Klas VoyagerVM 4.0 running KlasOS Keel 5.4.3 Security Target

Version 1.1
16 May 2025

intertek acumen security

2400 Research Blvd
Suite 395
Rockville, MD 20850

Revision History

Version	Date	Changes
Version 0.1	July 29, 2024	Initial Release
Version 0.2	September 20, 2024	Updated TDs
Version 0.3	September 26, 2024	Addressed QA Feedback
Version 0.4	September 27, 2024	Updated Claims
Version 0.5	November 18, 2024	Addressed QA Feedback for Check-In
Version 0.6	December 5, 2024	Updated Claims for SFR's
Version 0.7	February 28, 2025	Added TD
Version 0.8	March 19, 2025	Updated Claims for SFR's
Version 0.9	April 25, 2025	TSS Updates
Version 1.0	May 1, 2025	Removed claims
Version 1.1	May 16, 2025	Addressed ECR Comments

Contents

1 Introduction

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE, and the functional and assurance requirements that the TOE meets.

1.1 Security Target and TOE Reference

This section provides the information needed to identify and control the TOE and the ST.

Table 1 – TOE/ST Identification

Category	Identifier
ST Title	Klas VoyagerVM 4.0 running KlasOS Keel 5.4.3 Security Target
ST Version	1.1
ST Date	May 16, 2025
ST Author	Acumen Security, LLC.
TOE Identifier	Klas VoyagerVM 4.0 running KlasOS Keel 5.4.3
TOE Version	5.4.3
TOE Developer	Klas
Key Words	Network Device

1.2 TOE Overview

The TOE is Klas VoyagerVM 4.0 running KlasOS Keel 5.4.3 (herein referred to as the TOE). It runs the 5.4.3 firmware combining both connectivity and local compute capabilities. This provides users with cloud connectivity when necessary and local processing power for analytics when there is no backhaul. Administration can be performed locally or over a trusted SSH channel.

1.3 TOE Description

This section provides an overview of the TOE architecture, including physical boundaries, security functions, and relevant TOE documentation and references. All TOE models below run the same Klas Keel 5.4.3 binary file. The TOE supports SSH functionality for both management and export of logging information.

Table 2 – TOE Model

TOE Model	Specifications
VoyagerVM 4.0 	Xeon D-1746TER Intel(R) Xeon(R) Ice Lake D-1746TER 10-Core CPU @ 2.00GHz with 128GB RAM or Xeon D-1712TR Intel® Xeon® D-1712TR Processor 4-Core CPU @ 2.00GHz with 128GB RAM Network Ports: ° 4 x 25 Gbps SFP28 interfaces

TOE Model	Specifications
	° 2 x 2.5 Gbps RJ45 Ethernet ports ° 1 x RJ45 Ethernet for management ° 1 x RJ45 Serial console port Network Ports: 1 x console, 2 x 10 GB SFP, 2 x 1GB ethernet Storage: · 2 x E1.S 9.5mm NVMe SED SSDs · 1 x VIK+ NVMe boot or write-cache device · 1 x 256 GB NVMe internal boot device (optional)

1.3.1 Physical Boundaries

The TOE boundary is the hardware appliance which is comprised of hardware and the KlasOS Keel software component. The TOE hardware model is provided in Table 2 – TOE Model.

The TOE also supports secure connectivity with several other IT environment devices, including the ones identified in the following table.

The TOE implements SSHv2 to protect the remote management interface for administrators.

Table 3 – TOE Physical Boundary Components

Component	Required	Purpose/Description
Management Workstation	Yes	A management workstation that is directly connected to the TOE's console port may be used by the TOE administrator to support TOE administration. Note: Either a remote or local management workstation, or both can be used.
Remote (Management) Workstation / Remote SSH CLI	Yes	This includes any IT Environment Management workstation with a SSH client installed that is used by the TOE administrator to support TOE administration through SSH protected channel. Any SSH client that supports SSHv2 may be used. Note: Either a remote or local management workstation, or both can be used.
Syslog Server	Yes	The syslog audit server is used for remote storage of audit records that have been generated by and transmitted from the TOE. An SSH tunnel is established by the TOE and logs are transmitted using this encrypted method.
NTP Server	No	The NTP server is used to send reliable timestamps to the TOE using NTPv3 and SHA1 as the message digest algorithm.

1.3.2 Security Functions Provided by the TOE

The TOE provides the security functions required by the Collaborative Protection Profile for Network Devices, hereafter referred to as NDcPP v3.0e or NDcPP. In addition, the TOE provides the following security functions for Network Devices. The TOE implements the following security requirements:

- Security Audit (FAU)
- Cryptographic Support (FCS)
- User Data Protection (FDP)
- Identification and Authentication (FIA)
- Security Management (FMT)
- Protection of the TSF (FPT)
- TOE Access (FTA)
- Trusted Path/Channels (FTP)

1.3.2.1 Security Audit

The TOE generates audit events for all start-up and shutdown functions as well as all auditable events specified in Table 12 – Security Functional Requirements and Auditable Events. Audit events are also generated for management actions specified in FAU_GEN.1. The TOE stores audit records locally and will export them to an external syslog server using SSHv2 as a tunnel. Each audit record contains the date and time of the event, type of event, subject identity, and other relevant data for the event. Only a security administrator can enable logging to a syslog server.

1.3.2.2 Cryptographic Support

The cryptographic used in the TOE are presented in the following table.

Table 4 –TOE Cryptography Implementation

Cryptographic Methods	Usage
FCS_CKM.1 Cryptographic Key Generation	• Cryptographic key generation conforming to FIPS PUB 186-4 “Digital Signature Standard (DSS)”, Appendix B.3. • RSA Key sizes supported are 2048, 3072 and 4096 bits. • Cryptographic key generation conforming to FIPS PUB 186-4 “Digital Signature Standard (DSS)”, Appendix B.4. • Elliptic NIST curves supported are: P-256, P-384, and P-521.
FCS_CKM.2 Cryptographic Key Establishment	• Elliptical curve-based establishment conforming to NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”
FCS_CKM.4 Cryptographic Key Destruction	• Refer to Table 17 – Cryptographic Key Destruction for Key Zeroization details.
FCS_COP.1/DataEncryption Cryptographic Operations (AES Data Encryption/Decryption)	• AES encryption and decryption conforming to CBC and CTR as specified in ISO 10116.

Cryptographic Methods	Usage
	• AES key size supported is 128 and 256 bits • AES mode supported is CBC and CTR .
FCS_COP.1/Hash Cryptographic Operations (Hash Algorithm)	• Cryptographic hashing services conforming to ISO/IEC 10118-3:2004. • Hashing algorithms supported are: SHA-1, SHA-256, SHA-384, and SHA-512. • Message digest sizes supported are: 160, 256, 384, and 512 bits.
FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)	• Keyed-hash message authentication conforming to ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm2”. • Keyed hash algorithm supported are: HMAC-SHA1, HMAC-SHA-256, HMAC-SHA384, and HMAC-SHA-512 • Key sizes supported are: 160, 256, 384 and 512 bits. • Message digest sizes supported are: 160, 256, 384, and 512 bits.
FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)	• RSA digital signature algorithm conforming to 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. • RSA key sizes supported are: 2048, 3072 and 4096 bits. • Elliptical curve digital signature algorithm conforming to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing NIST curves ISO/IEC 14888-3, Section 6.4. • Elliptical curve key sizes supported are 256 and 384 bits.
FCS_NTP_EXT.1 NTP Protocol	• The TOE supports NTP v3 and adheres to RFC 1305. • Authentication is performed using SHA-1 as the message digest algorithm.
FCS_RBG_EXT.1 Random Bit Generation	• Random number generation conforming to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions” • The TOE leverages CTR_DRBG(AES) • CTR_DRBG seeded with a minimum of 256 bits of entropy.

Cryptographic Methods	Usage
FCS_SSHS_EXT.1 SSH Server Protocol	• The TOE supports SSH v2 protocol compliant to the following RFCs:4251, 4252, 4253, 4254, 5656, and 6668. • The TOE supports password-based and public-key-based authentication. • SSH public-key authentication uses ssh-rsa, ecdsa-sha2-nistp256 and ecdsa-sha2-nistp384. • SSH transport uses the following encryption algorithms: aes128-cbc, and aes256-cbc. • Packets greater than 262155 bytes in an SSH transport connection are dropped. • SSH transport uses the following data integrity MAC algorithms: hmac-sha2-256 and hmac-sha2-512 • Key exchange algorithms supported are: ecdh-sha2-nistp256 and ecdh-sha2-nistp384. • The TOE ensures that during 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.
FCS_SSHC_EXT.1 SSH Client Protocol	• The TOE supports SSH v2 protocol compliant to the following RFCs:4251, 4252, 4253, 4254, 5656, and 6668. • The TOE supports public-key-based authentication. • SSH public-key authentication uses ssh-rsa, ecdsa-sha2-nistp256 and ecdsa-sha2-nistp384. • SSH transport uses the following encryption algorithms: aes128-cbc, aes 128-ctr, aes128-cbc and aes256-ctr. • Packets greater than 262155 bytes in an SSH transport connection are dropped. • SSH transport uses the following data integrity MAC algorithms: hmac-sha2-256 and hmac-sha2-512 • Key exchange algorithms supported are: ecdh-sha2-nistp256 and ecdh-sha2-nistp384. • The TOE ensures that during 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

1.3.2.3 Identification and Authentication

All users must be authenticated by the TOE prior to carrying out any administrative actions. The TOE supports password-based and public-key based authentication. An administrator can set a minimum password length on the TOE which must be at least 15 characters. This is true of users accessing the TOE via the local console, or through protected paths using the remote CLI via SSH. Users can authenticate to the TOE using a username and password. In addition, when authenticating by the remote CLI, users can instead use SSH public-key authentication. Passwords can consist of upper-case letters, lower-case letters, numbers, and a set of selected special characters. Password information is never revealed during the authentication process including during login failures. Before a user authenticates the device, a customizable warning banner is configured to be displayed.

1.3.2.4 Security Management

The TOE supports local and remote management of its security functions including:

- Local console CLI administration
- Remote CLI administration via SSHv2
- Configurable banner displayable at login
- Timeouts to terminate administrative sessions after a set period of inactivity
- Timed user lockout after multiple failed authentication attempts
- Configurable authentication failure parameters
- Re-enabling locked accounts
- Configurable cryptographic parameters

The administrative user can perform all the above security-related management functions.

1.3.2.5 Protection of the TSF

The TOE protects all passwords, pre-shared keys, symmetric keys, and private keys from unauthorized disclosure. Passwords are stored as SHA 512 hashes. The TOE executes self-tests during initial start-up to ensure correct operation and enforcement of its security functions. The TOE internally maintains the date and time. An administrator can install software updates to the TOE after they are verified using a digital signature mechanism.

1.3.2.6 TOE Access

The TOE displays a customizable banner before any administrative session can be established with it. The TOE will terminate local or remote interactive sessions after a specified period of session inactivity configured by an administrator. An administrator can terminate their own interactive local or remote sessions.

The local and remote CLI interfaces display the default security banner prior to authentication that is also configurable. The TOE can terminate local CLI and remote CLI sessions after a specified time-period of inactivity. Administrative users have the capability to terminate their own sessions.

1.3.2.7 Trusted Path/Channels

The TOE supports SSH for secure communications with authorized IT entities such as syslog servers. The TOE supports SSHv2 (remote CLI) for secure remote administration.

1.3.3 TOE Documentation

The following documents are essential to understanding and controlling the TOE in the evaluated configuration:

- Klas VoyagerVM 4.0 running KlasOS Keel 5.4.3 Operational User Guidance, Version 0.7, May 2025

1.4 Product Functionality not Included in the Scope of the Evaluation

The following product functionality is not included in the CC evaluation:

Table 5 – Excluded Functionality

Components	Exclusion Rationale
SNMP	Remote management is performed using SSH
Spanning-Tree	Spanning-Tree is not used in the evaluated configuration
TACACS+	TACACS+ is not used for authentication on the TOE
Port Security	Port Security is not used in the evaluated configuration
RADIUS	RADIUS is not used in the evaluated configuration
SD-WAN	SD-WAN using the DTLS protocol is not enabled in the evaluated configuration
Firewall Functionality	The Firewall functionality is disabled in the evaluated configuration

2 Conformance Claims

This section identifies the TOE conformance claims, conformance rationale, and relevant Technical Decisions (TDs).

2.1 CC Conformance Claims

The TOE is conformant to the following:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017 (Extended)
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017 (Conformant)

2.2 Protection Profile Conformance

This ST claims exact conformance to the following:

- collaborative Protection Profile for Network Devices, Version 3.0e, 06 December 2023 [CPP_ND_V3.0E]
- Functional Package for SSH, Version 1.0, 14 May 2021 [PKG_SSH_v1.0]

2.3 Conformance Rationale

This ST provides exact conformance to the items listed in the previous section. The security problem definition, security objectives, and security requirements in this ST are all taken from the Protection Profile (PP) and PKG_SSH_v1.0, performing only the operations defined there.

2.3.1 Technical Decisions

All NIAP TDs issued to date and applicable to NDcPP v3.0e have been considered. Table 6 identifies all applicable TDs.

Table 6 – Relevant Technical Decisions

Technical Decision	Applicable (Y/N)	Protection Profile	Exclusion Rationale (if applicable)
TD0682: Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	Y	PKG_SSH_v1.0	
TD0695: Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.	Y	PKG_SSH_v1.0	
TD0732: FCS_SSHS_EXT.1.3 Test 2 Update	Y	PKG_SSH_v1.0	
TD0777: Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	Y	PKG_SSH_v1.0	
TD0836: Redundant Requirements in FPT_TST_EXT.1	Y	CPP_ND_V3.0E	

Technical Decision	Applicable (Y/N)	Protection Profile	Exclusion Rationale (if applicable)
TD0868: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	N	CPP_ND_V3.0E	IPSec is not claimed by the TOE
TD0879: Correction of Chapter Headings in CPP_ND_V3.0E	Y	CPP_ND_V3.0E	
TD0880: Removal of Duplicate Selection in FMT_SMF.1.1	Y	CPP_ND_V3.0E	
TD0886: Clarification to FAU_STG_EXT.1 Test 6	Y	CPP_ND_V3.0E	
TD0899: Correction of Renegotiation Test for TLS 1.2	N	CPP_ND_V3.0E	TLS is not claimed by the TOE
TD0900: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	Y	CPP_ND_V3.0E	

3 Security Problem Definition

The security problem definition has been taken directly from the claimed PP and any relevant EPs/Modules/Packages specified in Section 2.2 and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any Organizational Security Policies (OSPs) that the TOE is expected to enforce.

3.1 Threats

The threats included in Table 7 are drawn directly from the PP and any EPs/Modules/Packages specified in Section 2.2.

Table 7 – Threats

ID	Threat
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	Threat agents may attempt to gain Administrator access to the Network Device 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 Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents 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 traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical 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 network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents 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 network traffic is exposed and there could be a loss of

ID	Threat
	confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Nonvalidated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker. Threat agents may also be able to take advantage of weak administrative passwords to gain privileged access to the device.
T.SECURITY_FUNCTIONALITY_FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

3.2 Assumptions

The assumptions included in Table 8 are drawn directly from PP and any relevant EPs/Modules/Packages.

Table 8 – Assumptions

ID	Assumption
A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or cPP_ND_v3.0e, 06-Dec-2023 41 interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does 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. For vNDs, this assumption applies to the physical platform on which the VM runs.
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).
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 Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes 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 device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.
A.REGULAR_UPDATES	The Network Device 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 (private key) used to access the Network Device are protected by the platform on which they reside

ID	Assumption
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.

3.3 Organizational Security Policies

The OSPs included in Table 9 are drawn directly from the PP and any relevant EPs/Modules/Packages.

Table 9 – OSPs

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

4 Security Objectives

The security objectives have been taken directly from the claimed PP and any relevant EPs/Modules/Packages and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the Operational Environment

Security objectives for the operational environment assist the TOE in correctly providing its security functionality. These objectives, which are found in the table below, track with the assumptions about the TOE operational environment.

Table 10 – Security Objectives for the Operational Environment

ID	Objectives for the Operational Environment
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.
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	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
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 (private key) 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. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

5 Security Requirements

This section identifies the Security Functional Requirements (SFRs) for the TOE. The SFRs included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revisions 5, April 2017, and all international interpretations.

Table 11 – SFRs

Requirement	Description
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_NTP_EXT.1	NTP Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSH_EXT.1	SSH Protocol
FCS_SSHC_EXT.1	SSH Client Protocol
FCS_SSHS_EXT.1	SSH Server Protocol
FIA_AFL.1	Authentication Failure Handling
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU.7	Protected Authentication Feedback
FMT_MOF.1/Functions	Management of Security Functions Behaviour
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour
FMT_MOF.1/Services	Management of Security Functions Behaviour
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on security roles
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)
FPT_STM_EXT.1	Reliable Time Stamps
FPT_TST_EXT.1	TSF Testing
FPT_TUD_EXT.1	Trusted Update
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination

Requirement	Description
FTA_SSL.4	User-initiated Termination
FTA_TAB.1	Default TOE Access Banner
FTP_ITC.1	Inter-TSF Trusted Channel
FTP_TRP.1/Admin	Trusted Path

5.1 Conventions

The CC allows the following types of operations to be performed on the functional requirements: assignments, selections, refinements, and iterations. The following font conventions are used within this document to identify operations defined by CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration identifier after a slash, e.g., /SigGen.
- Where operation where completed in the PP and relevant Eps/Modules/Packages, the formatting used has been retained except for text within brackets. Operations completed by the ST author follow the formatting described in the 1st 3 bullets, otherwise, the text is in plaintext.
- Extended SFRs are identified by the addition of "EXT" after the requirement name.

5.2 Security Functional Requirements

This section includes the security functional requirements for this ST.

5.2.1 Security Audit (FAU)

5.2.1.1 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 shut-down of the audit functions;
b. All auditable events for the not specified level of audit; and
    - *All administrative actions comprising:*
    - *Administrative login and logout (name of Administrator 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 Administrator account shall be logged];
c. *Specifically defined auditable events listed in **Table 12***.

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 cPP/ST, information specified in column three of Table 12.

Table 12 – Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
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_NTP_EXT.1	• Configuration of a new time server • Removal of configured time server	• Identity if new/removed time server
FCS_RBG_EXT.1	None	None
FCS_SSH_EXT.1	[Failure to establish SSH connection]	[Reason for failure and Non-TOE endpoint of attempted connection (IP Address)]
FCS_SSH_EXT.1	[Establishment of SSH connection]	[Non-TOE endpoint of connection (IP Address)]
FCS_SSH_EXT.1	[Termination of SSH connection session]	[Non-TOE endpoint of connection (IP Address)]
FCS_SSH_EXT.1	[Dropping of packet(s) outside defined size limits]	[Packet size]
FCS_SSHC_EXT.1	None	None
FCS_SSHS_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 identification and authentication mechanism	Origin of the attempt (e.g., IP address)
FIA_UAU.7	None	None
FMT_MOF.1/Functions	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None
FMT_MOF.1/Services	None	None
FMT_MTD.1/CoreData	None	None
FMT_MTD.1/CryptoKeys	None	None

Requirement	Auditable Events	Additional Audit Record Contents
FMT_SMF.1	All management activities of TSF data.	None
FMT_SMR.2	None	None
FPT_SKP_EXT.1	None	None
FPT_TST_EXT.1	None	None
FPT_APW_EXT.1	None	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).
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None
FTA_SSL.3	The termination of a remote session by the session locking mechanism	None
FTA_SSL.4	The termination of an interactive session	None
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session locking mechanism	None
FTA_TAB.1	None	None
FTP_ITC.1	• Initiation of the trusted channel • Termination of the trusted channel • Failure of the trusted channel functions	• None • None • Reason for failure
FTP_TRP.1/Admin	• Initiation of the trusted path • Termination of the trusted path. • Failure of the trusted path functions.	• None • None • Reason for failure

5.2.1.2 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.

5.2.1.3 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 [

- The TOE shall consist of a single standalone component that stores audit data locally

].

FAU_STG_EXT.1.3

The TSF shall maintain a [log file] of audit records in the event that an interruption of communication with the remote audit server occurs.

FAU_STG_EXT.1.4

The TSF shall be able to store [non-persistent] audit records locally with a minimum storage size of [*10 MBs*].

FAU_STG_EXT.1.5

The TSF shall [overwrite previous audit records according to the following rule*: [oldest log file is overwritten*]] when the local storage space for audit data is full.

FAU_STG_EXT.1.6

The TSF shall provide the following mechanisms for administrative access to locally stored audit records [ability to view locally].

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1
The TSF shall generate **asymmetric** cryptographic key in accordance with a specified cryptographic key generation algorithm: [

- RSA schemes using cryptographic key sizes of [*2048, 3072 and 4096 bits*] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3;
- ECC schemes using “NIST curves” [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4;

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].~~

5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1
The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;

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

5.2.2.3 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

- F*or 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-pass] overwrite consisting of [zeroes];
    - instructs a part of the TSF to destroy the abstraction that represents the key]

that meets the following: *No Standard*

5.2.2.4 FCS_COP.1/DataEncryption Cryptographic Operations (AES Data Encryption/Decryption)

FCS_COP.1.1/DataEncryption
The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* [CBC, CTR] *mode* and cryptographic key sizes [128 bits, 256 bits] that meet the following: *AES as specified in ISO 18033-3,* [CBC as specified in ISO 10116, CTR as specified in ISO 10116].

5.2.2.5 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

FCS_COP.1.1/SigGen
The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm,
- Elliptic Curve Digital Signature Algorithm

]
and cryptographic key sizes [

- For RSA: modulus 2048 bits or greater,
- For ECDSA: 256 bits or greater

]

that meets the following: [

- 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,
- For ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” [P-256, P-384]; ISO/IEC 14888-3, Section 6.4,

].

5.2.2.6 FCS_COP.1/Hash Cryptographic Operations (Hash Algorithm)

FCS_COP.1.1/Hash
The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512] ~~and cryptographic key sizes~~ [~~*assignment: cryptographic key sizes*~~] and **message digest sizes** [160, 256, 384, 512] **bits** that meet the following: *ISO/IEC 10118-3:2004*.

5.2.2.7 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

FCS_COP.1.1/KeyedHash
The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes [*160, 256, 384, and 512 bits*] **and message digest sizes** [160, 256, 384, 512] **bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.2.2.8 FCS_NTP_EXT.1 NTP Protocol

FCS_NTP_EXT.1.1
The TSF shall use only the following NTP version(s) [NTP v3 (RFC 1305)].

FCS_NTP_EXT.1.2
The TSF shall update its system time using [

- Authentication using [SHA1] as the message digest algorithm(s);]

]

FCS_NTP_EXT.1.3
The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4
The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

5.2.2.9 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 [CTR_DRBG (AES)].

FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[one]* platform-based noise source] with a minimum of [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.

5.2.2.10 FCS_SSH_EXT.1.1 SSH Protocol (PKG_SSH_v1.0)

FCS_SSH_EXT.1.1
The TOE shall implement SSH acting as a [client, server] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [5656, 6668] and [no other standard].

FCS_SSH_EXT.1.2

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

- "password" (RFC 4252),
- "publickey" (RFC 4252): [
    - ssh-rsa (RFC 4253),
    - ecdsa-sha2-nistp256 (RFC 5656),
    - ecdsa-sha2-nistp384 (RFC 5656),
    ]

] and no other methods.

FCS_SSH_EXT.1.3
The TSF shall ensure that, as described in RFC 4253, packets greater than [*262155*] in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4
The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [

- aes128-cbc (RFC 4253),
- aes256-cbc (RFC 4253),
- aes128-ctr (RFC 4344),
- aes256-ctr (RFC 4344)

] and no other mechanisms.

FCS_SSH_EXT.1.5
The TSF shall protect data in transit from modification, deletion, and insertion using: [

- hmac-sha2-256 (RFC 6668),
- hmac-sha2-512 (RFC 6668),

] and no other mechanisms.

FCS_SSH_EXT.1.6
The TSF shall establish a shared secret with its peer using: [

- ecdh-sha2-nistp256 (RFC 5656),
- ecdh-sha2-nistp384 (RFC 5656),

] and no other mechanisms.

FCS_SSH_EXT.1.7
The TSF shall use SSH KDF as defined in [

- RFC 4253 (Section 7.2),
- RFC 5656 (Section 4)

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

FCS_SSH_EXT.1.8
The TSF shall ensure that [

- a rekey of the session keys,

] occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

5.2.2.11 FCS_SSHC_EXT.1 SSH Protocol – Client (PKG_SSH_v1.0)

FCS_SSHC_EXT.1.1
The TSF shall authenticate its peer (SSH server) using: [

- Using a local database by associating each host name with a public key corresponding to the following list: [
  - ssh-rsa (RFC 4253),
  - ecdsa-sha2-nistp256 (RFC 5656),
  - ecdsa-sha2-nistp384 (RFC 5656),

] as described in RFC 4251 section 4.1.

5.2.2.12 FCS_SSHS_EXT.1 SSH Protocol – Server (PKG_SSH_v1.0)

FCS_SSHS_EXT.1.1
The TSF shall authenticate itself to its peer (SSH Client) using: [

- ssh-rsa (RFC 4253),
- ecdsa-sha2-nistp256 (RFC 5656),
- ecdsa-sha2-nistp384 (RFC 5656),

].

5.2.3 Identification and Authentication (FIA)

5.2.3.1 FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1
The TSF shall detect when an Administrator configurable positive integer within [*1-255*] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

FIA_AFL.1.2
When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending Administrator from successfully establishing a remote session using any authentication method that involves a password until [*manual account unlocking*] is taken by an Administrator].

5.2.3.2 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: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [*“~”, “`”, “.”, “/”, “:”, “;”, “_”, “+”, “-“, “=”, “{“, “}”, “[“, “]”, “|”, “<”, “>”*]]

b) Minimum password length shall be configurable to between [*15*] and [*128*] characters.

5.2.3.3 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:

- Display the warning banner in accordance with FTA_TAB.1;
- [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.

FIA_UIA_EXT.1.3
The TSF shall provide the following remote authentication mechanisms [SSH password, SSH public key] and [no other mechanism]. The TSF shall provide the following local authentication mechanisms [password-based].

Application Note: This SFR has been updated as per TD0900.

FIA_UIA_EXT.1.4
The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

5.2.3.4 FIA_UAU.7.1 Protected Authentication Feedback

FIA_UAU.7.1
The TSF shall provide only *obscured feedback* to the **administrative** user while the authentication is in progress **at the local console**.

5.2.4 Security Management (FMT)

5.2.4.1 FMT_MOF.1/Functions Management of Security Functions Behaviour

FMT_MOF.1.1/Functions
The TSF shall restrict the ability to [modify the behaviour of] the functions [transmission of audit data to an external IT entity] to *Security Administrators*.

5.2.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behavior

FMT_MOF.1.1/ManualUpdate
The TSF shall restrict the ability to enable the function *to perform manual updates to Security Administrators.*

5.2.4.3 FMT_MOF.1/Services Management of Security Functions Behaviour

FMT_MOF.1.1/Services
The TSF shall restrict the ability to **start and stop** ~~the functions~~ **services** to *Security Administrators*.

5.2.4.4 FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1.1/CoreData
The TSF shall restrict the ability to manage the *TSF data to Security Administrators.*

5.2.4.5 FMT_MTD.1/CryptoKeys Management of TSF Data

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

5.2.4.6 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1
The TSF shall be capable of performing the following management functions:

- *Ability to administer the TOE remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the remote session inactivity time before session termination;*
- *Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;*
- [
  - Ability to start and stop services;
  - 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 re-enable an Administrator account;
  - Ability to set the time which is used for time-stamps;
  - Ability to configure NTP;
  - Ability to administer the TOE locally;
  - Ability to configure the local session inactivity time before session termination or locking;
  - Ability to configure the authentication failure parameters for FIA_AFL.1;
  - Ability to manage the trusted public keys database;

].

Application Note: This SFR has been addressed by TD0880.

5.2.4.7 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1
The TSF shall maintain the roles:

- *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

- *The Security Administrator role shall be able to administer the TOE remotely;*

are satisfied.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 FTP_APW_EXT.1 Protection of Administrator Passwords

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

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

5.2.5.2 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.

5.2.5.3 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 [allow the Security Administrator to set the time, synchronise time with an NTP server].

5.2.5.4 FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests:

- During initial start-up (on power on) to verify the integrity of the TOE firmware and software;
- Prior to providing any cryptographic service and [at no other time] to verify correct operation of cryptographic implementation necessary to fulfil the TSF;
- [no other] self-tests [*are run*].

to demonstrate the correct operation of the TSF.

Application Note: This SFR has been updated as per TD0836.

FPT_TST_EXT.1.2
The TSF shall respond to [[*integrity check fail, FIPS self-test failures, Entropy Health Test Failure*]] by [[*The device will revert to the default image (Integrity check failure) ,the boot will continue with crypto functions disabled(FIPS self-test failures, Entropy Health Test Failure)*]].

5.2.5.5 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1
The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and [no other TOE firmware/software version].

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

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

5.2.6 TOE Access (FTA)

5.2.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1
The TSF Shall, for local interactive sessions, [

- terminate the session]

after a Security Administrator-specified time period of inactivity

5.2.6.2 FTA_SSL.3 TSF-initiated Termination

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

5.2.6.3 FTA_SSL.4 User-initiated Termination

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

5.2.6.4 FTA_TAB.1 Default TOE Access Banners

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

5.2.7 Trusted Path/Channels (FTP)

5.2.7.1 FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1
The TSF shall **be capable of using** [SSH] **to** provide a trusted communication channel between itself and ~~another trusted IT product~~ **authorized IT entities supporting the following capabilities: audit server,** [no other capabilities] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from ~~modification or~~ disclosure **and detection of modification of the channel data**.

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

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

5.2.7.2 FTP_TRP.1/Admin Trusted Path

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

FTP_TRP.1.2/Admin
The TSF shall permit remote **Administrators** ~~users~~ to initiate communication via the trusted path.

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

5.3 TOE SFR Dependencies Rationale for SFRs

The PP and any relevant EPs/Modules/Packages contain(s) all the requirements claimed in this ST. As such, the dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the PP and any relevant EPs/Modules/Packages, which is/are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in Table 13.

Table 13 – Security Assurance Requirements

Assurance Class	Assurance Components	Component Description
Security Target	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic functionality specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative Procedures
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_IND.1	Independent testing – conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability survey

5.5 Assurance Measures

The TOE satisfied the identified assurance requirements. This section identifies the Assurance Measures applied by Klas to satisfy the assurance requirements. The following table lists the details.

Table 14 – TOE Security Assurance Measures

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1 ALC_CMS.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ATE_IND.1	Vendor will provide the TOE for testing.
AVA_VAN.1	Vendor will provide the TOE for testing. Vendor will provide a document identifying the list of software and hardware components.

6 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

Table 15 – TOE Summary Specification SFR Description

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td>FAU_GEN.1</td><td>The TOE generates a comprehensive set of audit logs that identify specific TOE operation whenever an auditable event occurs. Auditable events are specified in Table 12 – Security Functional Requirements and Auditable Events plus the additional events in FAU_GEN.1.1(a) and FAU_GEN.1.1(b). Each of the events specified in the audit records is in enough detail to identify the user with which the event is associated, when the event occurred, where the event occurred, the outcome of the event and the type of event that occurred. Administrative tasks of generating, importing and deleting cryptographic keys identify the keys unique name. SSH public keys are identified by the username in the logs on the TOE.</td></tr>
<tr><td>FAU_GEN.2</td><td>The TOE ensures that each auditable event is associated with the identity of the user that triggered the event.</td></tr>
<tr><td>FAU_STG_EXT.1</td><td>Audit events are stored locally and are also sent to an external audit server in real-time. SSHv2 is used to provide a trusted communication channel with the Syslog Server. Data stored locally is kept in an audit log file. Each log file is rotated at approximately 10MB in size but due to the lag between the appending to the log and the rotation of the log, the size may grow larger than this. Each log will never grow larger than 20MB in size. The previous log is overwritten by the new log if audit files are full. The TOE user or Security Administrator is not able to modify the audit records. There are no conditions where a user or administrator can perform a deletion of audit records. The TOE is standalone and audit data is stored locally. All log records are non-persistent on the TOE. Only administrators can view the log files on the TOE.</td></tr>
<tr><td>FCS_CKM.1</td><td>The TOE supports several cryptographic key generation schemes which include RSA 2048, 3072 and 4096-bit, ECC P-256, ECC P-384, ECC P-521. These are detailed in FCS_CKM.1.
<table>
<tr><th>Key Generation</th><th>SFR</th><th>Usage</th></tr>
<tr><td>Elliptic curve</td><td>FCS_SSH_EXT.1</td><td>SSH Server for administration and SSH Client to syslog server</td></tr>
</table>
</td></tr>
<tr><td>FCS_CKM.2</td><td>In agreement with the key generation schemes the RSA-based and Elliptic curve-based are supported as detailed in FCS_CKM.2.
<table>
<tr><th>Key Establishment Scheme</th><th>SFR</th><th>Usage</th></tr>
<tr><td>Elliptic curve</td><td>FCS_SSHS_EXT.1
FCS_SSHC_EXT.1</td><td>SSH Server for administration and SSH Client to syslog server</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Requirement</th><th>TSS Description</th></tr>
<tr><td></td><td></td></tr>
<tr><td>FCS_CKM.4</td><td>RSA keys and ECDSA keys are stored in plaintext in non-volatile memory. They are deleted with read-verify when any of the designated cryptographic key zeroization commands identified in AGD are executed by the administrator. The TOE stores keys in plaintext only. The TOE satisfies all requirements for destruction of keys and CSPs as specified in Table 17 – Cryptographic Key Destruction

There are no configurations or circumstances that may not conform to the key destruction requirements.</td></tr>
<tr><td>FCS_COP.1/DataEncryption</td><td>The TOE supports AES encryption and decryption conforming to CBC and CTR as specified in ISO 18033-3, ISO 19772 and ISO 10116. The AES key sizes supported are 128 and 256 bits. AES is implemented in the following protocol: SSH. Please refer to Table 16 – CAVP Algorithm Certificate References for NIST CAVP certificate numbers for AES.</td></tr>
<tr><td>FCS_COP.1/Hash</td><td>SSH and NTP support cryptographic hashing using the following Hash Sizes and Digest Sizes:
<table><tr><th>Protocol</th><th>Hash Size</th><th>Digest Size</th></tr><tr><td>SSH</td><td>SHA-1, SHA-256, SHA-384 and SHA-512</td><td>160, 256, 384, and 512 bits</td></tr><tr><td>NTP</td><td>SHA-1</td><td>160 bits</td></tr></table>Hashing is also leveraged for Signature Verification for Trusted Updates, Password storage, Self-Integrity tests and HMAC_DRBG tests on the TOE.</td></tr>
<tr><td>FCS_COP.1/KeyedHash</td><td>SSH and NTP support cryptographic hashing using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, or HMAC-SHA-512 with message digest sizes of 160, 256, 384, and 512 bits. The key length, hash function used, block size, and output MAC lengths are identified in the table below.
<table><tr><th>Algorithm</th><th>Block Size</th><th>Key Size</th><th>Digest Size</th></tr><tr><td>HMAC-SHA-1</td><td>512 bits</td><td>160 bits</td><td>160 bits</td></tr><tr><td>HMAC-SHA-256</td><td>512 bits</td><td>256 bits</td><td>256 bits</td></tr><tr><td>HMAC-SHA-384</td><td>1024 bits</td><td>384 bits</td><td>384 bits</td></tr><tr><td>HMAC-SHA-512</td><td>1024 bits</td><td>512 bits</td><td>512 bits</td></tr></table>Hashing is also leveraged for Signature Verification for Trusted Updates, Password storage, Self-Integrity tests and HMAC_DRBG tests on the TOE.</td></tr>
<tr><td>FCS_COP.1/SigGen</td><td>The TOE provides cryptographic signature generation and verification services in accordance with the following cryptographic algorithms:

RSA Key sizes of 2048, 3072 and 4096 bits

ECDSA Key sizes of 256 and 384 bits</td></tr>
</table>

Requirement	TSS Description
FCS_NTP_EXT.1	The TOE uses NTP v3 (RFC 1305) and uses SHA1 for authenticating time stamps received. The NTP sources are defined by the Security Administrator. Up to three sources can be configured. NTPv3 is implemented on the TOE using Chrony version 3.4. Time updates from a broadcast and multicast address are not supported by the TOE. The TOE uses the combination of the configured IP address alongside the SHA1 hash algorithm to properly authenticate the NTP server. If either of these values are not correct, the NTP server will not synchronize with the TOE until the proper authentication configuration is set.
FCS_RBG_EXT.1	The TOE implements a DRBG in accordance with ISO/IEC 18031:2011 using a CTR DRBG with AES. The noise source is the Intel RDSEED CPU instruction and is seeded with a minimum of 256 bits of entropy. Refer to the ancillary document: Entropy Assessment Report.
FCS_SSH_EXT.1 FCS_SSHC_EXT.1 FCS_SSHS_EXT.1	The TOE implements SSH client and server capabilities that comply with RFC(s) 4251, 4252, 4253, 4254, 5656, and 6668. SSH public key authentication is supported with the following key pairs: ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384. Packet sizes up to 262155 bytes are accepted and packets exceeding this size are dropped and this event is logged by the TOE. The TOE supports encryption algorithms AES-128-CBC, AES-256-CBC, AES-128-CTR and AES-256-CTR to ensure confidentiality of the session. When the TOE is acting as an SSH Client (Syslog), passwords are not supported. When the TOE is acting as an SSH Server, passwords are supported. The TOE supports the following hostkey algorithms: ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384. An IP address is associated with each host-key public key when a key is uploaded to the TOE. The TOE identifies the public key that is presented by the server and verifies if it matches one of the stored keys within the client. If the presented key does not match, authentication is prevented. The TOE supports the following data integrity algorithms: hmac-sha2-256, and hmac-sha2-512. The TOE supports the following key exchange algorithms: ecdh-sha2-nistp256, and ecdh-sha2-nistp384. The TOE supports the following RSA key sizes: 2048, 3072, and 4096. The TOE is capable of rekeying and verifies the following thresholds: No longer than one hour No more than 1 GB of transmitted data The TOE continuously checks both conditions. When either of the conditions are met, the TOE will initiate a rekey. When a user logs into the TOE, they are authenticated via a username and password or public key. Password-based authentication is not required if a public key is being used. If public key authentication is not available, all users must log in using a password specified for that user account. The password is determined by the user and must conform to the requirements set out in FIA_PMG_EXT.1. When verifying a user's password, the one way hash is computed and the result is checked against the value stored for the username in the /etc/passwd file. Only

Requirement	TSS Description
	certain programs on the TOE can access the /etc/passwd file, for example sshd. Users/admins do not have access. All values are hashed within the /etc/passwd directory so there is no access to plaintext password values. The TOE identifies the public key that is presented by the client and verifies if it matches one of the stored keys within the server. If the presented key does not match, authentication is prevented.
FIA_AFL.1	An administrator can configure the maximum number of failed attempts using the CLI interface. The configurable range is between 1 and 255 attempts. When a user account has sequentially failed authentication for the configured number of times, the account will be locked, until a local administrator manually unlocks the account. If the lockout attempts are set to, for example, 5 attempts, then the user will be locked out after the 5th consecutive failed login attempt. This means that the 6th and subsequent attempts will fail to gain access to the TOE even if the credential being offered is correct. All failed attempts and lockouts are tracked by the TOE audit logs. The TOE will always allow a user to authenticate using the local console port, even if the user account is locked. This behavior is not configurable.
FIA_PMG_EXT.1	The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of upper- and lower-case letters, numbers, and at least one of the special characters that include the following: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”,“~”, “,”, “.”, “/”, “:”, “;”, “_”, “+”, “-“, “=”, “{“, “}”, “[“, “]”, “|”, “<”, “>”. The minimum password length can be configured by the Administrator and can range from 15 to 128 characters.
FIA_UIA_EXT.1	The TOE requires all users to be successfully identified and authenticated as an administrator before allowing any TSF mediated actions to be performed. Access to the TOE is facilitated through directly connecting to the TOE through serial console or remotely connecting to the TOE through SSHv2. Every user that authenticates is first logged in with non-administrative privileges with limited viewing functionalities. The user may then authenticate as an administrator with additional credentials to gain access to modifying functionalities. Only one user can be escalated to the Administrator Role. Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality. No TOE administrative access is permitted until an administrator is successfully identified and authenticated. For remote administration, the TOE supports public key authentication and password-based authentication. If the user uses public key-based authentication when prompted and it is successful, then the user is granted access to the TOE. If the user uses password-based authentication and they provide valid username and password, then user is granted access to the TOE. If the user enters invalid user credentials, they will not be granted access. The TOE does not provide a reason for failure in case of a login failure. The TOE displays a banner in accordance with FTA_TAB.1 before a user can log into the device. No other services are available before a user logs in with valid credentials.

Requirement	TSS Description
FIA_UAU.7	For all authentication at the local CLI the TOE does not display any authentication data. The TOE does not display even obscured data such as asterisks during authentication attempts.
FMT_MOF.1/Functions	The Security administrator can configure a SSH tunnel for secure transmission of audit data to a syslog server. The IP address of the system log and the port to be used can be configured.
FMT_MOF.1/ManualUpdate	The TOE restricts the ability to perform software updates to Security Administrators. There can be only one Security Administrator at all times.
FMT_MOF.1/Services	The TOE may be managed via the CLI (console and remote SSH). The specific services the administrator can start and stop and how they do it are shown below: SSH Administration Enabling and disabling remote SSH access can be done via the CLI SSH syslog connections Enabling and disabling SSH syslog can be done via the CLI Local console and remote administration provide the same functionalities based on the level of authentication.
FMT_MTD.1/CoreData	The TOE restricts the ability to manage the TOE to Security Administrators. Administrative users are required to login before being provided with access to any administrative functions. Non-security administrators are not allowed to modify any TOE functions and TSF data. No interface is available to an unauthenticated user except the login prompt. Any commands used to modify TOE functions are not made available to non-administrative users and its attempt to use them will result in an invalid action error. These requirements also apply to the TOE trust store and all the certificates inside.
FMT_MTD.1/CryptoKeys	The security administrator can generate, import, and delete cryptographic keys. The specific keys they can manage are listed below: EC Session Keys and Diffie Hellman Group 14 Session keys used for FCS_SSHS_EXT.1 and FCS_SSHC_EXT.1. SSH hostkeys can be stored in the TOE's database along with the identity of the host the TOE is connecting to.
FMT_SMF.1	The available management functions are listed below and these can be accessed via the SSH command line interface remotely. The local interface can be accessed via a serial port and is identified with "tty" in the audit record. • Ability to administer the TOE remotely • Ability to configure the access banner • Ability to configure the remote session inactivity time before session termination • Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates • Ability to start and stop services • Ability to modify the behaviour of the transmission of audit data to an external IT entity

Requirement	TSS Description
	• Ability to manage the cryptographic keys • Ability to configure the cryptographic functionality • Ability to re-enable an Administrator account • Ability to set the time which is used for time-stamps • Ability to configure NTP • Ability to administer the TOE locally • Ability to configure the local session inactivity time before session termination or locking • Ability to configure the authentication failure parameters for FIA_AFL.1 • Ability to manage the trusted public keys database
FMT_SMR.2	The TOE supports a security administrator role. The security administrator can administer the TOE locally or remotely.
FPT_APW_EXT.1	The TOE stores all password authentication data in a secure directory that is not readily accessible to administrators. Passwords are obscured from the user for local and remote CLI interfaces. The passwords are stored as SHA-512 hash and are not in plaintext.
FPT_SKP_EXT.1	The TOE stores all private, symmetric and asymmetric keys in secure storage and is not accessible through an interface to administrators. Passwords are obscured from the user from local and remote CLI interfaces. The TOE stores all password authentication data in a secure directory that is not accessible to administrators. Private keys may be destroyed or replaced but cannot be read. Refer to Table 17 – Cryptographic Key Destruction for more key storage details.
FPT_STM_EXT.1	The TOE provides reliable time stamps. The clock function is reliant on the system clock provided by the underlying hardware. This clock is kept accurate and reliable using NTP, which is optional, or manual setting by the administrator. The following security functions make use of the system time: • Audit events • Session inactivity • SSH Rekey The time can be manually updated by a Security Administrator.
FPT_TST_EXT.1	The TOE runs a suite of self-tests during initial start-up to verify its correct operation. If any of the tests fail, the TOE will enter an error state. When the TOE is in an error state, all crypto functions are disabled and the only way to access the TOE is through a local console. The user guidance also instructs the administrator to reach out to the vendor in the case the self-tests fail. The TOE executes the following self-tests when powered on: Integrity check – The TOE performs an integrity check of the installed firmware by comparing the 4096-bit digital signature of the complete firmware image during bootup before any configuration is loaded and interfaces are enabled. If the signature verification fails, all crypto functionality is disabled and a message will be sent to the system log.

Requirement	TSS Description
	FIPS module self-tests in accordance with the OpenSSL 3.0.8 FIPS 140-2 Policy – The TOE performs FIPS self-tests to test the integrity of the crypto libraries when the cryptographic module is first initialized during boot-up. This includes KAT (Known-Answer Test) and PCT (Pair-wise Consistency Test) on all supported algorithms. If any cryptographic self-test fails, the TOE will complete the boot process with all cryptographic functions disabled. The entropy noise source health tests are performed during bootup as part of the self-tests. They also are run continuously during system runtime. Entropy health testing – If the entropy noise source health testing fails, the TOE immediately reboots and logs an audit message at the local console. If all self-tests pass, a success log will be generated on the local console window for the administrator to verify. The TOE will then continue to boot in a normal state.
FPT_TUD_EXT.1	Before posting a new image for customer download, Klas creates a SHA256 hash of the image and then cryptographically digitally signs the hash using an RSA private key. This signed hash is then appended to the end of the firmware image. The public key is burned into the image already. The key that is burned into the image is used to validate the cryptographic signature of the update file. The Security Administrator can query the software version running on the TOE using the 'show version' command and is able to perform manual software updates. When software updates are made available by Klas, the Security Administrator can download and initiate installation of the update. The TOE will verify that the signed hash on the new image is valid before booting with the new image. If the image fails the signature check, then the image is deleted from the device and no upgrade occurs. Delayed activation for trusted updates is not supported on the TOE.
FTA_SSL.3 FTA_SSL_EXT.1	A Security Administrator can configure maximum inactivity times for administrative sessions through the TOE local CLI and remote SSH interfaces. The configuration of inactivity periods are applied on a per-interface basis and can be applied to both local, and remote sessions in the same manner. When the interface has been idle for more than the configured period of time, the session will be terminated and will require reauthentication to establish a new session.
FTA_SSL.4	A Security Administrator is able to exit out of both local, and remote administrative sessions. For both local and remote sessions, the session is terminated by entering the "exit" command.
FTA_TAB.1	Access to the TOE is facilitated through by directly connecting to the TOE through serial console or remotely connecting to the TOE through SSHv2. Security Administrators can define a customized login banner that will be displayed at the local CLI and remote CLI (SSH). This banner will be displayed prior to allowing Security Administrators access.
FTP_ITC.1	A remote audit server must be configured and the communication between the TOE and the audit server is protected by SSHv2 tunnel using public-key based authentication. SSH provides assured identification of

Requirement	TSS Description
	the non-TSF endpoint by validating the public key received from the endpoint. The TOE acts as a client in the syslog connection. All cryptographic information that pertains to syslog connections can be found under FCS_SSHC_EXT.1. The TOE assures identification of all endpoints by recording the details of the IP address to the log file. For example, a log will be generated when the TOE connects to an SSH syslog server indicating the identity of the server being connected to.
FTP_TRP.1/Admin	Remote administration is performed using a CLI interface that is protected by SSHv2 using AES encryption. All requirements that secure this connection can be found in FCS_SSHS_EXT.1.

6.1 CAVP Algorithm Certificate Details

Each of these cryptographic algorithms have been validated as identified in the table below.

Table 16 – CAVP Algorithm Certificate References

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_CKM.1	RSA schemes using cryptographic key sizes of 2048, 3072 or 4096-bit that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3	KlasOS Keel 5.4	RSA KeyGen	A4573
	ECC schemes using "NIST curves" [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	KlasOS Keel 5.4	ECDSA KeyGen	A4573
FCS_CKM.2	Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"	KlasOS Keel 5.4	KAS-ECC-SSC	A4573

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
FCS_COP.1/ DataEncryption	AES used in [CBC, CTR] mode and cryptographic key sizes [128 bits, 256 bits] as specified in ISO 10116.	KlasOS Keel 5.4	AES-CBC 128 bits, 256 bits AES-CTR 128 bits, 256 bits	A4573
FCS_COP.1/ Hash	[SHA-1, SHA-256, SHA-384, SHA-512] and message digest sizes [160, 256, 384, 512] bits that meet the following: ISO/IEC 10118-3:2004.	KlasOS Keel 5.4	SHA-1 SHA2-256 SHA2-384 SHA2-512	A4573
FCS_COP.1/ KeyedHash	[HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes [160, 256, 384, and 512 bits] and message digest sizes [160, 256, 384, 512] bits and message digest sizes that meet the following: ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.	KlasOS Keel 5.4	HMAC-SHA-1 HMAC-SHA2-256 HMAC-SHA2-384 HMAC-SHA2-512	A4573
FCS_COP.1/ SigGen	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	KlasOS Keel 5.4	RSA-SigGen RSA-SigVer 2048, 3072 and 4096	A4573
	For ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” [P-256, P-384]; ISO/IEC 14888-3, Section 6.4	KlasOS Keel 5.4	ECDSA-SigGen ECDSA-SigVer P-256, P-384	A4573
FCS_RBG_EXT.1	CTR_DRBG (AES-256) Random bit generation	KlasOS Keel 5.4	Counter DRBG	A4573

SFR	Algorithm in ST	Implementation name	CAVP Alg.	CAVP Cert #
	services in accordance with ISO/IEC 18031:2011		(AES 256)	

6.2 Cryptographic Key Destruction

The table below describes the key zeroization provided by the TOE and as referenced in FCS_CKM.4.

Table 17 – Cryptographic Key Destruction

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
EC Session Keys	Ephermeral Session Key for SSH Session Establishment	Epheremal; stored in RAM (Volatile storage)	Overwritten with zeroes at end of session.
RSA Key	Signature Generation, Signature Verification for SSH public key authentication.	Restricted key partition in plaintext not available to users or administrators (Non-Volatile storage)	Deleted with read-verify when any of the designated cryptographic key zeroization commands identified in AGD are executed by the administrator. Key zeroization will instruct a part of the TOE to destroy the abstraction that represents the key. Generating a new key will overwrite and erase any existing keys and replacing the old keys with a new key value.
		While in use, RSA keys are held in RAM (Volatile storage)	Overwritten with zeroes when the key is no longer in use (after performing a cryptographic operation) or overwritten with a new value of the key when a new key value.
ECDSA Key	Signature Generation. Signature Verification for SSH public key authentication and verification of trusted updates.	Restricted key partition in plaintext not available to users or administrators (Non-Volatile storage)	Deleted with read-verify when any of the designated cryptographic key zeroization commands identified in AGD are executed by the administrator. Key zeroization will instruct a part of the TOE to destroy the

Keys/CSPs	Purpose	Storage Location	Method of Zeroization
			abstraction that represents the key. Generating a new key will overwrite and erase any existing keys and replacing the old keys with a new key value.
		While in use, ECDSA keys are held in RAM (Volatile storage)	Overwritten with zeroes when the key is no longer in use (after performing a cryptographic operation)
HMAC Key	Keyed Hashing for SSH	While in use, keys for HMAC keyed hashing are held in RAM (Volatile storage)	Overwritten with zeroes when the key is no longer in use (after performing a cryptographic operation).
AES Session Keys	SSH Data Encryption	Ephemeral; stored in RAM (Volatile storage)	Overwritten with zeroes at end of session

7 Acronym Table

Acronyms should be included as an Appendix in each document.

Table 18 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CAVP	Cryptographic Algorithm Validation Program
CC	Common Criteria
CLI	Command Line Interface
EP	Extended Package
GUI	Graphical User Interface
HMAC	Hash-Based Message Authentication Code
IP	Internet Protocol
HTTPS	Hypertext Transfer Protocol Secure
ISO	Internation Organization for Standardization
KAT	Known-Answer Test
MB	Megabyte
NDcPP	Network Device Collaborative Protection Profile
NIAP	Nation Information Assurance Partnership
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
PCT	Pair-wise Consistency Test
PP	Protection Profile
RBG	Random Bit Generator
RSA	Rivest, Shamir & Adleman
SFR	Security Functional Requirement
SSH	Secure Shell
SHA	Secure Hash
ST	Security Target
TOE	Target of Evaluation
TLS	Transport Layer Security
TSS	TOE Summary Specification
TSF	TOE Security Functionality
WS	Work Station