NORTHROP GRUMMAN

Information Systems

Cyber Solutions Division

M5 Network Security

SCS-100 & SCS-200

Security Target

Version 1.5.3

28/07/2015

Document information

Document identification

Document ID	SCS_NDPP_ST
Document title	SCS-100 & SCS-200 NDPP
Release authority	Andrew Coward
Product version	SCS_NDPP_ST

Document history

Version	Date	Description
0.1	11/10/2012	Initial draft for vendor. Missing some cryptographic and packet filter details.
0.2	29/11/2012	Final draft
1.0	06/12/2012	Version 1.0 released
1.1	25/10/2013	Updated in response to EOR_ASE 1.0
1.2	22/01/2014	Updated in response to EOR_ASE 2.0
1.3	06/06/2014	Updated for correctness and completeness
1.4	18/09/2014	Removed firewall extension
1.5	22/10/2014	Updated in response to EOR_ASE 3.0
1.5.1	30/10/2014	Additions per ALC PP requirements
1.5.2	15/05/2015	Updated with FIPS validation certificates.
1.5.3	28/07/2015	Updated based on ACA comments.

Table of Contents

1 Security Target Introduction

1.1 Background

Reliable computer networks require a significant amount of underlying infrastructure, which can grow proportionally as the size and complexity increases. In order to build, configure, operate and maintain such complex networks, highly skilled specialist personnel are required. Highly secure networks, such as those used in government and enterprise scenarios are also subject to additional requirements and restrictions, which lead to additional installation and administrative complexity. This situation does not lend itself towards mobility or unpredictable environments. Infrastructure relocation can be costly and time consuming, and require specialist personnel.

The M5 Secure Communications System (SCS) is a next-generation secure communications solution for military, government and large corporations. The SCS has been designed to allow mobile teams to securely exchange data in a cost-effective manner, with minimal administrative and configuration overheads.

The SCS products have been specifically designed for operation by non-IT specialists. The SCS-100 and SCS-200 devices feature intuitive graphical user interfaces that allow one-touch set up and simple configuration. To further assist the user, the system utilises network traffic, SNMP, and event log data to detect and repair faults and provide clear advice on system or device status. The system also features remote administration capabilities through the SCS-NMS (not included in the scope of this evaluation).

Each SCS device can manage multiple simultaneous external connections and select the optimal communications path based on performance and/or monetary considerations. Further, the SCS can sense and automatically establish connections with other SCS devices on the same network, further enhancing the communications paths at a system level.

1.2 ST reference

ST Title	Security Target for the SCS-100 & SCS-200
ST Version/Date	1.5.3 (27/07/2015)
TOE Reference	SCS-100 (Firmware 23) & SCS-200 RevC (Firmware 35d) Software Build: 5.3.6
Protection Profile	Protection Profile for Network Devices Version 1.1
PP Extended Package(s)	None
CC Identification	Common Criteria for Information Technology (IT) Security Evaluation, Version 3.1 (REV 3) July 2009, incorporating: • Part One – Introduction and General Model (Ref. [1]), • Part Two – Security Functional Components (Ref. [2]), and • Part Three – Security Assurance Components (Ref. [3]).

1.3 Document organization

This document is organized into the following major sections:

a) Section 1 provides the introductory material for the ST as well as the Target of Evaluation (TOE) overview.

b) Section 2 provides the conformance claims for the evaluation.

c) Section 3 provides the definition of the security problem addressed by the TOE.

d) Section 4 defines the security objectives for the TOE and the environment.

e) Section 5 contains the security functional requirements derived from the Protection Profile for Network Devices and the Common Criteria, Part 2.

f) Section 6 contains the security assurance requirements derived from the Protection Profile for Network Devices and the Common Criteria, Part 3.

g) Section 7 provides the TOE summary specification that demonstrates how the TOE implements the claimed security functions.

References

[1] *Common Criteria for Information Technology Security Evaluation Part 1: Introduction and general model, Version 3.1, Revision 3 Final, CCMB-2009-07-001,* 2009.

[2] *Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 3 Final, CCMB-2009-07-002,* 2009.

[3] *Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 3 Final, CCMB-2009-07-003,* 2009.

[4] National Institute of Standards and Technology, *Federal Information Processing Standards Publication 140-2: Security Requirements for Cryptographic Modules,* 2001 (change notices 2002).

[5] D. J. &. M. S. Elaine Barker, *NIST Special Publication 800-56A: Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised),* National Institute of Standards and Technology, 2007.

[6] L. C. A. R. &. M. S. Elaine Barker, *NIST Special Publication 800-56B: Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography,* National Institute of Standards and Technology, 2009.

[7] Information Assurance Directorate, *Network Device Protection Profile (NDPP) Extended Package: Stateful Traffic Filter Firewall,* 2011.

[8] Information Assurance Directorate, *Protection Profile for Network Devices, Version 1.1,* 2012.

[9] N. C. Team, “IPTABLES Man Page,” 03 July 2008. [Online]. Available: http://ipset.netfilter.org/iptables.man.html.

[10] F. Marie, “Netfilter Extensions HOWTO, Revision 3822,” 03 April 2005. [Online]. Available: http://netfilter.org/documentation/HOWTO/netfilter-extensions-HOWTO.txt.

[11] R. Russel, “Linux 2.4 Packet Filtering HOWTO, Revision 492,” 24 January 2002. [Online].

Available: http://netfilter.org/documentation/HOWTO//networking-concepts-HOWTO.txt.

1.4 Defined terms

The following table defines all subjects, objects, operations, security attributes, external entities and other key terms that are used within the statements of security functional and assurance requirements.

Term/Acronym	Definition
Black Network	A public network or network which is not considered controlled or secure.
Blue Network	A high security private network to which the TOE provides access.
Critical Security Parameter	Security-related information (e.g., secret and private cryptographic keys, and authentication data such as passwords and PINs) whose disclosure or modification can compromise the security of a cryptographic module (as defined in FIPS 140-2 (Ref. [4])).
Data Exfiltration	Data is removed without proper authorization to remove it.
Egress Traffic	Traffic from threat agents that exist inside a protected network.
Ingress Traffic	Traffic from threat agents that exist outside a protected network.
Protected Network	An attached network for which rules are defined to control access.
Network Device	An infrastructure device that can be connected to a network.
Network Management System (NMS)	A central administration and management component of the M5 Secure Communication System (SCS). (Also called SCS-NMS)
Red Network	A very high security private network to which the TOE provides access. The highest security network which the SCS can connect to.
Secure Communication System (SCS)	The M5 Secure Communication System. A secure communication solution for military, government and large corporations, designed to allow mobile teams to exchange data most efficiently, securely, and cost-effectively.
SCS-EI	The SCS Enterprise Interface. This is an IP gateway which connects enterprise network infrastructure to the SCS mesh network via public networks.

1.5 ST overview

1.5.1 TOE overview

The TOE is a series of dedicated embedded hardware devices, which provides secure IP-based communication services over any internet connection. It is designed for use in varied environments and mobile and sometimes unpredictable situations, which may include environments such as hotel rooms, offices or battlefields.

When deployed, the TOE forms a secure DMVPN mesh network with other SCS devices, either via direct connection or over an internet connection. It provides a combination of routing functionalities for an “all-in-one” mobile secure network solution. It also provides three network domains - these networks are intended for use with increasingly sensitive data, and each is afforded increased data transport protection.

The TOE encompasses two models: the SCS-100 and SCS-200. Each of these devices implement the core security functionality that meets the SFRs listed in section 5 of this document, but may also offer differing extended functionality.

The SCS-100 is designed with size and portability in mind. It provides secure communication services for a single user.

The SCS-200 provides secure communication services for one to four users. The SCS-200 also provides the ability to make use of external cryptographic modules. The SCS-200 includes two touch screen interfaces, allowing two separate configurations simultaneously.

1.5.2 TOE usage and major security features

The TOE utilises industry standard and specialised cryptography to protect outgoing communications. It supports three segregated networks of increasing security requirements simultaneously:

- The Black network is protected by an IPsec tunnel. This network is designed for use with unsecured or public networks, such as a remote internet gateway. All communications, including those form the other networks, are routed through this tunnel.

- The Blue network runs within the Black network, and is segregated from the other networks and protected by an additional IPsec tunnel. This network is designed for communications with a remote network with higher security requirements.

- The Red network also runs within the Black network, and is segregated from the other networks and protected by specialised cryptography. This network is designed for communications with a remote network with higher security requirements than the Blue network.

Each of these external connections terminates either at another instance of the TOE or at a remote SCS Enterprise Interface (SCS-EI). The SCS-EI is a network gateway located within a trusted network infrastructure. The SCS-EI and associated remote infrastructure lies outside of the scope of this evaluation.

In addition to secure communication, the TOE also provides security audit, information flow control, identification & authentication of administrators, secure management and self-test functionality.

The TOE provides a local touchscreen interface display basic network and status information and allows users without a high level of technical knowledge to setup and enable network connectivity. A local SSH interface is provided for administrative management. The SSH interface may also be accessed remotely within the IPsec tunnel.

The TOE also provides an interface that allows management and configuration of the TOE from an SCS-Network Management System (SCS-NMS) located in a central location (typically an SCS-EI). The NMS communicates with the TOE over a dedicated TLS connection within the Black IPsec tunnel. The NMS can be used to access administrative functions, and automatically receives audit log events from the TOE. All management functions accessible to the NMS are also accessible to a local administrator. The NMS is not within in the scope of the NDPP Common Criteria evaluation.

1.5.3 TOE type

The TOE is an all-in-one network device, providing routing, network segregation and mesh network connectivity. The TOE is categorised as a ***network device***, as described in section 1.1 of the Protection Profile for Network Devices (NDPP), version 1.1. It can be categorised as ***network and network-related devices and systems*** in accordance with the categories identified on the Common Criteria Portal that lists all certified products.

2 Conformance Claim

The ST and TOE are strictly conformant to the Protection Profile for Network Devices, version 1.1. In addition, the TOE also claims conformance to the FPT_TST.1 security functional requirement.

The specific conformance claims are made for this ST and TOE are:

a) **Protection Profile for Network Devices, version 1.1;**

b) **CC Part 2 extended; and**

c) **CC Part 3 conformant.**

The set of threats, assumptions and organisational security policies defined in this security target are a superset of the threats, assumptions and organisational security policies defined in the protection profiles to which strict conformance is claimed.

3 Security Problem Definition

3.1 Overview

This section describes the nature of the security problem that the TOE is designed to address. The security problem is described through:

a) a set of ***threats*** that the TOE must mitigate,

b) specific ***assumptions*** about the security aspects of the environment (both IT related and non-IT related elements) in which the TOE will operate, and

c) relevant ***organisational security policies*** that specify rules or guidelines that must be followed by the TOE and/or the operational environment.

3.2 Threats

In the context of this ST, the TOE has the following threat agents:

a) Individuals that have not been granted access to the application who attempt to gain access to information or functions provided by the TOE. This threat agent is considered an ***unauthorised individual***.

b) Individuals that are registered and have been explicitly granted access to the application who may attempt to access information or functions that they are not permitted to access. This threat agent is considered an ***authorised user***.

Identifier	Threat statement
T.ADMIN_ERROR	An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNDETECTED_ACTIONS	Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data.

Identifier	Threat statement
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.USER_DATA_REUSE	User data may be inadvertently sent to a destination not intended by the original sender
T.INTRA_NETWORK_DISCLOSURE	Sensitive information on a protected network might be disclosed to an untrusted entity while in transit between the TOE and a trusted network device.
T.STORED_DATA_MODIFICATION	A malicious party may attempt to modify data stored within the TOE or the TOE firmware.

3.3 Organisational security policies

In the context of this ST, the following organisational security policies (OSPs) are used to provide the basis for security objectives that are most often desired by acquirers and users of the TOE.

Identifier	OSP statement
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.

3.4 Assumptions

The following assumptions provide the foundation for security objectives for the operational environment for the TOE.

Identifier	Assumption statement
A.NO_GENERAL_PURPOSE	It is assumed that 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.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

4 Security Objectives

4.1 Overview

The security objectives are a concise statement of the intended response to the security problem defined in Section 3. There are security objectives for the TOE to address and additional objectives that provide specific direction for the intended environment in which the TOE is to operate.

4.2 Security objectives for the TOE

Identifier	Objective statements
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities.
O.VERIFIABLE_UPDATES	The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators.
O.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.SESSION_LOCK	The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked.
O.TSF_SELF_TEST	The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly.
O.DATA_MODIFICATION_DETECTION	The TOE will detect unauthorised modification to sensitive stored data or to the TOE firmware.

4.3 Security objectives for the environment

Identifier	Objective statements
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.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.CONNECTIONS	TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic flowing among attached networks.

4.4 TOE security objectives rationale

The following table demonstrates that all security objectives for the TOE trace back to the threats and OSPs in the security problem definition.

Threats/OSPs	Objectives	Rationale
T.ADMIN_ERROR	O.SYSTEM_MONITORING	T.ADMIN_ERROR is the threat that an administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms. The TOE mitigates this by providing the capability to generate audit data and send those data to an external IT entity. (O.SYSTEM_MONITORING)
T.TSF_FAILURE	O.TSF_SELF_TEST	T.TSF_FAILURE is the threat that security mechanisms of the TOE may fail, leading to a compromise of the TSF. The TOE mitigates this by provide the capability to test some subset of its security functionality to ensure it is operating properly. (O.TSF_SELF_TEST)
T.UNDETECTED_ACTIONS	O.SYSTEM_MONITORING	T.UNDETECTED_ACTIONS is the threat that malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated. The TOE mitigates this by providing the capability to generate audit data and send those data to an external IT entity. (O.SYSTEM_MONITORING)
T.UNAUTHORIZED_ACCESS	O.SYSTEM_MONITORING O.PROTECTED_COMMUNICATIONS	T.UNAUTHORIZED_ACCESS is the threat that a user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an

Threats/OSPs	Objectives	Rationale
	O.SESSION_LOCK O.TOE_ADMINISTRATION	authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data. The TOE mitigates this threat through several security objectives: The TOE will provide the capability to generate audit data and send those data to an external IT entity. (O.SYSTEM_MONITORING) The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities. (O.PROTECTED_COMMUNICATIONS) The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked. (O.SESSION_LOCK) The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators. (O.TOE_ADMINISTRATION).
T.UNAUTHORIZED_UPDATE	O.VERIFIABLE_UPDATES	T.UNAUTHORIZED_UPDATE is the threat that a malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE. The TOE mitigates this threat by providing the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source. (O.VERIFIABLE_UPDATES)

Threats/OSPs	Objectives	Rationale
T.USER_DATA_REUSE	O.RESIDUAL_INFORMATION_CLEARING	T.USER_DATA_REUSE is the threat that user data may be inadvertently sent to a destination not intended by the original sender The TOE mitigates this by ensuring that any data contained in a protected resource is not available when the resource is reallocated. (O.RESIDUAL_INFORMATION_CLEARING)
T.NETWORK_MISUSE	O.SYSTEM_MONITORING	T.NETWORK_MISUSE is the threat that access to services made available by a protected network might be used counter to Operational Environment policies. The TOE mitigates this threat through several security objectives: The TOE will provide the capability to generate audit data and send those data to an external IT entity. (O.SYSTEM_MONITORING)
T.INTRA_NETWORK_DISCLOSURE	O.PROTECTED_COMMUNICATIONS	T.INTRA_NETWORK_DISCLOSURE is the threat that sensitive information on a protected network might be disclosed to an untrusted entity while in transit between the TOE and a trusted network device. The TOE mitigates this threat by ensuring that all communications with trusted network devices are protected through the use of cryptographic tunnelling.
T.STORED_DATA_MODIFICATION	O.DATA_MODIFICATION_DETECTION	T.STORED_DATA_MODIFICATION is the threat that a malicious

Threats/OSPs	Objectives	Rationale
	O.SYSTEM_MONITORING	party may attempt to modify data stored within the TOE or the TOE firmware. The TOE mitigates this threat through several security objectives: The TOE will provide the means to monitor changes to critical system files (O.DATA_MODIFICATION_DETECTION). The TOE will provide the capability to generate audit data and send those data to an external IT entity. (O.SYSTEM_MONITORING)
P.ACCESS_BANNER	O.DISPLAY_BANNER	P.ACCESS_BANNER is an organisational policy whereby 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. The TOE meets this policy by display an advisory warning regarding use of the TOE.

4.5 Environment security objectives rationale

The following table demonstrates that all security objectives for the operational environment all trace back to assumptions or OSPs in the security problem definition.

Assumptions	Objectives	Rationale
A.NO_GENERAL_PURPOSE	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.
A.PHYSICAL	OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
A.TRUSTED_ADMIN	OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
A.CONNECTIONS	OE.CONNECTIONS	TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic flowing among attached networks.

5 Security Requirements

The Security Functional Requirements included in this section are derived from the *Protection Profile for Network Devices, version 1.1*. The Protection Profile, in turn, derives these requirements from Part 2 of the *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 3*, with additional extended functional components.

5.1 Conventions

Part 2 of the CC defines an approved set of operations that may be applied to security functional requirements. This document uses the following font conventions to identify the operations defined by the CC:

- **Assignment.** The assignment operation provides the ability to specify an identified parameter within a requirement. Assignments are indicated using *italicised* text.
- **Selection.** The selection operation allows the specification of one or more items from a list. Selections are indicated using underlined text.
- **Assignment within a Selection:** Indicated with *italicised and underlined* text.
- **Refinement.** The refinement operation allows the addition of extra detail to a requirement. Refinements are indicated using bolded text for **additions**, and strike-through for ~~deletions~~.
- **Iteration.** The iteration operation allows a component to be used more than once with varying operations. Iterations are indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).
- Square brackets (‘[‘ and ‘]’) are used to identify assignment and selection operations performed by the security target author (as opposed to those performed by the protection profile authors).

Explicitly stated SFRs are identified by having a label ‘EXT’ after the requirement name for TOE SFRs.

5.2 TOE Security Functional Requirements

This section identifies the Security Functional Requirements for the TOE. The TOE Security Functional Requirements that appear below in the Table 1 are described in more detail in the following subsections.

Table 1: TOE Security Functional Requirements and Auditable Events

Identifier	Auditable Events	Additional Audit Record Contents

Identifier	Auditable Events	Additional Audit Record Contents
Security audit (FAU)		
FAU_GEN.1	None.	
FAU_GEN.2	None.	
FAU_STG_EXT.1	None.	
Cryptographic support (FCS)		
FCS_CKM.1	None.	
FCS_CKM_EXT.4	None.	
FCS_COP.1(1)	None.	
FCS_COP.1(2)	None.	
FCS_COP.1(3)	None.	
FCS_COP.1(4)	None.	
FCS_RBG_EXT.1 (1)	None.	
FCS_RGB_EXT.1(2)	None	
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA. Establishment/Termination of an IPsec SA.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_TLS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_SSH_EXT.1	Failure to establish an SSH session Establishment/Termination of an SSH session	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
User data protection (FDP)		
FDP_RIP.2	None.	
Identification and authentication (FIA)		
FIA_PMG_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).

Identifier	Auditable Events	Additional Audit Record Contents
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	None.	
FIA_UAU.7	None.	
Security management (FMT)		
FMT_MTD.1	None.	
FMT_SMF.1	None.	
FMT_SMR.2	None.	
Protection of the TOE security functions (FPT)		
FPT_SKP_EXT.1	None.	
FPT_APW_EXT.1	None.	
FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update.	No additional information.
FPT_TST_EXT.1	System startup, shutdown, failures and restarts.	No additional information.
FPT_TST.1	Detection of file integrity errors	Name of file Description of the error
TOE Access (FTA)		
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.
FTA_SSL.4	The termination of an interactive session.	No additional information.
FTA_TAB.1	None.	
Trusted Path/Channels (FTP)		
FTP_ITC.1 (1)	Initiation of the trusted channel.	Identification of the initiator and target of failed trusted channels

Identifier	Auditable Events	Additional Audit Record Contents
	Termination of the trusted channel. Failure of the trusted channel functions.	establishment attempt.
FTP_ITC.1 (2)	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_ITC.1 (3)	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

5.2.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1	The TSF shall be able to generate an audit record of the following auditable events: a) Start-up and shutdown of the audit functions; b) All auditable events for the not specified level of audit; and c) *All administrative actions;* d) *Specifically defined auditable events listed in Table 1*
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 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, *information specified in column three of Table 1*.
Notes:	None

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.
Notes:	None

FAU_STG_EXT.1 External Audit Trail 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 implementing the TLS protocol.
Notes:	Audit data is sent to the NMS, if it is accessible. If the NMS is inaccessible, the audit data will be retroactively be sent once the NMS becomes available again. The TOE may be associated with multiple NMS instances. The NMS connects to the TOE via a TLS socket, which runs within the *Black* IPSec tunnel.

5.2.2 Cryptographic support (FCS)

FCS_CKM.1 Cryptographic Key Generation (for asymmetric keys)

FCS_CKM.1.1	The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with • *NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;* • *NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes* and specified cryptographic key sizes *equivalent to, or greater than, a symmetric key strength of 112 bits*.
Notes:	None.

FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_CKM_EXT.4.1	The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.
Notes:	None.

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

FCS_COP.1.1(1)	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *AES operating in **CBC mode*** and cryptographic key sizes 128-bits, and 256-bits that meets the following: • **FIPS PUB 197, "Advanced Encryption Standard (AES)"** • **NIST SP 800-38A**
Notes:	None.

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

FCS_COP.1.1(2)	The TSF shall perform **cryptographic signature services** in accordance with a ***RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater*** that meets the following: **FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"**.
Notes:	None.

FCS_COP.1(3) Cryptographic Operation (for cryptographic hashing)

FCS_COP.1.1(3)	The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm **SHA-1, SHA-256, SHA-384, SHA-512 and message digest sizes selection: 160, 256, 384, 512 bits** that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*
Notes:	None.

FCS_COP.1(4) Cryptographic Operation (for keyed-hash message authentication)

FCS_COP.1.1(4)	The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm **HMAC SHA-1, SHA-256, SHA-384, SHA-512, key size *160, 256, 384, 512* bits, and message digest sizes 160, 256, 384, 512 bits** that meet the following: *FIPS Pub 198-1, "The KeyedHash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."*
Notes:	None.

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

FCS_RBG_EXT.1.1 (1)	The TSF shall perform all random bit generation (RBG) services in accordance with FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES seeded by an entropy source that accumulated entropy from a software-based noise source.
FCS_RBG_EXT.1.2 (1)	The deterministic RBG shall be seeded with a minimum of 128 bits of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.
Notes:	None.

FCS_RBG_EXT.1 (2) Extended: Cryptographic Operation (Random Bit Generation – HASH DRBG)

FCS_RBG_EXT.1.1 (2)	The TSF shall perform all RBG services in accordance with NIST Special Publication 800-90 using Hash_DRBG (SHA256) seeded by an entropy source that accumulated entropy from a software-based noise source.
FCS_RBG_EXT.1.2 (2)	The deterministic RBG shall be seeded with a minimum of 128 bits of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.
Notes:	None.

FCS_IPSEC_EXT.1 Explicit: IPSEC

FCS_IPSEC_EXT.1.1	The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602), no other algorithms, and using IKEv1 as defined in RFCs 2407, 2408,

	2409, RFC 4109, and no other RFCs for hash functions.
FCS_IPSEC_EXT.1.2	The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.
FCS_IPSEC_EXT.1.3	The TSF shall ensure that IKEv1 SA lifetimes are able to be limited to 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs.
FCS_IPSEC_EXT.1.4	The TSF shall ensure that IKEv1 SA lifetimes are able to be limited to *100* MB of traffic for Phase 2 SAs.
FCS_IPSEC_EXT.1.5	The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and *no other DH groups*.
FCS_IPSEC_EXT.1.6	The TSF shall ensure that all IKE protocols implement Peer Authentication using the *rDSA* algorithm.
FCS_IPSEC_EXT.1.7	The TSF shall support the use of pre-shared keys (as referenced in the RFCs) for use in authenticating its IPsec connections.
FCS_IPSEC_EXT.1.8	The TSF shall support the following: 1. *Pre-shared keys shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”;* 2. *Pre-shared keys of 22 characters and keys of length between (and including) 5 and 127 characters*.
Notes:	None.

FCS_TLS_EXT.1 Explicit: TLS

FCS_TLS_EXT.1.1	The TSF shall implement one or more of the following protocols TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246) supporting the following ciphersuites: **Mandatory Ciphersuites:** TLS_RSA_WITH_AES_128_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA TLS_DHE_RSA_WITH_AES_128_CBC_SHA TLS_DHE_RSA_WITH_AES_256_CBC_SHA **Optional Ciphersuites:** *None*.
Notes:	None.

FCS_SSH_EXT.1 Explicit: SSH

FCS_SSH_EXT.1.	The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252,

1	4253, and 4254.
FCS_SSH_EXT.1.2	The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.
FCS_SSH_EXT.1.3	The TSF shall ensure that, as described in RFC 4253, packets greater than 262144 bytes in an SSH transport connection are dropped.
FCS_SSH_EXT.1.4	The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, and *no other algorithms*.
FCS_SSH_EXT.1.5	The TSF shall ensure that the SSH transport implementation uses SSH_RSA and *no other public key algorithms*] as its public key algorithm(s).
FCS_SSH_EXT.1.6	The TSF shall ensure that data integrity algorithms used in SSH transport connection is *hmac-sha1, hmac-sha1-96*.
FCS_SSH_EXT.1.7	The TSF shall ensure that diffie-hellman-group14-sha1 is the only allowed key exchange method used for the SSH protocol.
Notes:	None.

5.2.3 User data protection (FDP)

FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the *allocation of the resource to* all objects.
Notes:	None.

5.2.4 Identification and authentication (FIA)

FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1	The TSF shall provide the following password management capabilities for administrative passwords: 1. *Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”,_”+”, “-”, “=”, “{”, “}”, “|”, “ [”, “]”, “;”, “'”, “:”, “,”, “.”, “/”, “<”, “>”, “?”,“`”, “~”, “ “;* 2. *Use of the “?” character is not permitted when setting passwords using the TOE command line interface. Setting passwords that include “?” should be performed via the NMS.* 3. *Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;*
Notes:	None

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.
Notes:	None

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

FIA_UAU_EXT.2.	The TSF shall provide a local password-based authentication mechanism, and a

1	*remote password-based authentication mechanism* to perform administrative user authentication.
Notes:	Remote NMS administrators, and remote SSH administrators, use standard password-based mechanisms to authenticate to the TOE.

FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7	The TSF shall provide only *obscured feedback* to the administrative user while the authentication is in progress at the local console.
Notes:	None

5.2.5 Security management (FMT)

FMT_MTD.1 Management of TSF data

FMT_MTD.1.1	The TSF shall restrict the ability to manage the *TSF data* to the S*ecurity Administrators*.
Notes:	None

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1	The TSF shall be capable of performing the following management functions: • *Ability to administer the TOE locally and remotely;* • *Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;* • *Ability to configure the cryptographic functionality.*
Notes:	None

FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1	The TSF shall maintain the roles: **a) Authorized 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 • **Authorized Administrator role shall be able to administer the TOE locally;** • **Authorized Administrator role shall be able to administer the TOE remotely;** are satisfied.
Notes:	None

5.2.6 Protection of the TOE security functions (FPT)

FPT_SKP_EXT.1 Extended: 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.
Notes:	None.

FPT_APW_EXT.1 Extended: 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.
Notes:	None.

FPT_STM.1 Reliable time stamps

FPT_STM.1.1	The TSF shall be able to provide reliable time stamps for its own use.
Notes:	None

FPT_TUD_EXT.1 Extended: Trusted Update

FPT_TUD_EXT.1.1	The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.
FPT_TUD_EXT.1.2	The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.
FPT_TUD_EXT.1.3	The TSF shall provide a means to verify firmware/software updates to the TOE using a [digital signature mechanism] prior to installing those updates.
Notes:	None

FPT_TST_EXT.1 TSF testing

FPT_TST_EXT.1.1	The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.
Notes:	None

FPT_TST.1 TSF self test

FPT_TST.1.1	The TSF shall run a suite of self tests periodically during normal operation to demonstrate the correct operation of the TSF.
FPT_TST.1.2	The TSF shall provide authorised users with the capability to verify the integrity of TSF data.
FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of TSF.
Notes:	The TOE monitors the file integrity of the critical system components that implement the TSF or are required by the TSF for secure operation.

5.2.7 TOE Access (FTA)

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.
Notes:	None

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*.
Notes:	None

FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1	The TSF shall allow Administrator-initiated termination of the Administrator's own interactive session.
Notes:	None

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 use of the TOE.
Notes:	None

5.2.8 Trusted Path/Channels (FTP)

FTP_ITC.1 (1) Inter-TSF Trusted Channel

FTP_ITC.1.1	The TSF shall **use TLS** to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: audit server (NMS management 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 of the channel data**.
FTP_ITC.1.2	The TSF shall permit *the TSF*, ***or the authorized IT entities*** to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for *communication of audit data, all remote management actions from the NMS, software updates from the NMS*.
Notes:	In this case, the Audit server *is* the NMS, however the NMS also allows remote management of the TOE and other capabilities. The TLS link runs inside the *Black* IPsec tunnel (FTP_ITC.1 (2)).

FTP_ITC.1 (2) Inter-TSF Trusted Channel

FTP_ITC.1.1	The TSF shall **use IPSec** to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: ~~audit server~~, remote instance of the TOE, remote SCS-EI gateway** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **disclosure and detection of modification of the channel data**.
FTP_ITC.1.2	The TSF shall permit *the TSF*, ***or the authorized IT entities*** to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for *all communications with the remote authorized IT entities*.
Notes:	This IPSec tunnel envelops and protects all communications between remote instances of the TOE and the SCS-EI. This is known as the *Black Network*.

FTP_ITC.1 (3) Inter-TSF Trusted Channel

FTP_ITC.1.1	The TSF shall **use IPSec** to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: ~~audit server~~, remote instance of the TOE, remote SCS-EI gateway** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from

	disclosure and detection of modification of the channel data.
FTP_ITC.1.2	The TSF shall permit *the TSF*, ***or the authorized IT entities*** to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for [*all communications with the remote authorized IT entities via the Blue network port]*.
Notes:	This IPSec tunnel operates within the *Black Network* (specified in FTP_ITC.1 (2)). It envelops and protects all communications between remote instances of the TOE and the SCS-EI over the *Blue Network* port.

FTP_TRP.1 Trusted Path

FTP_TRP.1.1	The TSF shall **use SSH** provide **a trusted** communication path between itself and **remote administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *disclosure and detection of modification of the communicated data*.
FTP_TRP.1.2	The TSF shall permit **remote administrators** to initiate communication via the trusted path.
FTP_TRP.1.3	The TSF shall require the use of the trusted path for *initial administrator authentication and all remote administration actions*.
Notes:	Remote administration is available via an SSH shell and from the NMS server. The NMS link is over TLS, which is inside an IPsec tunnel. The SSH session is also within the *Black Network*.

5.3 Mapping of SFRs to security objectives for the TOE

Security objective	Mapped SFRs	Rationale
O.PROTECTED_COMMUNICATIONS	FCS_CKM.1, FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2-1), FCS_COP.1(2-2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1 (1), FCS_RBG_EXT.1 (2), FPT_SKP_EXT.1, FTP_ITC.1 (1), FTP_ITC.1 (2), FTP_ITC.1 (3), FTP_TRP.1, (FCS_IPSEC_EXT.1, FCS_SSH_EXT.1, FCS_TLS_EXT.1, FCS_HTTPS_EXT.1)	To address the issues concerning transmitting sensitive data to and from the TOE described in Section 2.1 of the NDPP, "Communications with the TOE", compliant TOEs will provide encryption for these communication paths between themselves and the endpoint. These channels are implemented using one (or more) of three standard protocols: IPsec, TLS/HTTPS, and SSH. These protocols are specified by RFCs that offer a variety of implementation choices. Requirements have been imposed on some of these choices (particularly those for cryptographic primitives) to provide interoperability and resistance to cryptographic attack. While compliant TOEs must support all of the choices specified in the ST, they may support additional algorithms and protocols. If such additional mechanisms are not evaluated, guidance must be given to the administrator to make clear the fact that they are not evaluated. In addition to providing protection from disclosure (and detection of modification) for the communications, each of the protocols described in this document (IPsec, SSH, and TLS/HTTPS) offer two-way authentication of each endpoint in a cryptographically secure manner, meaning that even if there was a malicious attacker between the two endpoints, any attempt to represent themselves to either endpoint of the communications path as the other communicating party would be detected. The requirements on each protocol, in addition to the structure of the protocols themselves, provide protection against replay attacks such as those described in Section 2.1 of the NDPP, usually by including a unique value in each communication so that replay of that communication can be detected.

Security objective	Mapped SFRs	Rationale
O.VERIFIABLE_UPDATES	FPT_TUD_EXT.1, FCS_COP.1(2-1), FCS_COP.1(2-2), FCS_COP.1(3)	As outlined in Section 2.2 of the NDPP, "Malicious Updates", failure by the Security Administrator to verify that updates to the system can be trusted may lead to compromise of the entire system. A first step in establishing trust in the update is to publish a hash of the update that can be verified by the System Administrator prior to installing the update. In this way, the Security Administrator can download the update, compute the hash, and compare it to the published hash. While this establishes that the update downloaded is the one associated with the published hash, it does not indicate if the source of the update/hash combination has been compromised or can't be trusted. So, there remains a threat to the system. To establish trust in the source of the updates, the system can provide cryptographic mechanisms and procedures to procure the update, check the update cryptographically through the TOE-provided digital signature mechanism, and install the update on the system. While there is no requirement that this process be completely automated, administrative guidance documentation will detail any procedures that must be performed manually, as well as the manner in which the administrator ensures that the signature on the update is valid.

Security objective	Mapped SFRs	Rationale
O.SYSTEM_MONITORING	FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1, FPT_STM.1, FPT_TST.1	In order to assure that information exists that allows Security Administrators to discover intentional and unintentional issues with the configuration and/or operation of the system as discussed in Section 2.3 of the NDPP, "Undetected System Activity", compliant TOEs have the capability of generating audit data targeted at detecting such activity. Auditing of administrative activities provides information that may hasten corrective action should the system be configured incorrectly. Audit of select system events can provide an indication of failure of critical portions of the TOE (e.g., a cryptographic provider process not running) or anomalous activity (e.g., establishment of an administrative session at a suspicious time, repeated failures to establish sessions or authenticate to the system) of a suspicious nature. In some instances there may be a large amount of audit information produced that could overwhelm the TOE or administrators in charge of reviewing the audit information. The TOE must be capable of sending audit information to an external trusted entity, which mitigates the possibility that the generated audit data will cause some kind of denial of service situation on the TOE. This information must carry reliable timestamps, which will help order the information when sent to the external device. Loss of communication with the audit server is problematic. While there are several potential mitigations to this threat, this PP does not mandate that a specific action takes place; the degree to which this action preserves the audit information and still allows the TOE to meet its functionality responsibilities should drive decisions on the suitability of the TOE in a particular environment.
O.DISPLAY_BANNER	FTA_TAB.1	Before establishing an administrative user session the TSF shall display a Security Administrator-specified advisory notice and consent warning message regarding use of the TOE.

Security objective	Mapped SFRs	Rationale
O.TOE_ADMINISTRATION	FIA_UIA_EXT.1, FIA_PMG_EXT.1, FIA_UAU.7, FMT_MTD.1, FMT_SMF.1, FMT_SFR.1, FPT_APW_EXT.1, FTA_SSL.3	In order to provide a trusted means for administrators to interact with the TOE, the TOE provides a password-based logon mechanism. The administrator must have the capability to compose a strong password, and have mechanisms in place so that the password must be changed regularly. To avoid attacks where an attacker might observe a password being typed by an administrator, passwords must be obscured during logon. Session locking or termination must also be implemented to mitigate the risk of an account being used illegitimately. Passwords must be stored in an obscured form, and there must be no interface provided for specifically reading the password or password file such that the passwords are displayed in plain text.
O.RESIDUAL_INFORMATION_CLEARING	FDP_RIP.2	In order to counter the threat that user data is inadvertently included in network traffic not intended by the original sender, the TSF ensures that network packets sent from the TOE do not include data "left over" from the processing of previous network information.
O.SESSION_LOCK	FTA_SSL_EXT.1	In order to mitigate the risk of unattended sessions being hijacked, the TOE shall, for local interactive sessions, either terminate or lock the session, after a Security Administrator-specified time period of inactivity. Locking the session will disable any activity of the user’s data access/display devices other than unlocking the session, and the TOE will require that the administrator re-authenticate to unlocking the session.
O.TSF_SELF_TEST	FPT_TST_EXT.1	In order to detect some number of failures of underlying security mechanisms used by the TSF, the TSF will perform self-tests. The extent of this self-testing is left to the product developer, but a more comprehensive set of self-tests should result in a more trustworthy platform on which to develop enterprise architecture.
O.DATA_MODIFICATION_DETECTION	FPT_TST.1	To address the issue of unauthorised firmware or stored data modification, the TOE will implement a file integrity monitoring and detection system. This system will raise an alert when unauthorised modifications are detected, and will cause a log message to be generated.

6 Security Assurance Requirements

This ST implements the Security Assurance Requirements (SARs) of the Protection Profile for Network Devices, version 1.1. The PP draws from the CC Security Assurance Requirements (SARs) to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

While Table 2 contains the complete set of SARs from the CC, the Assurance Activities to be performed by an evaluator are detailed both in section 4.2 of the NDPP, as well as in this Table 2.

Table 2: TOE Security Assurance Requirements

Assurance class	Assurance components
ADV: Development	ADV_FSP.1 Basic Functional Specification
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ATE: Tests	ATE_IND.1 Independent testing - conformance
AVA: Vulnerability assessment	AVA_VAN.1 Vulnerability survey
ALC: Life cycle support	ALC_CMC.1 Labelling of the TOE
	ALC_CMS.1 TOE CM coverage

ALC_CMS.2.2C – The configuration list shall uniquely identify the configuration items

Description	Version	Identifier
SCS-100	Rev A	S/N 100-002
SCS-200	Rev C	S/N 200-1310
TOE Software Build	5.3.6	
TOE Firmware Build	SCS-100 (v23) SCS-200 (v35d)	
TOE Admin Guide	0.3.1 (28 Jul 15)	SCS-100 Admin Guide v0.3.docx
TR Document	1.0.6.1 (30 Oct 14)	M5_SCS_TR_NDPP-NO-FWEXT 1.0.6.1 (30Oct2014).docx
ST Document	1.5.3 (28 Jul 15).	M5_SCS_ST_NDPP-NO-FWEXT 1.5.3 (28Jul2015).docx

7 TOE Summary Specification

7.1 Overview

This chapter provides a high-level description of how the TOE implements the claimed security functional requirements. The TOE implements the following security functions:

- Security Audit
- Cryptographic Support
- User Data Protection
- Identification and Authentication
- Security Management
- Protection of the TOE Security Functionality
- TOE Access
- Trusted Path/Channels
- Traffic Filtering

7.2 Security Audit

The TOE generates audit log records for a large range of events, including security events, configuration changes, user and administrator events, system events and errors. A full list of the TSF-relevant audit events can be found in Table 1. In addition to the events listed in Table 1, all administrative actions performed at the SSH, NMS or Touchscreen interfaces raise auditable events.

For each audit event the following details are recorded:

- date and time of the event;
- relevant system users or process;
- event description;
- success or failure of the event;
- event source (for example, application name); and
- Information and Communications Technology equipment location/identification.

Additional details may be recorded depending on the nature of the event triggering the audit record generation, including those listed in Table 1.

If the TOE is connected to an NMS it will automatically attempt forward audit logs over a dedicated TLS channel (via the *Black Network*) as they are generated. If the TOE is unable to contact an NMS, the next time it establishes a connection it will retroactively upload the log events generated in the period of downtime. Logs may also be manually extracted via the SSH interface and transferred to an NMS or elsewhere.

Log rotation (i.e. deletion of the oldest records) will occur when the local audit storage memory is close to capacity. Prior to log rotation, a system alert will be raised in order to notify operators that the audit store is nearly full. This alert will be logged and will be displayed on the touchscreen interface.

A number of access constraints protect the audit logs against unauthorised access. Direct read/write/delete access to the audit logs can only be performed by an administrator via the SSH restricted shell; all other interfaces provide read-only access. The NMS receives a copy of all audit logs, but is unable to modify or delete logs. The touchscreen interface and the user SSH interface provide commands to display a small subset of the audit logs (most recent events), and will also display any system alerts.

The Security Audit function is designed to meet the following SFRs:

- FAU_GEN.1: The TOE can generate audit records based on a set of auditable events, including those listed in Table 1 and all administrative actions. Each audit record generated contains common information including timestamp, event type and outcome, and more detailed information specific to the event.
- FAU_GEN.2: Each audit record stores the details of the specific user, process or component that caused the auditable event.
- FAU_STG_EXT.1: The TOE is able to transmit audit records to a remote NMS device via a trusted channel implementing the TLS protocol.

7.3 Cryptographic Support

The TOE contains three cryptographic modules (SCS Linux Kernel, OpenSSL and Sun JSSE) that provide FIPS 140-2 validated implementation of the cryptographic services listed in Table 3, 4 and 5 respectively.

The TOE's Kernel cryptographic module implements encryption/decryption algorithms, message digest algorithms, keyed hash algorithms and a random bit generator. The random bit generator conforms to the requirements specified in FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4, section 3 (Using AES). The random bit generator provides input values for the generation of asymmetric key pairs, and is seeded with a value with a minimum size of 128-bit.

Table 3: SCS Linux Kernel Cryptographic Algorithms

Function	Algorithm	Options	Cert#
Encryption/Decryption			
AES (128, 256)	FIPS 197, SP 800-38A	CBC	#3361
Random Number Generation			
PRNG	ANSI X9.31	AES 128	#1365

Function	Algorithm	Options	Cert#
Message Digest			
SHA-1, SHA-2 (224, 256, 384, 512)	FIPS 180-3		#2786
Keyed Hash			
HMAC-SHA-1, HMAC-SHA-2 (256, 384, 512)	FIPS 180-3		#2141

The TOE provides cryptographic services through secondary and tertiary cryptographic libraries (OpenSSL and SunJSSE), as shown in Table 4 and Table 5.

Table 4: OpenSSL Cryptographic Module Algorithms

Function	Algorithm	Options	Cert #
Digital Signature & Asymmetric Key Generation			
DSA	FIPS 186-4	Key Pair Gen (2048)	#952
RSA	FIPS 186-3		#1722
Encryption/Decryption			
AES (128, 256)	FIPS 197, SP 800-38A	CBC	#3360
Random Number Generation			
PRNG	ANSI X9.31	AES 128	#1364
Message Digest			
SHA-1, SHA-2 (256, 384, 512)	FIPS 180-3		#2785
Keyed Hash			
HMAC-SHA-1, HMAC-SHA-2 (256, 384, 512)	FIPS 180-3, FIPS Pub 198-1		#2140

Table 5: SunJSSE Cryptographic Module Algorithms

Function	Algorithm	Options	Cert #
Digital Signature & Asymmetric Key Generation			
DSA	FIPS 186-3	Key Pair Gen (2048)	#951
RSA	FIPS 186-3	GenKey9.31, SigGenPKCS1.5, SigVerPKCS1.5, (2048/3072 with SHA-256/384/512)	#1721
Encryption/Decryption			

Function	Algorithm	Options	Cert #
AES (128, 256)	FIPS 197, SP 800-38A	CBC	#3359
Random Number Generation			
Hash_DRBG-SHA-2 (256)	SP800-90A		#789
Message Digest			
SHA-1, SHA-2 (256, 384, 512)	FIPS 180-3		#2784
Keyed Hash			
HMAC-SHA-1, HMAC-SHA-2 (256, 384, 512)	FIPS 180-3, FIPS Pub 198-1		#2139

The TSF complies with NIST SP 800-56A (Ref. [5]) and NIST 800-56B (Ref. [6]) without extension. Table 9 and Table 10 in Appendix A list the specific sections of those documents containing "shall not", "should" and "should not" statements, and indicate the conformance of the TOE against each statement.

The TOE zeroizes secret and private keys and critical security parameters (CSPs) when they are no longer in use. After each use, the value in volatile memory (RAM) is overwritten with zeroes. When keys are no longer required, they are automatically zeroized on persistent storage, as shown in Table 6. Key pairs generated by the TOE for the purposes of key establishment are not used for any other purpose.

Table 6: Secret/Private Key Zeroization

Name	Zeroized	Zeroization Procedure
IPSec ESP data key	after use	overwrite with zeroes
Private key Black (IPSec)	never	not zeroed
Private key Blue (IPSec)	never	not zeroed
IPSec ESP HMAC key	after use	overwrite with zeroes
SSH datakey	after use	overwrite with zeroes
SSH data HMAC key	after use	overwrite with zeroes
Private key SSH host	never	not zeroed
Private key Black (TLS)	never	not zeroed

Name	Zeroized	Zeroization Procedure
Private key Blue (TLS)	never	not zeroed
Private key Red (TLS)	never	not zeroed

IPSec is used by the TOE to provide end-to-end security for the *Blue Network* and the *Black Network* DMVPNs. The DMVPN service configures encrypted network links as required. Initially a tunnel to the head-end router(s), then direct to other peers as required using information learned from the head-end router.

The packets are first encapsulated in a GRE layer. The GRE packets are then encrypted using IPSec in transport mode. The combination of GRE wrapped in IPSec transport mode provides the same security as just IPSec in tunnel mode, it encrypts the source and destination IPs, but also supports the multicast packets required for the dynamic routing.

IPSec ESP security associations are configured by default to use confidentiality and authentication mode. This mode is enabled through configuration settings to the IKE keying daemon. Since the keying daemon operates in either confidentiality and authentication mode ***or*** confidentiality only mode, confidentiality only mode is therefore disabled.

The TOE implements IKEv1 conformant to RFCs 2407, 2408, 2409 and RFC 4109.

The IKEv1 Phase 1 key exchange is triggered by the DMVPN service, and is configured to use main mode only (aggressive mode is not used). The IKE exchange provides Diffie-Hellman Group 14 utilizing RSA (rDSA) for key agreement. In the IKE exchanges, the TOE negotiates the DH group with the peer. If there are no common supported groups the exchange is terminated, otherwise the highest common group is selected for use.

By default, the TOE uses certificates for the peer authentication process. Pre-shared keys can be used, by reconfiguring the IKE keying daemon. Generation of pre-shared keys is performed according to procedures defined in the TOE operational guidance.

Pre-shared keys can be composed of any combination of upper and lower case letters, numbers, and the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")". Pre-shared keys of 22 characters and keys of length between (and including) 5 and 127 characters are supported.

IKEv1 security association (SA) lifetimes are also configurable by a Security Administrator, and are, by default, limited to 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs. The Security Administrator can also set a phase 2 lifetime in bytes, which should be set to 100MB in an NDPP-compliant configuration. The lifetime of a SA is negotiated during each IKEv1 exchange; if the two lifetime values differ, the smaller of the two values (for both time-based and data-based lifetimes) is chosen in order to remain compatible with the security configuration of each peer.

The TOE implements TLS 1.0, TLS 1.1 and TLS 1.2 without extensions. The use of either RSA (rDSA) or ephemeral Diffie-Hellman key exchange with AES 128/256 CBC and SHA is supported. Table 7 lists the TLS cipher-suites supported by the TOE. The TOE supports TLS client authentication using pre-shared keys.

Table 7: Supported TLS Cipher-suites

Name	Key Agreement	Encryption	Hash
TLS_RSA_WITH_AES_128_CBC_SHA	RSA	AES 128 CBC	SHA
TLS_RSA_WITH_AES_256_CBC_SHA	RSA	AES 256 CBC	SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	Ephemeral Diffie-Hellman	AES 128 CBC	SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	Ephemeral Diffie-Hellman	AES 256 CBC	SHA

The TOE includes an implementation of the SSHv2 protocol that complies with RFC4251, RFC4252, RFC4253 and RFC4254. The SSHv2 implementation uses AES CBC (128 or 256 bit) for transport encryption and may use HMAC-SHA1, HMAC-SHA1-96, HMAC-MD5 or HMAC-MD5-96 for transport data integrity. Diffie-Hellman group 14 (SHA-1) and RSA public keys are used for transport key agreement.

The SSHv2 service supports the use of either password-based or RSA public key-based authentication. Client users authenticating with a password are given a maximum of three attempts to enter the correct password, after which the session automatically terminates. Password-based authentication is the only form of authentication used by default; public key-based authentication may optionally be enabled by a Security Administrator. SSH sessions will automatically terminate after a period of inactivity (configurable by a Security Administrator).

SSHv2 sessions are rekeyed based on either time or transmitted data; these values may be modified by a Security Administrator. In addition to these conditions, SSHv2 sessions always automatically rekeyed once 2^{31} packets have been transferred. Once a rekeying condition has been met (number of packets transferred, bytes of data transferred or time limit reached), a key exchange takes place resulting in a new transport key. As described in RFC4253, the SSH daemon will drop large network packets. Any packet larger than 1MB in size will be dropped by the server.The Cryptographic Support function is designed to meet the following SFRs:

- FCS_CKM.1: The TSF complies with NIST SP 800-56A (Ref. [5]) and NIST 800-56B (Ref. [6]) without extensions.
- FCS_CKM_EXT.4: Keys and CSPs are zeroized in memory and persistent storage automatically.
- FCS_COP.1 (1): See Table 3.
- FCS_COP.1 (2): See **Error! Reference source not found.**. For RSA and DSA a minimum key modulus of 2048 bits is used by the TSF.

- FCS_COP.1 (3): See Table 3.
- FCS_COP.1 (4): See Table 3.
- FCS_RBG_EXT.1 (1): See Table 3 and Table 4.
- FCS_RBG_EXT.1 (2): See Table 5.
- FCS_IPSEC_EXT.1: The TOE provides IPSec as specified by this SFR.
- FCS_TLS_EXT.1: The TOE provides TLS as specified by this SFR.
- FCS_SSH_EXT.1: The TOE provides SSH as specified by this SFR.

7.4 User Data Protection

In order to ensure that any previous information content (such as any residual network packet fragments) residing in the network buffer is made unavailable when new data is assigned to the buffer, the SCS zeroes any memory locations prior to use. This is performed by the memory allocation function in the network driver, prior to returning a reference to the allocated memory.

The TOE also overwrites cryptographic keys and other CSPs in memory after each use, as shown in Table 6.

The User Data Protection function is designed to meet the following SFR:

- FDP_RIP.2: The TOE overwrites resources in network buffers upon allocation of that resource.

7.5 Identification and Authentication

The TOE implements user identification and authentication controls at each of its user interfaces. Use of these security mechanisms is necessary in order to perform TSF-mediated actions or access the majority of non-security enforcing functionality offered by the TOE. The Authentication process requires a valid account on the TOE for each user.

Users of the TOE may access TSF and other functionality via three separate interfaces, each of which implements user identification and authentication controls. The TOE’s user interfaces are:

1. the Touchscreen interface, which provides fast access to user functions (such as VoIP communication) and non-TSF configuration options;
2. the SSH interface, which can be accessed either locally or remotely via the *Black Network* for configuration and low-level access; and
3. the NMS interface, used by a remote NMS to monitor and manage the TOE via the *Black Network*.

By default, each interface uses username and password credentials in order to identify and authenticate users. Passwords may be composed of any combination of upper and lower case letters, numbers and special characters (including: !@#$%^&*()). The minimum password length may be configured by the Security Administrator, and passwords longer than 15 characters are supported. The SSH interface may also be configured by the Security Administrator to utilise public key cryptography for authentication, but this is not enabled by default.

Successful authentication with a correct username & password combination will present the user with a command line shell prompt via the SSH interface, or a list of options via the touchscreen interface. The NMS interface indicates a successful login by returning a login success value to the SCS-NMS. When authenticating with a password, obscured feedback in the form of asterisk (*) characters is provided, rather than echoing back the password itself.

User accounts for approved NMS users are automatically pushed to the TOE by the NMS. The NMS authenticates to the TOE using a preconfigured NMS system administrator account in order to create and deploy new user accounts. Each NMS user is authenticated to their account on the TOE (after entering a username and password) before gaining access to any administrative functionality on the TOE. The NMS system passes the NMS user's authentication data to the TOE, which will authenticate that user against its internal NMS user account. The NMS and TOE also undergo TLS pre-shared key authentication when initiating communications.

The Identification and Authentication function is designed to meet the following SFRs:

- FIA_PMG_EXT.1: The TOE allows passwords comprised of mixed upper-case, lower-case, numeric and special characters and enforces a Security Administrator configurable minimum password length.
- FIA_UIA_EXT.1: Access to TOE user functions, configuration or TSF data is not allowed prior to user identification and authentication.
- FIA_UAU_EXT.2: The TOE provides local and remote password-based authentication.
- FIA_UAU.7: The TOE only provides obscured feedback during user authentication.

7.6 Security Management

The TOE provides two channels for security management and administration:

- The NMS interface; and
- The SSH interface, either via a direct local network connection or via the *Black Network*.

The functionality accessible for each user is controlled through the use of role-based user accounts; configuration of the TSF and access to TSF data is restricted to Security Administrator type accounts. The restriction of functionality based on the user account role is enforced by the operating system. The TOE's third user interface (the Touchscreen interface) does not provide access to any security management or administration functionality.

The SSH interface is accessible either from a direct local Ethernet connection or via the *Black* IPSec network. The SSH interface provided by the TOE is the same, regardless of the network channel used to access it (local or *Black Network*). This interface provides command line access to the TOE's underlying systems for advanced management and configuration.

The SSH interface provides access to two alternative user interfaces: the admin shell and the restricted shell.

The admin shell provides access to TSF-mediated actions and files through the SSH interface, including full "root" access to the TOE. Security Administrators may use this interface to manually apply and verify TOE update packages. In addition to the standard *nix based operating system functions available through the admin shell, two additional command interface shells may be invoked which provide access to additional configuration and administration functions. These shells are the Network Shell and the Front-end Shell.

The Network Shell provides advanced network configuration functionality. Configuration of the TOE's routing, switching and network interfaces may be performed using this shell. The Front-end Shell provides command line access to functions similar to those offered through the Touchscreen interface.

Standard (i.e. non-administrator) user accounts are able to connect to the SSH interface, but only have access to the restricted (user) shell, from which they are unable to access or modify system files or utilise certain functions. The restricted shell offers the following capabilities:

- Display recent logs (read-only)
- Display the current configuration
- Display system information (such as current time, network details, temperature, fan speed, system resource usage)
- Access the Front-end Shell

The remote NMS administration channel provides an interface which may be used by remote NMS devices for administration of the TOE and transmission of audit logs and events. The NMS is a central administration and management device, which provides the ability to manage multiple instances of the TOE using a graphical user interface. The NMS itself is not included in the scope of this evaluation. The TOE provides the following capabilities through this interface:

- Remote management of:
- Quality of service
- Routing policies
- Administrator accounts
- Provision of remote system software and configuration (policy) updates.
- Audit event tracking and extraction.

TOE cryptographic functionality can be configured for each of the trusted channels - SSH, TLS & IPSec. Effecting a specific cryptographic function mode is performed either directly via the command line interface, or via update packages delivered from the management server (NMS).

Trusted Channel	Configurable cryptographic functions
TLS	Cipher, mode, key length, message authenticator, key exchange method.
SSH	Cipher, mode, key length, message authenticator, key exchange method.
IPSec	Cipher, key length, message authenticator, Diffie-Hellman exponentiations,

Configuration of each trusted channel is determined by settings present in the relevant configuration files:

Trusted Channel	Configuration files
TLS	/usr/share/scs-daemon/config/config.d/templates/ssl.def (defaults) /usr/share/scs-daemon/config/config.d/ssl.custom (overrides)
SSH	/etc/ssh/sshd_config
IPSec	/etc/racoon/racoon.conf

The Security Management function is designed to meet the following SFRs:

- FMT_MTD.1: The TOE restricts the ability to manage TSF data to Security Administrators.
- FMT_SMF.1: The TOE provides local and remote interfaces for management of the TOE, including the ability to apply and verify updates and manage cryptographic functions and the firewall.
- FMT_SMR.2: The TOE provides a Security Administrator role and is able to associate that role with those users.

7.7 Protection of the TOE Security Functionality

User authentication data is stored in a standard *nix shadow file. The file is only accessible to the super user and contains salted hashes of each user's password, along with user names and password expiry information. No plaintext or readable passwords are stored on the TOE. When SSH public-key authentication is enabled, the TOE will store public keys in a plaintext file with restricted file permissions.

The TOE stores a number of cryptographic keys and critical security parameters (CSPs) that are used in cryptographic operations. Table 8 lists the keys and CSPs used by the TOE, along with their purpose and the method of storage. Items that are stored in plaintext are protected by the operating system's file permissions and readable only through the SSH admin shell by user accounts with full root permissions (Security Administrators).

Table 8: Cryptographic Keys and CSPs

Name	Type	Purpose	Storage location	Storage method
IPSec ESP data key	AES	Transport Encryption	Memory	Key Schedule
Private key Black (IPSec)	RSA	Authentication	Flash memory	plaintext restricted file perms
Private key Blue (IPSec)	RSA	Authentication	Flash memory	encrypted
IPSec ESP HMAC key	HMAC	Integrity	Memory	plaintext restricted file perms
SSH datakey	AES	Transport Encryption	Memory	key schedule
SSH data HMAC key	HMAC	Integrity	Memory	plaintext restricted file perms
private key SSH host	RSA	Authentication	Flash memory	plaintext restricted file perms
Private key Black (TLS)	RSA	Authentication	Flash memory	java key store, restricted file perms, passphrase to unlock stored in encrypted config file
Private key Blue (TLS)	RSA	Authentication	Flash memory	
Private key Red (TLS)	RSA	Authentication	Flash memory	

The TOE includes a hardware GPS timing module that is capable of providing trusted time stamps. The TOE also implements NTP client/server hybrid functionality. The NTP server functionality will try to gather time stamps from the GPS module and Administrator chosen NTP servers reachable through the secure tunnels. The GPS module is given preference if it can provide a synchronised time feed. The Administrator can also configure NTP servers that can be reached without a functioning tunnel on the black module of the TOE. In normal operation these will not be reachable. However if the user puts a bearer into 'webkit' mode then these NTP servers may become reachable over that bearer and thus be considered as a source of time. This allows a device that cannot bring up tunnels due to incorrect time setting to correct its time setting and establish a tunnel. Once a time source has been found and time set for the first time (each boot), the NTP service restarts and no longer considers the external NTP servers.

The NTP server functionality will provide time stamps to clients on its trusted sides only. The operation of the NTP functionality may be configured by administrators. Trusted time stamps provided by the NTP timing module are used by all functions of the TOE that require the time, including the audit logs.

The following security functions make use of time stamps:

- Security Audit;
- Cryptographic Support;
- Identification and Authentication;
- Security Management;
- Protection of the TOE Security Functionality;
- TOE Access;
- Trusted Path/Channels; and
- Traffic Filtering.

The TOE is capable of receiving software updates and packages from a trusted NMS. All software updates are in RPM format and are digitally signed by the manufacturer. The integrity and authenticity of updates are verified by the RPM Package Manager component during installation. RPM uses the well-known GNU Privacy Guard (GPG) system to verify the authenticity of the cryptographic signatures. Public keys that are used to verify update packages are stored within a GPG key-ring on the device. An update that fails verification should not be installed. Updates may also be manually installed and verified via the SSH administrator interface and the RPM Package Manager.

The TOE performs self-tests upon its internal cryptographic components, including power-on self-tests (POST) during start up and continuous condition tests during operation. Cryptographic functions are inhibited while the POST is being conducted. The self-tests include known answer tests on each of the TOE’s cryptographic algorithms, software integrity tests on the cryptographic modules, and continuous testing of the pseudo random number generator (PRNG). The cryptographic module also checks the integrity of its own binary and library files and the integrity of the Linux kernel binary. If a test fails or an error occurs, all functions are inhibited until the module is reset. The cryptographic module is designed to meet the self-test requirements specified by the FIPS 140-2 standard and therefore should be considered sufficient to verify correct operation of the TOE’s cryptographic algorithm implementations.

Known answer tests exercise the cryptographic algorithms implemented within the TOE using set parameters. The output of each test is compared against a known answer. Software integrity tests use HMAC SHA-256 to verify that the cryptographic modules within the TOE have not been altered.

The output of the hash is compared against a stored hash for verification. The continuous PRNG test exercises the random bit generator, causing it to generate a number of random blocks. Each block is compared against the previously generated block; if they match the test fails.

The TOE implements a host-based intrusion detection system (HIDS), which detects potentially malicious unusual changes to the host system by monitoring changes in files and system resources. It performs log analysis, file integrity checking, policy monitoring, rootkit detection, real-time alerting and active response. It also monitors changes to critical system files and the installation of new software; as well as all failed attempts by users to execute a command with administrator privileges and all modifications to device access permissions.

Alerts generated by the HIDS are logged and also reported to the NMS. Alerts generated by the HIDS contain:

- the unique identifier of the TOE;
- the data and time of the alert;
- a unique reference to the alert (a hash of the alert type, networking domain, time and TOE identifier); and
- the reason for the detection.

Alerts may be viewed at the NMS, the SSH interface or via the Touchscreen interface.

The Protection of the TOE Security Functionality function is designed to meet the following SFRs:

- FPT_SKP_EXT.1: The TOE's file system prevents reading of all pre-shared keys, symmetric keys and private keys.
- FPT_APW_EXT.1: The TOE stores user passwords as salted hashes.
- FPT_STM.1: The TOE includes a hardware GPS module that is able to provide accurate, trusted timestamps, and is also able to synchronise time using NTP over IPSec.
- FPT_TUD_EXT.1: Security Administrators are able to apply and verify TOE updates locally and remotely. Updates are signed with a digital certificate.
- FPT_TST_EXT.1: The TOE executes self-tests during start-up to verify the correct operation of its cryptographic algorithms.
- FPT_TST.1: The TOE performs periodic self-tests to verify the integrity of system files and components.

7.8 TOE Access

Although each of the TOE's user interfaces is accessed in a different way, the logon process is similar. Prior to login, a Security Administrator-configurable warning banner may be displayed, followed by a username and password login prompt. In the case of SSH public-key based authentication no login prompt is displayed (as authentication is performed transparently in the background), except in the case of authentication failure. Five unsuccessful authentication attempts will result in termination of the login session.

Security Administrator access (via the SSH or NMS user interfaces) requires a valid administrator account on the TOE. Security Administrator access is required in order to create or modify user accounts.

All user sessions (local & remote) will terminate automatically after a set period of inactivity, after which the user will need to re-authenticate in order to access the TOE. The automatic termination time period may be configured by a Security Administrator. SSH sessions may be terminated manually by the user by either sending the appropriate command or by interruption of the session (such as terminating the SSH client software). Active users of the NMS interface or the Touchscreen interface may manually exit the session by logging out.

The TOE Access function is designed to meet the following SFRs:

- FTA_SSL_EXT.1: The TOE terminates local user sessions after a Security Administrator-specified period of inactivity.
- FTA_SSL.3: The TOE terminates remote user sessions after a Security Administrator-specified period of inactivity.
- FTA_SSL.4: All users may terminate their own user session by logging out or interrupting their remote session.
- FTA_TAB.1: The TOE displays a Security Administrator-configurable warning banner prior to login.

7.9 Trusted Path/Channels

The TOE provides the ability for multiple communications domains to communicate across a shared external connection, while remaining separated. The TOE is able to form up to three segregated mesh networks with other connected instances of the TOE over existing public (internet) or private networks. The TOE provides local access to these networks via dedicated Ethernet ports.

The TOE utilises industry standard and specialised cryptography to protect and separate outgoing communications. It supports three segregated networks of increasing security requirements simultaneously:

1. The *Black Network* is protected by an IPsec tunnel. This network is designed for use with unsecured or public networks, such as a remote internet gateway. All external communications, including encrypted traffic from the other networks, are routed through this tunnel.
2. The *Blue Network* runs within the *Black Network*, and is segregated and protected by an additional IPsec tunnel. This network is designed for communications with a remote network with higher security requirements than the *Black Network*.

3. The *Red Network* also runs within the *Black Network*, and is segregated and protected by specialised (external) cryptography. This network is designed for communications with a remote network with higher security requirements than the *Blue Network*.

Each of these external connections terminates either at another instance of the TOE or at a remote SCS Enterprise Interface (SCS-EI). The SCS-EI is a network gateway device located within a trusted external network. The SCS-EI and associated remote infrastructure lies outside of the scope of this evaluation. Each of the IPSec channels are used to tunnel general TCP/IP network communications. The TOE acts as a router in order to separate and forward these communications to another instance of the TOE or to a local network. Figure 1 depicts the secure communications features of the TOE (peered TOE instances are not pictured).

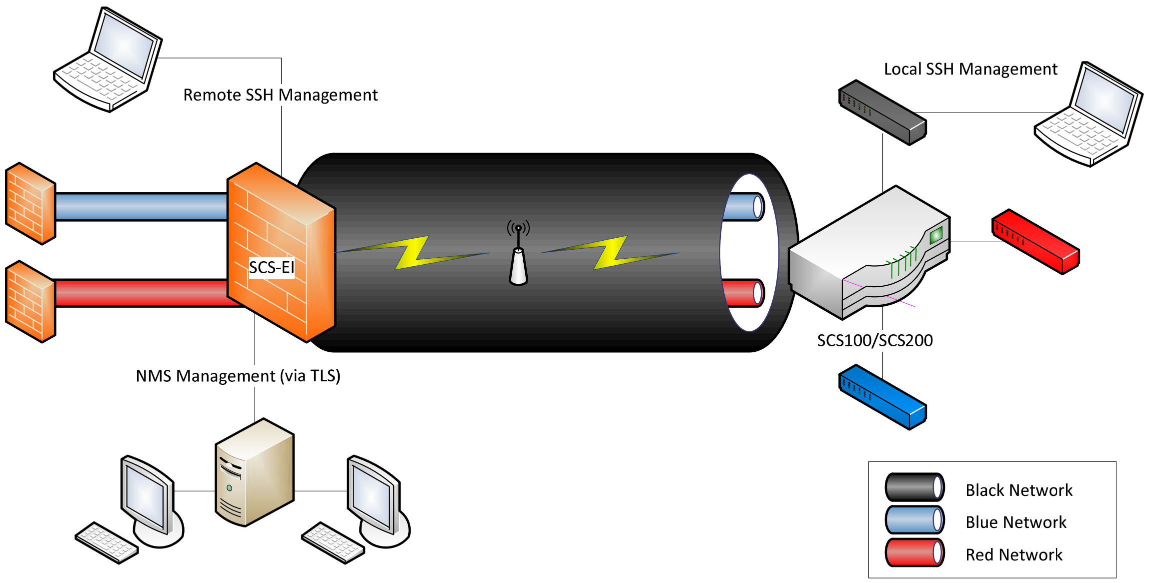

Figure 1 – SCS Secure Communications

In addition to the three tunnelled networks, there are two encrypted management channels, both of which operate within the *Black Network*:

1. Communications with the SSH interface are protected by an isolated SSH path. Communications via this path uses the SSH protocol for remote command-line login and command execution (the SSH channel is not used in tunnel mode by default).
2. Communications with the TOE's NMS management interface are protected by an isolated TLS channel. Communications via this channel are limited to the NMS communications protocol.

The Trusted Path/Channels function is designed to meet the following SFRs:

- FTP_ITC.1 (1): The TOE uses TLS to provide a trusted channel for all communications with the NMS.

- FTP_ITC.1 (2): The TOE uses IPSec to provide a trusted channel for all communications with the SCS-EI or other instances of the TOE.
- FTP_ITC.1 (3): The TOE uses IPSec to provide a trusted channel for all *Blue Network* communications with the SCS-EI or other instances of the TOE.
- FTP_TRP.1: The TOE uses SSH to provide a trusted path for local and remote Security Administrators to administer the TOE.

Appendix A NIST S.P. 800-56A/800-56B Conformance

While the TOE fulfils all of the requirements specified in NIST S.P. 800-56A (Ref. [5]), Table 9 identifies each of the “Should”, “Should Not” and “Shall Not” statements and indicates the conformance of the TOE against these statements specifically.

Table 9: NIST S.P. 800-56A Conformance

Section	Description	Conformant
5.4	When using a nonce, a random nonce **should** be used.	yes
5.5.1.1	The Recommendation for Key Management [7] provides guidance on selecting an appropriate security strength and an appropriate FFC parameter set. If the appropriate security strength does not have an FFC parameter set, then Elliptic Curve Cryptography **should** be used (see Section 5.5.1.2).	yes
5.5.2	The application performing the key establishment on behalf of the party **should** determine whether or not to allow key establishment based upon the method(s) of assurance that was used.	yes
5.6.2	Both the owner and a recipient of a candidate public key **shall** have assurance of its arithmetic validity before using it, as specified below. The application performing the key establishment on behalf of the owner and recipient **should** determine whether or not to allow key establishment based upon the method(s) of assurance that was used.	yes
5.6.2.1	The application performing the key establishment on behalf of the owner **should** determine whether or not to allow key establishment based upon the method(s) of assurance that was used.	yes
5.6.2.2	The application performing the key establishment on behalf of the recipient **should** determine whether or not to allow key establishment based upon the method(s) of assurance that was used.	yes
5.6.2.3	The application performing the key establishment on behalf of the recipient **should** determine whether or not to allow key establishment based upon the method(s) of assurance that was used.	yes
5.6.3.1	The owner of a static public key (or agents trusted to act on the owner’s behalf) **should** determine that the method used for obtaining assurance of the owner’s possession of the correct static private key is sufficient and appropriate to meet the security requirements of the owner’s intended application(s).	yes

Section	Description	Conformant
5.6.3.2.1	The recipient of a static public key (or agents trusted to act on behalf of the recipient) **should** know the method(s) used by the third party, in order to determine that the assurance obtained on behalf of the recipient is sufficient and appropriate to meet the security requirements of the recipient's intended application(s).	yes
5.6.4.1	A public/private key pair **shall** be correctly associated with its corresponding specific set of domain parameters. Each key pair **shall not** be used with more than one set of domain parameters.	yes
5.6.4.2	A static key pair may be used in more than one key establishment scheme. However, one static public/private key pair **shall not** be used for different purposes (for example, a digital signature key pair is not to be used for key establishment or vice versa) with the following possible exception: when requesting the (initial) certificate for a public static key establishment key, the key establishment private key associated with the public key may be used to sign the certificate request. See SP 800-57, Part 1 on Key Usage for further information.	yes
5.6.4.2	An owner and a recipient of a static public key **shall** have assurance of the validity of the public key. This assurance may be provided, for example, through the use of a public key certificate if the CA obtains sufficient assurance of public key validity as part of its certification process. See Section 5.6.2. The application performing the key establishment on behalf of the recipient **should** determine whether or not to allow key establishment based upon the method(s) of assurance that was used.	yes
5.6.4.3	An ephemeral key pair **should** be generated as close to its time of use as possible.	yes
5.6.4.3	A recipient of an ephemeral public key **shall** have assurance of the validity of the public key (see Section 5.6.2). The application performing the key establishment on behalf of the recipient **should** determine whether or not to allow key establishment based upon the method(s) of assurance that was used.	yes
5.8	A static key pair may be used in more than one key establishment scheme. However, one static public/private key pair **shall not** be used for different purposes (for example, a digital signature key pair is not to be used for key establishment or vice versa) with the following possible exception: when requesting the (initial) certificate for a public static key establishment key, the key establishment private key associated with the public key may be used to sign the certificate request. See SP 800-57, Part 1 on Key Usage for further information.	yes

Section	Description	Conformant
5.8	The derived secret keying material may be parsed into one or more keys or other secret cryptographic keying material (for example, secret initialization vectors). If Key Confirmation (KC) or implementation validation testing are to be performed as specified in Section 8 or Section 5.2.3, respectively, then the MAC key **shall** be formed from the first bits of the KDF output and zeroized after its use (i.e., the MAC key **shall not** be used for purposes other than key confirmation or implementation validation testing).	yes
6	Key confirmation may be added to many of these schemes to provide assurance that the participants share the same keying material; see Section 8 for details on key confirmation. Each party **should** have such assurance.	yes
7	DLC-based key transport **shall not** be used with C(2, 0) schemes, or C(1, 1) schemes with the receiver serving as the scheme initiator.	yes
7	A default "rule" of this Recommendation is that ephemeral keys **shall not** be reused (see Section 5.6.4.3). An exception to this rule is that the sender may use the same ephemeral key pair in step 1 above in multiple DLC-based Key Transport transactions if the same secret keying material is being transported in each transaction and if all these transactions occur "simultaneously" (or within a short period of time). However, the security properties of the key establishment scheme may be affected by reusing the ephemeral key in this manner.	yes
9	An ephemeral private key **shall** be zeroized after use and, therefore, **shall not** be recoverable.	yes

While the TOE fulfils all of the requirements specified in NIST S.P. 800-56B (Ref. [6]), Table 10 identifies each of the "Should", "Should Not" and "Shall Not" statements and indicates the conformance of the TOE against these statements specifically.

Table 10: NIST S.P. 800-56B Conformance

Section	Description	Conformant
5.6	When using a nonce, a random nonce **should** be used.	yes
5.8	For the purposes of this Standard, MGF **shall not** be run more than once by each party during a given transaction, using a given MGF seed (i.e., a mask shall be derived at most once from a given MGF seed).	yes
5.9	Non-secret keying material (such as a non-secret initialization vector) **shall not** be generated using the shared secret.	yes
5.9	In all cases, MacKey **shall** be zeroized after its use (in particular, MacKey **shall not** be used for purposes other than key confirmation).	yes

Section	Description	Conformant
6.1	One key pair **shall not** be used for different cryptographic purposes (for example, a digital signature key pair **shall not** be used for key establishment or vice versa) with the following possible exception: when requesting the (initial) certificate for a public key-establishment key, the private key establishment key associated with the public key may be used to sign the certificate request. A key pair may be used in more than one key establishment scheme. However, a key pair used for schemes specified in this recommendation **should not** be used for any schemes not specified herein.	yes
6.1	The owner of a key pair **shall** have assurance of the key pair's validity (see Section 6.4.1); that is, the owner **shall** have assurance that the key pair was generated in an **approved** manner (see Section 6.3), consistent with the criteria of Section 6.2. The owner **shall** have this assurance prior to using the key pair in a key-establishment transaction. By obtaining assurance of key pair validity, the owner of the key pair also obtains assurance of the validity of the public key and assurance of possession of the correct private key. (Additional methods for obtaining owner assurance of private key possession are included in Section 6.5.1.) The owner of the key pair (or agents trusted to act on behalf of the owner) **should** determine that the methods used for obtaining these assurances are sufficient and appropriate to meet the security requirements of the owner's intended application(s).	yes
6.1	A recipient of a public key **shall** have assurance of the validity of the owner's public key (see Section 6.4.2). This assurance may be provided, for example, through the use of a public key certificate if the CA obtains sufficient assurance of public key validity as part of its certification process. The recipient of a public key (or agents trusted to act on behalf of the recipient) **should** determine which method(s) for obtaining these assurances are sufficient and appropriate to meet the security requirements of the owner's intended application(s). The application performing the key establishment on behalf of the recipient **should** determine whether or not to allow the key establishment, based upon the method(s) used to obtain this assurance. Such knowledge may be explicitly provided to the application in some manner, or may be implicitly provided by the operation of the application itself.	yes
6.1	A recipient of a public key **shall** have assurance of the owner's possession of the associated private key (see Section 6.5.2). This assurance may be provided, for example, through the use of a public key certificate if the CA obtains sufficient assurance of possession as part of its certification process. The recipient may also obtain assurance of the owner's possession of the correct private key through the use of key confirmation as specified in this Recommendation (see Section 6.6). The recipient of a public key (or agents trusted to act on behalf of the recipient) **should** determine that the method used for obtaining this assurance is sufficient and appropriate to meet the security requirements of the recipient's intended application(s).	yes

Section	Description	Conformant
6.2.3	The MacKey length **shall** meet or exceed the target security strength, and **should** meet or exceed the security strength of the modulus.	yes
6.5.1	The owner of a public key (or agents trusted to act on the owner's behalf) **should** determine that the method used for obtaining assurance of the owner's possession of the correct private key is sufficient and appropriate to meet the security requirements of the owner's intended application(s)	yes
6.5.2	The recipient of a public key (or agents trusted to act on the recipient's behalf) **should** determine that the method used for obtaining assurance of the owner's possession of the correct private key is sufficient and appropriate to meet the security requirements of the owner's intended application(s).	yes
6.5.2.1	The recipient of a public key (or agents trusted to act on behalf of the recipient) **should** know the method(s) used by the third party, in order to determine that the assurance obtained on behalf of the recipient is sufficient and appropriate to meet the security requirements of the recipient's intended application(s).	yes
6.6	The MacKey **shall not** be used for purposes other than key confirmation or implementation validation testing. (See Sections 5.2, 5.9, 6.2, 8 and 9 for details)	yes
7.1.2	Care **should** be taken to ensure that an implementation of RSADP does not reveal even partial information about the value of k.	yes
7.2.1.3	Care **should** be taken to ensure that an implementation does not reveal information about the encapsulated secret value Z. For instance, the observable behavior of the I2BS routine **should not** reveal even partial information about the byte string Z.	yes
7.2.2.3	Check for RSA-OAEP decryption errors: a. If Y is not a 00 byte, then DecryptErrorFlag = True. b. If HA' does not equal HA, then DecryptErrorFlag = True. c. If X does not have the form PS || 01 || K, where PS consists of zero or more consecutive 00 bytes, then DecryptErrorFlag = True. The type(s) of any error(s) found **shall not** be reported.	yes
7.2.2.3	Care **should** be taken to ensure that the different error conditions that may be detected in Step 5 above cannot be distinguished from one another by an opponent, whether by error message or by process timing. Otherwise, an opponent may be able to obtain useful information about the decryption of a chosen ciphertext C, leading to the attack observed by Manger [17]. A single error message **should** be employed and output the same way for each type of decryption error. There **should** be no difference in the observable behavior for the different RSA-OAEP decryption errors.	yes

Section	Description	Conformant
7.2.2.3	In addition, care **should** be taken to ensure that even if there are no errors, an implementation does not reveal partial information about the encoded message EM. For instance, the observable behavior of the mask generation function **should not** reveal even partial information about the MGF seed employed in the process (since that could compromise portions of the maskedDB′ segment of EM).	yes
7.2.3.3	Care **should** be taken to ensure that the different error conditions in Steps 2.2, 4, and 6 cannot be distinguished from one another by an opponent, whether by error message or timing. Otherwise, an opponent may be able to obtain useful information about the decryption of a chosen ciphertext C, leading to the attack observed by Manger [17]. A single error message **should** be employed and output the same way for each error type. There **should** be no difference in timing or other behavior for the different errors. In addition, care **should** be taken to ensure that even if there are no errors, an implementation does not reveal partial information about the shared secret Z.	yes
7.2.3.3	In addition, care **should** be taken to ensure that an implementation does not reveal information about the encapsulated secret value Z. For instance, the observable behavior of the KDF **should not** reveal even partial information about the Z value employed in the key derivation process.	yes
8	Key confirmation is included in some of these schemes to provide assurance that the participants share the same keying material; see Section 6.6 for the details of key confirmation. When possible, each party **should** have such assurance.	yes
8.3.2	It is extremely important that an implementation not reveal any sensitive information. It is also important to conceal partial information about Z_U, Z_V and Z to prevent chosen-ciphertext attacks on the secret value encapsulation scheme. In particular, the observable behaviour of the key-agreement process **should not** reveal partial information about the shared secret Z.	yes

Appendix B NDPP ALC

Provided here are the version numbers for all artefacts that participated in the NDPP evaluation:

Admin Guide Document: v0.4
Security Target: v1.5.2
Test Report: v1.06.1
SCS Build: v5.3.6
SCS-100: Firmware v23, H/W RevA, S/N: 100-002
SCS-200: Firmware v35d, H/W RevC, S/N: 200-1310

SCS v5.3.6 build checksums:

60021032296962c4d0e1d2885af8a95265978d09cef4ac9cd62207b22f988354	scs-100
96d792c004e6e7cd6a4d5eaed72baa2db5361a349d80aac1ea3a26bc9644bc44	scs-200-black
1f68bf7f8f3f2ce3d20815f5372861b845c45b79c68956bec11ef63f64a75c8b	scs-200-blue
70a55ae5da54b5d7fd319cc836c3efb25c1c1a195b68665524f04a861cfa682f	scs-200-red
b11435a985368923e79603bd5436dfe53e50d9328af712d38ea39fea9281acb9	scs-400-black
315b744739556203d53fbf4faffec5d51953121acce06f1c81f7d104bd95b21e	scs-400-blue
f7894da757e69318ed551ef3f0acdaf334679c7b0016d2fb11134a24ac91ccbd	scs-400-red

SCS v9.3.7 update checksum:

7257076bb778b8136b4d6fd8a8aa1f14062e610622ea23f5d9f0ea46fe5c01f6 scs-version-9.3.7-1.tgz

NDPP verification checklist script:

895d756536a75e6f98b032a63a80b7c1369b054b1665c4ddeff512afb7d25001
/usr/local/bin/scs_ndpp_checklist