**) and that is listed in Section 6 of this ST.

Management of security functions behavior related to manual updates is provided by **FMT_MOF.1/ManualUpdate**. In order to meet this SFR, The TSF restricts the ability to enable the functions to perform manual updates to Security Administrators. In addition, only administrators have the right to create or delete users in the TOE. While changing the local user privilege level, the configured new level of the local user cannot be higher than that of the login-in user. In this way no user except administrators can change another user to be at the privilege level of administrator, and only administrators have the ability to perform manual update. Therefore, the manual update is restricted to administrators. The TOE uses groups to organize users.

Management of security functions behavior related to transmission of audit data to external IT entities is provided **FMT_MOF.1/Functions**. The TOE meets this SFR by enforcing that:

- Only administrators have right to configure audit servers where audit records are exported to.
- Only administrators have the privilege to choose the trusted channel for external audit server and decide whether transmit the audit data to an external IT entity or not.
- Only administrators have the privilege to modify the behavior of TOE Security Functions (e.g. cryptographic algorithm, audit server).

Management of TOE services by administrators is provided by **FMT_MOF.1/Services**. The TSF enforces that only administrators have the ability to enable and disable the functions and services, the other users are disallowed to do it. The services that can be enabled and disabled by administrators are SSH, TLS, NTP and SYSLOG.

Management of security functions behavior related to TSF data is provided by **FMT_MTD.1/CoreData**. This SFR is met by the TOE as follows:

- Only administrators have privilege to manage the TSF data, the other users are disallowed to do it.
- The TOE provides the ability for authorized administrators to access TOE data, such as audit data, configuration data. Each of the predefined and administratively configured user has different right to access the TOE data.

Management functions related to cryptographic keys are defined by **FMT_MTD.1/CryptoKeys**. The TOE enforces that only administrators have the right to delete, generate, import the cryptographic keys or certificates., the other users are disallowed to do this.

Specification of management functions is provided by **FMT_SMF.1**. The TOE provides all the capabilities necessary to securely manage the TOE. The administrative user can connect to the TOE using the CLI to perform these functions via SSH encrypted session. The management functionality provided by the TOE includes the list of management functions described in **FMT_SMF.1** requirement, which can be exercised by administrator logged into the TOE through local or remote sessions. The ability to configure the available services before identification and authentication is not supported.

Restrictions on security roles are provided by **FMT_SMR.2**. A Security Administrator is able to administer the TOE through the local console or through a remote mechanism (SSH). An administrator can create, delete and modify the other users and endow them with a proper right according to the users' roles. The TOE uses groups to organize users. Different kinds of users are in different groups and every group has a specific level that identity its roles and scope of rights. Every user in one group has

the same scope of rights that the group owns. The TOE has 4 default user groups: "manage-ug, system-ug, monitor-ug, and visitor-ug.

7.5 Protection of the TSF

This section describes how the TOE meets each security functional requirement that belongs to class FPT (defined in **[CC31R5P2]**) and that is listed in Section 6 of this ST.

TSF data protection for reading symmetric cryptographic keys is provided by **FPT_SKP_EXT.1**. In order to meet this SFR, the TOE stores all pre-shared keys, symmetric keys, and private keys in the file system in Flash that can't be read, copy or extract by administrators; hence no interface access is available.

Protection of administrator passwords is given by **FPT_APW_EXT.1**. The administrator passwords are stored to configuration file in cryptographic form hashed with salt by SHA-256, including username passwords, authentication passwords, console and virtual terminal line access passwords. In this manner, the TOE ensures that plaintext user passwords will not be disclosed to anyone through normal interfaces including administrators.

TSF Self-testing is provided by **FPT_TST_EXT.1**. In order to meet this SFR, the TSF run a suite of self-tests during initial start-up to demonstrate the correct operation of the TSF, including software integrity verification by digital signature check and the correct operation of cryptographic functions. During initial power on start-up, software integrity is verified at first. If the digital signature-based check for the software fails, the start-up procedure is stopped. The cryptographic functions that are tested at start-up are RSA 2048 and SHA256.

Reliable Time-stamps functionality is defined by **FPT_STM_EXT.1**. The TOE provides timestamp service for internal use (e.g for associating a timestamp to a log). Reliable Time-stamps functionality is provided by FPT_STM_EXT.1, using the Network Time Protocol (NTP) as source of time. The TOE synchronizes its time source by means of an external time server using NTP protocol. The NTP communications carried out by the TOE support authentication using HMAC-SHA-256 algorithm.

Trusted updates-related TSF is provided by **FPT_TUD_EXT.1**. The TOE meets this SFR as follows:

- Only authenticated administrators have the ability to manually initiate an update to TOE firmware/software. During the updating procedure, digital signature as defined at **FCS_COP.1/SigGen** will be verified by the TOE at first.
- The administrators can query the currently executing version of the TOE firmware/software as well as the most recently installed version by a command. The currently executing patches and most recently installed patches can also be checked out.
- The validation of the firmware/software integrity is always performed before the process of replacing the current installed version of the TOE with a software update. All parts of the TOE software are archived together into a whole package and the single package is digitally signed. RSA as specified in **FCS_COP.1/SigGen** can be used for firmware/software digital signature verification mechanism, in order to authenticate it prior to installation, and that installation fails if the verification fails.

7.6 TOE access

This section describes how the TOE meets each security functional requirement that belongs to class FPT (defined in) and that is listed in Section 6 of this ST.

Locking of TSF-initiated sessions is provided by **FTA_SSL_EXT.1**. An administrator can configure maximum inactivity times for both local and remote administrative sessions. When a session is inactive (i.e., not session input) for the configured period of time the TOE will terminate the session, will flush the screen, and no further activity is allowed, requiring the administrator to log in (be successfully identified and authenticated) again to establish a new session. The allowable range is from 0 minute 0 second to 35791 minutes 59 seconds.

TSF-initiated termination of remote interactive sessions is given by **FTA_SSL.3** In order to meet this SFR, when the remote session is inactive (i.e., not session input) for the configured period of time the TOE will terminate the session.

User-initiated termination of sessions is provided by **FTA_SSL.4**. Administrators can use a command to proactively terminate their interactive session in the TOE.

Default access banners TSF is provided by **FTA_TAB.1**. The TOE meets this SFR as follows:

- To provide some prompts or alarms to users, Administrator can use the “header” command to configure a notice and consent informative message to be displayed when an administrator logs in the TOE. Administrator can directly specify the informative message, or they can specify the message information by using the contents of a file. The message displayed is the same for both local and remote users.
- When a terminal (remote or local) connection is activated and attempt to log in, the terminal displays the contents of the title that is set by using the header login command. After the successful login, the terminal displays the contents of the title that is configured by using the header shell command.
- The local Console port and the remote SSH are used for an administrator to communicate with the TOE.

7.7 Trusted path/channels

This section describes how the TOE meets each security functional requirement that belong to class FTP (defined in **[CC31R5P2]**) and that is listed in Section 6 of this ST.

FTP_ITC.1 Inter-TSF trusted channel is met by the TOE by protecting communications with audit server with a TLS-encrypted channel.

FTP_TRP.1 Trusted Path is met by the TOE by ensuring that all remote administrative communications take place over a secure encrypted SSH session. The remote users are able to initiate SSH communications with the TOE. When the administrators establish a session through SSH, an informative banner will be displayed.

TLS/SSH protects the data from disclosure by encryption defined by **FCS_COP.1/Hash** and ensure that the data has not been modified by MAC defined by **FCS_COP.1/KeyedHash**.

8 Acronyms

The following table shows the acronyms used in this document.

Acronym	Meaning
PP	Protection Profile
CC	Common Criteria
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFi	TSF Interface
OSP	Organisational Security Policies
EAL	Evaluation Assurance Level
ST	Security Target
IT	Information Technology
NTP	Network Time Protocol
NMS	Network Management Server
SMP	System Manage Plane
VRP	Versatile Routing Platform
SCP	Service Control Plane
GCP	General Control Plane
ARP	Address Resolution Protocol
STP	Spanning Tree Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
SSP	System Service Plane
DP	Data Plane
MAC	Media Access Control Address
AAA	Authentication Authorization Accounting
BGP	Border Gateway Protocol
RSA	Rivest, Shamir and Adleman
SHA	Secure Hash Algorithm
HMAC	Hash-based message authentication code
AES	Advanced Encryption Standard
DRBG	Deterministic random bit generator
DPF	Data Packet Forwarding

Acronym	Meaning
IC	Information Center
IS-IS	Intermediate System to Intermediate System
ISO	International Organization for Standardization
CLNP	Connectionless Network Protocol
IETF	Internet Engineering Task Force
OSI	Open System Interconnection
VLAN	Virtual Local Area Network
OSPF	Open Shortest Path First
RADIUS	Remote Authentication Dial-In User Service
SSH	Secure Shell
TLS	Transport Layer Security
CA	Certification Authority

Table 10 *Abbreviations*

9 Glossary of Terms

Term	Meaning
Augmentation	Addition of one or more requirement(s) to a package
Evaluation Assurance Level	Set of assurance requirements drawn from CC Part 3, representing a point on the CC predefined assurance scale, that form an assurance package
Operational Environment	Environment in which the TOE is operated
Protection Profile	Implementation-independent statement of security needs for a TOE type
Security Target	Implementation-dependent statement of security needs for a specific identified TOE
Target Of Evaluation	Set of software, firmware and/or hardware possibly accompanied by guidance

Table 11 *Glossary of terms*

10 Document References

The following table shows the acronyms used in this document.

Reference	Document
[CC31R5P1]	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, Part 1: Introduction and general model
[CC31R5P2]	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, Part 2: Security functional components
[CC31R5P3]	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, Part 3: Security assurance components
[CEM31R5P3]	Common Criteria Evaluation methodology, Version 3.1, Revision 5
[RFC-1195]	Use of OSI IS-IS for Routing in TCP/IP and Dual Environments (https://tools.ietf.org/html/rfc1195)
[FIPS 186-4]	National Institute of Standards and Technology, Digital Signature Standard (DSS), Federal Information Processing Standards Publication FIPS PUB 186-4, July 2013
[PKCS#1]	RSA Cryptography Specifications Version 2.1(RFC3447)
[PKCS#3]	A cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel.
[FIPS 198-1]	The Keyed-Hash Message Authentication Code (HMAC)--2008 July
[RFC 4251]	The Secure Shell (SSH) Protocol Architecture, January 2006
[RFC 4252]	The Secure Shell (SSH) Authentication Protocol, January 2006
[RFC 4253]	The Secure Shell (SSH) Transport Layer Protocol, January 2006
[RFC 4254]	The Secure Shell (SSH) Connection Protocol, January 2006
[RFC 6668]	SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol
[RFC 3268]	Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)
[RFC 5246]	The Transport Layer Security (TLS) Protocol Version 1.2
[RFC 8446]	The Transport Layer Security (TLS) Protocol Version 1.3
[RFC 6125]	Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)
[NIST SP 800-56A]	National Institute of Standards and Technology, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography, May 2013

Reference	Document
[NIST SP 800-56B]	National Institute of Standards and Technology, Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography August 2009
[ISO/IEC 18031:2011]	Information technology -- Security techniques -- Random bit generation
[ISO 18033-3]	Information technology — Security techniques — Encryption algorithms
[ISO/IEC 9796-2]	Information technology -- Security techniques -- Digital signature schemes giving message recovery
[ISO/IEC 9797-2:2011]	Information technology -- Security techniques -- Message Authentication Codes (MACs)
[ISO/IEC 10118-3:2004]	Information technology -- Security techniques -- Hash-functions
[ISO/IEC 14888-3]	Information technology -- Security techniques -- Digital signatures with appendix
[RFC 3526]	More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)
[ISO 19772]	Information technology — Security techniques — Authenticated encryption
[FIPS Pub 180-3]	Secure Hash Standard (SHS)
[RFC 2104]	HMAC: Keyed-Hashing for Message Authentication
[SP800-90A]	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
[RFC 5289]	TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)
[RFC 5280]	Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
[RFC 5759]	Suite B Certificate and Certificate Revocation List (CRL) Profile
[CPP_ND]	Collaborative Protection Profile for Network Devices, Version 2.2

Table 20 *List of document references*

11 Appendices

11.1 Crypto Disclaimer

The following cryptographic algorithms are used by NetEngine 8000 to enforce its security policy:

#	Purpose	Cryptographic Mechanism	Standard of Implementation	Key Size in Bits	Standard of Application	Comments
1	Key Generation	RSA and ECC schemes	-	3072-BIT OR GREATER	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3	FCS_CKM.1
2	Key Establishment	Elliptic-curve based key establishment schemes	Pair Wise Key Establishment Schemes Using Integer Factorization Cryptography	3072-bit or greater	NIST Special Publication 800 56B	FCS_CKM.2
3	Confidentiality	AES in GCM mode		128 bits or 256 bits	AES as specified in ISO 18033-3, GCM as specified in ISO 19772	FCS_COP.1/ DataEncryption
4	Authentication	RSA Signature	RSA: PKCS#1_V2.1, RSASSA-PKCS2v1_5	3072 bits	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5	FCS_COP.1/ SigGen
			Digital signature scheme 2 or	3072 bits	ISO/IEC 9796-2, Digital	FCS_COP.1/ SigGen

			Digital Signature scheme 3		signature scheme 2 or Digital Signature scheme 3	
	Integrity	HMAC-SHA-256	-	256 bits	ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”	FCS_COP.1/Hash
5	Cryptographic Primitive	SHA-256, SHA-384, SHA-512	-	256 bits, 384 bits and 512 bits	ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”	FCS_COP.1/Hash
6	Random Bit Generation	Hash_DRBG (any); DRG.2 acc. to SP800-90A	-	256 bits	SP800-90A ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”	FCS_RBG.1
7	Trusted Channel	SSH V2.0	RFC 6668 RFC 4251 RFC 4252 RFC 4253 RFC 4254	-	-	FTP_TRP.1/ Admin
		TLS1.2	RFC 6125 RFC 3268 RFC 5246	-	-	FTP_ITC.1

Table 21 *Crypto Disclaimer*