www.GossamerSec.com

Assurance Activity Report for Cisco Secure Network Analytics (SNA) 7.4

Version 1.0
02/21/2023

Prepared by:
Gossamer Security Solutions
Accredited Security Testing Laboratory – Common Criteria Testing
Columbia, MD 21045

Prepared for:
National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

Document: AAR-VID11313

Revision History

Revision	Date	Authors	Summary
Version 0.1	01/13/2023	John Messiha	Initial draft
Version 0.2	02/17/2023	Cody Cummins	Addressed ECR comments
Version 0.2	02/21/2023	Cody Cummins	Docs dates/versions updated

The TOE Evaluation was Sponsored by:

Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134

Evaluation Personnel:

- Cody Cummins
- John Messiha

Common Criteria Versions:

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017

Common Evaluation Methodology Versions:

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 5, April 2017

Table of Contents

1. Introduction 6
1.1 Equivalence 6
1.1.1 Evaluated Platform Equivalence 6
1.1.2 CAVP Certificate Justification 7
2. Protection Profile SFR Assurance Activities 9
2.1 Security audit (FAU) 9
2.1.1 Audit Data Generation (NDcPP22e:FAU_GEN.1) 9
2.1.2 User identity association (NDcPP22e:FAU_GEN.2) 11
2.1.3 Security Audit Generation (NDcPP22e:FAU_GEN_EXT.1) 12
2.1.4 Protected Audit Event Storage (NDcPP22e:FAU_STG_EXT.1) 12
2.1.5 Protected Local Audit Event Storage for Distributed TOEs (NDcPP22e:FAU_STG_EXT.4) 16
2.2 Communication (FCO) 18
2.2.1 Component Registration Channel Definition (NDcPP22e:FCO_CPC_EXT.1) 18
2.3 Cryptographic support (FCS) 25
2.3.1 Cryptographic Key Generation (NDcPP22e:FCS_CKM.1) 25
2.3.2 Cryptographic Key Establishment (NDcPP22e:FCS_CKM.2) 29
2.3.3 Cryptographic Key Destruction (NDcPP22e:FCS_CKM.4) 32
2.3.4 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP22e:FCS_COP.1/DataEncryption) 33
2.3.5 Cryptographic Operation (Hash Algorithm) (NDcPP22e:FCS_COP.1/Hash) 38
2.3.6 Cryptographic Operation (Keyed Hash Algorithm) (NDcPP22e:FCS_COP.1/KeyedHash) 40
2.3.7 Cryptographic Operation (Signature Generation and Verification) (NDcPP22e:FCS_COP.1/SigGen) ...41
2.3.8 HTTPS Protocol (NDcPP22e:FCS_HTTPS_EXT.1) 42
2.3.9 NTP Protocol (NDcPP22e:FCS_NTP_EXT.1) 43
2.3.10 Random Bit Generation (NDcPP22e:FCS_RBG_EXT.1) 47
2.3.11 TLS Client Protocol Without Mutual Authentication - per TD0634 (NDcPP22e:FCS_TLSC_EXT.1) ...49
2.3.12 TLS Client Support for Mutual Authentication (NDcPP22e:FCS_TLSC_EXT.2) 59
2.3.13 TLS Server Protocol Without Mutual Authentication - per TD0635 (NDcPP22e:FCS_TLSS_EXT.1) ..60
2.3.14 TLS Server Support for Mutual Authentication (NDcPP22e:FCS_TLSS_EXT.2) 67

2.4 Identification and authentication (FIA) 73
2.4.1 Authentication Failure Management (NDcPP22e:FIA_AFL.1) 73
2.4.2 Password Management (NDcPP22e:FIA_PMG_EXT.1) 76
2.4.3 Protected Authentication Feedback (NDcPP22e:FIA_UAU.7) 77
2.4.4 Password-based Authentication Mechanism (NDcPP22e:FIA_UAU_EXT.2) 78
2.4.5 User Identification and Authentication (NDcPP22e:FIA_UIA_EXT.1) 78
2.4.6 X.509 Certificate Validation (NDcPP22e:FIA_X509_EXT.1/ITT) 82
2.4.7 X.509 Certificate Validation (NDcPP22e:FIA_X509_EXT.1/Rev) 86
2.4.8 X.509 Certificate Authentication (NDcPP22e:FIA_X509_EXT.2) 91
2.4.9 X.509 Certificate Requests (NDcPP22e:FIA_X509_EXT.3) 93
2.5 Security management (FMT) 95
2.5.1 Management of security functions behaviour (NDcPP22e:FMT_MOF.1/ManualUpdate) 95
2.5.2 Management of TSF Data (NDcPP22e:FMT_MTD.1/CoreData) 96
2.5.3 Management of TSF Data (NDcPP22e:FMT_MTD.1/CryptoKeys) 98
2.5.4 Specification of Management Functions - per TD0631 (NDcPP22e:FMT_SMF.1) 99
2.5.5 Restrictions on Security Roles (NDcPP22e:FMT_SMR.2) 101
2.6 Protection of the TSF (FPT) 103
2.6.1 Protection of Administrator Passwords (NDcPP22e:FPT_APW_EXT.1) 103
2.6.2 Basic internal TSF data transfer protection (NDcPP22e:FPT_ITT.1) 104
2.6.3 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) (NDcPP22e:FPT_SKP_EXT.1) 106
2.6.4 Reliable Time Stamps - per TD0632 (NDcPP22e:FPT_STM_EXT.1) 107
2.6.5 TSF testing (NDcPP22e:FPT_TST_EXT.1) 109
2.6.6 Trusted update (NDcPP22e:FPT_TUD_EXT.1) 111
2.7 TOE access (FTA) 116
2.7.1 TSF-initiated Termination (NDcPP22e:FTA_SSL.3) 116
2.7.2 User-initiated Termination (NDcPP22e:FTA_SSL.4) 117
2.7.3 TSF-initiated Session Locking (NDcPP22e:FTA_SSL_EXT.1) 118
2.7.4 Default TOE Access Banners (NDcPP22e:FTA_TAB.1) 119
2.8 Trusted path/channels (FTP) 120
2.8.1 Inter-TSF trusted channel (NDcPP22e:FTP_ITC.1) 120

2.8.2 Trusted Path (NDcPP22e:FTP_TRP.1/Admin) 124
2.8.3 Trusted Path (NDcPP22e:FTP_TRP.1/Join) 125
3. Protection Profile SAR Assurance Activities 130
3.1 Development (ADV) 130
3.1.1 Basic Functional Specification (ADV_FSP.1) 130
3.2 Guidance documents (AGD) 131
3.2.1 Operational User Guidance (AGD_OPE.1) 131
3.2.2 Preparative Procedures (AGD_PRE.1) 133
3.3 Life-cycle support (ALC) 134
3.3.1 Labelling of the TOE (ALC_CMC.1) 134
3.3.2 TOE CM Coverage (ALC_CMS.1) 135
3.4 Tests (ATE) 135
3.4.1 Independent Testing - Conformance (ATE_IND.1) 135
3.5 Vulnerability assessment (AVA) 137
3.5.1 Vulnerability Survey (AVA_VAN.1) 137
3.5.2 Additional Flaw Hypotheses (AVA_VLA.1) 139

1. Introduction

This document presents evaluations results of the Cisco Secure Network Analytics (SNA) 7.4 evaluation. The TOE claims conformance to the following protection profile:

- collaborative Protection Profile for Network Devices Version 2.2e, 23 March 2020 (NDcPP22e)

This document contains a description of the assurance activities and associated results as performed by the evaluators.

1.1 Equivalence

This section explains why the test subset was adequate to address all product installations.

1.1.1 Evaluated Platform Equivalence

Within the Security Target, Section *1.3 TOE Description* includes a table with the following hardware information:

SNA Appliance Type	**Part Number**	**Server platform**	**Entropy Source**
SNA appliances on UCS C-Series M5 servers			
SNA Management Console	ST-SMC2210-K9 L-ST-SMC-VE-K9	UCSC-C220-M5SX	Intel® Skylake Scalable Processor
SNA UDP Director	ST-UDP2210-K9 L-ST-UDP-VE-K9		
SNA Flow Sensor	ST-FS1210-K9 ST-FS3210-K9 ST-FS4210-K9 ST-FS4240-k9 L-ST-FS-VE-K9		
SNA Flow Collector	ST-FC4210-K9 L-ST-FC-VE-K9		
SNA Flow Collector Engine	ST-FC5210E		
SNA Flow Collector Database	ST-FC5210D	UCSC-C240-M5SX	
SNA appliances on UCS C-Series M4 servers			
SNA Management Console	ST-SMC2200-K9 L-ST-SMC-VE-K9	UCSC-C220-M4S	Intel® Xeon® E5-26XX
SNA UDP Director	ST-UDP2200-K9 L-ST-UDP-VE-K9		
SNA Flow Sensor	ST-FS1200-K9 ST-FS2200-K9 ST-FS3200-K9 ST-FS4200-K9 L-ST-FS-VE-K9		
SNA Flow Collector	ST-FC4200-K9 L-ST-FC-VE-K9		
SNA Flow Collector Engine	ST-FC5200E		
SNA Flow Collector Database	ST-FC5200D	UCSC-C240-M4S2	

All physical models of each device type, share the same processor assembly x86_64 and run the same 7.1 software. The Cisco Secure Network Analytics (SNA) Enterprise components that comprise the TOE have common hardware characteristics. Any hardware differences, e.g. the amount of RAM or drive space, or the number of network interfaces, affect only non-TSF relevant functionality such as throughput and amount of storage, and therefore support security equivalency of the TOE component models. Thus one instance of each component type was tested, with the exception of the SNA Flow Collector. The Flow Collector as a physical model can be found in two forms as a standalone Flow Collector or as a pair of appliances with the FC Engine on one appliance and the FC Database on the other appliance. Both the standalone Flow Collector and FC Engine share the same image so it was sufficient to test the FC Engine and FC Database in place of the standalone Flow Collector.

The Security Target also identifies virtual models of each TOE component. These images are identical to the physical images from a TSF relevant functionality standpoint. The only differences being some extra code required for the virtualization process and varying physical hardware. The evaluation team verified this by performing a all testing against the virtual devices as well as the physical devices. It should be noted that the virtual environment only supports a standalone Flow Collector, not the FC Engine and FC Database.

1.1.2 CAVP Certificate Justification

The TOE models, processors, and cryptographic modules included in the evaluation are shown in the following table. The CAVP-certified modules of the TOE are listed in the table below along with the CPU for which they were certified, and the TOE component on which they are used. The cryptographic module used in all TOE platforms is the CiscoSSL FOM 7.2a.

CAVP #	Processor	TOE Appliance / Platform
A1420	Intel Xeon Gold 6130 (Skylake)	ST-SMC2210-k9, ST-FC4210-k9, ST-FC5210E, ST-FC5210D
A1420	Intel Xeon Gold 6254 (Cascade Lake)	ST-FS4240-K9
A1420	Intel Xeon Gold 5118 (Skylake)	ST-FS3210-k9, ST-FS4210-k9, ST-UDP2210-k9
A1420	Intel Xeon Bronze 3106 (Skylake)	ST-FS1210-k9
A1420	Intel Xeon E5-2609 v4 (Broadwell)	ST-FS1200-K9
A1420	Intel Xeon E5-2650 v4 (Broadwell)	ST-FS2200-K9, ST-FS3200-K9, ST-FS4200-K9, ST-UDP2200-K9
A1420	Intel Xeon E5-2680 v3 (Haswell)	ST-SMC2200-K9, ST-FC4200-K9
A1420	Intel Xeon E5-2680 v4 (Broadwell)	ST-FC5200E
A1420	Intel Xeon E5-2695 v3 (Haswell)	ST-FC5200D
A2697	Intel Xeon Gold 6128[1] (Skylake) w/Linux 4 on ESXi 6.7 or 7.0; or Intel Xeon Silver 4116 (Skylake) w/Linux 4 on ESXi 6.7 or 7.0	Cisco UCS C220-M5, or UCS C240-M5 (with any of: L-ST-SMC-VE-K9, L-ST-FC-VE-K9, L-ST-FS-VE-K9, and L-ST-UDP-VE-K9)
A2697	Intel Xeon E5-2609 v4 (Broadwell) w/Linux 4 on ESXi 7.0; or	Cisco UCS C220-M4, or UCS C240-M4 (with any of: L-ST-SMC-VE-K9, L-ST-FC-VE-K9, L-ST-FS-VE-K9, and L-ST-UDP-VE-K9)

[1] While tested on the Intel Xeon Gold 6130 (Skylake), Intel Xeon Gold 6128 (Skylake) may also be used as part of the evaluated configuration.

	Intel Xeon E5-2697 v4 (Broadwell) w/Linux 4 on ESXi 6.7	

2. Protection Profile SFR Assurance Activities

This section of the AAR identifies each of the assurance activities included in the claimed Protection Profiles and describes the findings in each case. The following evidence was used to complete the Assurance Activities:

- Cisco Secure Network Analytics (SNA) 7.4 Security Target, Version 1.2, February 20, 2023 [ST]
- Cisco Secure Network Analytics (SNA) 7.4 Preparative Procedures & Operational User Guide for the Common Criteria Certified Configuration, Version 1.1, February 16, 2023 [Guide]

2.1 Security audit (FAU)

2.1.1 Audit Data Generation (NDcPP22e:FAU_GEN.1)

2.1.1.1 NDcPP22e:FAU_GEN.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.1.1.2 NDcPP22e:FAU_GEN.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: For the administrative task of generating/import of, changing, or deleting of cryptographic keys as defined in FAU_GEN.1.1c, the TSS should identify what information is logged to identify the relevant key.

For distributed TOEs the evaluator shall examine the TSS to ensure that it describes which of the overall required auditable events defined in FAU_GEN.1.1 are generated and recorded by which TOE components. The evaluator shall ensure that this mapping of audit events to TOE components accounts for, and is consistent with, information provided in Table 1, as well as events in Tables 2, 4, and 5 (where applicable to the overall TOE). This includes that the evaluator shall confirm that all components defined as generating audit information for a particular SFR should also contribute to that SFR as defined in the mapping of SFRs to TOE components, and that the audit records generated by each component cover all the SFRs that it implements.

Section 6.1, Table 18 (FAU_GEN.1) in the ST states that each TOE component generates its own audit messages, and stores its own audit messages internally. The types of events that cause audit records to be generated include, cryptography related events, identification and authentication related events, and administrative events. The specific events and the contents of each audit record are identified in Table 15 of the ST. For audit messages related to management of cryptographic keys, the audit message details include the name of the certificate associated with the key (keys cannot be managed independently from the certificates with which they're associated).

Section 9, Table 23, in the ST further maps the SFRS and audit generation to the TOE components. The evaluator confirmed that the audit records generated by each component cover all the SFRS that it implements.

Component Guidance Assurance Activities: The evaluator shall check the guidance documentation and ensure that it provides an example of each auditable event required by FAU_GEN.1 (i.e. at least one instance of each auditable event, comprising the mandatory, optional and selection-based SFR sections as applicable, shall be provided from the actual audit record).

The evaluator shall also make a determination of the administrative actions related to TSF data related to configuration changes. The evaluator shall examine the guidance documentation and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the cPP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are related to TSF data related to configuration changes. The evaluator may perform this activity as part of the activities associated with ensuring that the corresponding guidance documentation satisfies the requirements related to it.

Section *Syslog and Audit Logs* in the **Admin Guide** provides a description of the fields in each audit record in syslog format and a description of the fields in the audit log column via the Web UI.

The *Security Relevant Events* section of the **Admin Guide** provides a list of all auditable events required by FAU_GEN.1. From a review of the ST, the Admin Guide and through testing, the evaluator determined that the list of auditable events includes all administrative actions related to TSF data configuration changes. These administrative actions are consistent with the security requirements implemented in the TOE and were found to have appropriate management capabilities identified in the guidance documentation.

The *Security Relevant Events* section of the **Admin Guide** indicates that many messages have the same format regardless of whether they were generated on SMC or other appliances. There are, however, cases where the SMC message is different than corresponding messages on other appliances. The messages are provided in the following format, where metadata includes date-time and/or process numbers:

<date-time> <hostname> <process>[process-id]: <metadata>,<user>(<user-id>), <message-detail>

The term "metadata" includes date-time and/or process numbers. Additionally, log messages might indicate Syslog severity level; for example, INFO, ERROR, etc.

Component Testing Assurance Activities: The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in the table of audit events and administrative actions listed above. This should include all instances of an event: for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the guidance documentation, and that the fields in each audit record have the proper entries.

For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of auditable events to TOE components in the Security Target. For all events involving more than one TOE component when an audit event is triggered, the evaluator has to check that the event has been audited on both sides (e.g. failure of building up a secure communication channel between the two components). This is not limited to error cases but includes also events about successful actions like successful build up/tear down of a secure communication channel between TOE components.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly.

The evaluator created a list of the required audit events and collected these audit events from each applicable TOE component when running the other security functional tests described by the protection profiles. The evaluator then recorded these audit events for in the proprietary Detailed Test Report (DTR) in the test case for FAU_GEN.1. The security management events are handled in a similar manner. The evaluator collected the required audits for all TOE components, via syslog, during the course of testing. When the administrator was required to set a value for testing, the audit record associated with the administrator action was collected and recorded in the DTR. The evaluator compared the audit events with the required content in the Admin Guide and concluded he audit records were sufficiently addressed.

2.1.2 User identity association (NDcPP22e:FAU_GEN.2)

2.1.2.1 NDcPP22e:FAU_GEN.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

See FAU_GEN.1

Component Guidance Assurance Activities: The TSS and Guidance Documentation requirements for FAU_GEN.2 are already covered by the TSS and Guidance Documentation requirements for FAU_GEN.1.

See FAU_GEN.1

Component Testing Assurance Activities: This activity should be accomplished in conjunction with the testing of FAU_GEN.1.1.

For distributed TOEs the evaluator shall verify that where auditable events are instigated by another component, the component that records the event associates the event with the identity of the instigator. The evaluator shall perform at least one test on one component where another component instigates an auditable event. The evaluator shall verify that the event is recorded by the component as expected and the event is associated with the instigating component. It is assumed that an event instigated by another component can at least be generated for building up a secure channel between two TOE components. If for some reason (could be e.g. TSS or Guidance Documentation) the evaluator would come to the conclusion that the overall TOE does not generate any events instigated by other components, then this requirement shall be omitted.

This test was accomplished in conjunction with the testing of FAU_GEN.1.

2.1.3 Security Audit Generation (NDcPP22e:FAU_GEN_EXT.1)

2.1.3.1 NDcPP22e:FAU_GEN_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.1.4 Protected Audit Event Storage (NDcPP22e:FAU_STG_EXT.1)

2.1.4.1 NDcPP22e:FAU_STG_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.1.4.2 NDcPP22e:FAU_STG_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.1.4.3 NDcPP22e:FAU_STG_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided.

The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access.

The evaluator shall examine the TSS to ensure it describes whether the TOE is a standalone TOE that stores audit data locally or a distributed TOE that stores audit data locally on each TOE component or a distributed TOE that contains TOE components that cannot store audit data locally on themselves but need to transfer audit data to other TOE components that can store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs it contains a list of TOE components that store audit data locally. The evaluator shall examine the TSS to ensure that for distributed TOEs that contain components which do not store audit data locally but transmit their generated audit data to other components it contains a mapping between the transmitting and storing TOE components.

The evaluator shall examine the TSS to ensure that it details the behavior of the TOE when the storage space for audit data is full. When the option 'overwrite previous audit record' is selected this description should include an outline of the rule for overwriting audit data. If 'other actions' are chosen such as sending the new audit data to an external IT entity, then the related behaviour of the TOE shall also be detailed in the TSS.

The evaluator shall examine the TSS to ensure that it details whether the transmission of audit information to an external IT entity can be done in real-time or periodically. In case the TOE does not perform transmission in real-time the evaluator needs to verify that the TSS provides details about what event stimulates the transmission to be made as well as the possible acceptable frequency for the transfer of audit data.

For distributed TOEs the evaluator shall examine the TSS to ensure it describes to which TOE components this SFR applies and how audit data transfer to the external audit server is implemented among the different TOE components (e.g. every TOE components does its own transfer or the data is sent to another TOE component for central transfer of all audit events to the external audit server).

For distributed TOEs the evaluator shall examine the TSS to ensure it describes which TOE components are storing audit information locally and which components are buffering audit information and forwarding the information to another TOE component for local storage. For every component the TSS shall describe the behaviour when local storage space or buffer space is exhausted.

Section 6.1, Table 18 (FAU_STG_EXT.1) in the ST states that each TOE component is configured to export syslog records to a specified, external syslog server. The TOE uses TLS to protect communications with external syslog server. The TOE transmits its audit events in real time to all configured syslog servers at the same time logs are written to the local log buffer.

When the local audit data storage on each TOE component is full, the TOE component will overwrite the oldest stored audit records when writing new audit records. Each TOE component stores its own audit records and each TOE component has the same audit retention rules: each TOE component maintains a log buffer that stores up to 5MB of log messages (the average message size on the TOE is ~180 bytes, so each 5MB file contains an average of ~27,000 messages.), and when the 5MB limit is reached the entire log is overwritten with a new log. A log rotation job runs nightly; and if the active log has grown to 5 MB, it is rotated. This results in the oldest records being purged to allow space for new records to be written.

No administrative interface allows Administrators to clear the local audit logs or to modify the contents of local audit logs.

Section 9, Table 23, in the ST further maps the SFRS and audit generation to the TOE components. The evaluator confirmed that the audit records generated by each component cover all the SFRS that it implements.

Section 6.1, Table 18 (FAU_GEN.1) in the ST states that each TOE component generates its own audit messages, and stores its own audit messages internally.

Component Guidance Assurance Activities: The evaluator shall also examine the guidance documentation to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server.

The evaluator shall also examine the guidance documentation to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and 'cleared' periodically by sending the data to the audit server.

The evaluator shall also ensure that the guidance documentation describes all possible configuration options for FAU_STG_EXT.1.3 and the resulting behaviour of the TOE for each possible configuration. The description of possible configuration options and resulting behaviour shall correspond to those described in the TSS.

Section *Review External Service Certificate Requirements* of the **Admin Guide** describes the requirements for certificates of external services such as a remote syslog server. Section A*dd External Service Certificates to the Trust Stores* describes how to add the relevant certificates for external services, including a remote syslog server, to the appliances trust store. This section states that the Root CA of the syslog server should be added to all of the SNA components. Section *Audit Log Destination* describes the configuration process for enabling sending audits to a remote syslog server over TLS from each of the TOE appliances.

Section *Syslog and Audit Logs* of the **Admin Guide** describes that, audits are sent both to a remote syslog server and stored locally. This section describes the format for both audits sent as syslog messages and the audits when they are stored locally. This section also states that when the local audit log file reaches the maximum size (5MB), a new audit log file is started. A log rotation job runs nightly; and if the active log has grown to 5 MB, it is rotated. This results in the oldest records being purged to allow space for new records to be written.

Security Relevant Events in the **Admin Guide** all log messages are transmitted to a remote syslog server, except for messages indicating connection failures to syslog servers. As messages are transmitted to remote syslog servers, those messages are also written to circular log files, which can be viewed on each appliance console. Each local log file is rotated/replaced when it reaches 5MB. Messages for each appliance are also visible in the Audit Log, which is accessible via Central Management on SMC. A log rotation job runs nightly; and if the active log has grown to 5 MB, it is rotated. This results in the oldest records being purged to allow space for new records to be written.

Component Testing Assurance Activities: Testing of the trusted channel mechanism for audit will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall perform the following additional test for this requirement:

a) Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing. The evaluator shall verify that the TOE is capable of transferring audit data to an external audit server automatically without administrator intervention.

b) Test 2: The evaluator shall perform operations that generate audit data and verify that this data is stored locally. The evaluator shall perform operations that generate audit data until the local storage space is exceeded and verifies that the TOE complies with the behaviour defined in FAU_STG_EXT.1.3. Depending on the configuration this means that the evaluator has to check the content of the audit data when the audit data is just filled to the maximum and then verifies that

1) The audit data remains unchanged with every new auditable event that should be tracked but that the audit data is recorded again after the local storage for audit data is cleared (for the option 'drop new audit data' in FAU_STG_EXT.1.3).

2) The existing audit data is overwritten with every new auditable event that should be tracked according to the specified rule (for the option 'overwrite previous audit records' in FAU_STG_EXT.1.3)

3) The TOE behaves as specified (for the option 'other action' in FAU_STG_EXT.1.3).

c) Test 3: If the TOE complies with FAU_STG_EXT.2/LocSpace the evaluator shall verify that the numbers provided by the TOE according to the selection for FAU_STG_EXT.2/LocSpace are correct when performing the tests for FAU_STG_EXT.1.3.

d) Test 4: For distributed TOEs, Test 1 defined above should be applicable to all TOE components that forward audit data to an external audit server. For the local storage according to FAU_STG_EXT.1.2 and FAU_STG_EXT.1.3 the Test 2 specified above shall be applied to all TOE components that store audit data locally. For all TOE components that store audit data locally and comply with FAU_STG_EXT.2/LocSpace Test 3 specified above shall be applied. The evaluator shall verify that the transfer of audit data to an external audit server is implemented.

Test 1: The external audit server was utilizing rsyslogd version 8.16.0. The successful configuration and implementation of the TLS channel to an external syslog server was demonstrated in FTP_ITC.1 test 1. All syslog data was sent encrypted/not in plaintext.

Test 2: Each TOE component stores its own audit data locally. The evaluator verified that when the local audit storage was filled on each TOE component, the existing audit data was overwritten using the following rule: Overwrite oldest records first.

Test 3: Not applicable. The TOE does not claim FAU_STG_EXT.2/LocSpace.

Test 4: This test was performed as part of test 1 and test 2. All TOE components store their audits locally and remotely by sending them to an external syslog server.

2.1.5 Protected Local Audit Event Storage for Distributed TOEs (NDcPP22e:FAU_STG_EXT.4)

2.1.5.1 NDcPP22e:FAU_STG_EXT.4.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator examines the TSS to confirm that it describes which TOE components store their security audit events locally and which send their security audit events to other TOE components for local storage. For the latter, the target TOE component(s) which store security audit events for other TOE components shall be identified. For every sending TOE component the corresponding receiving TOE component(s) need to be identified. For every transfer of audit information between TOE components it shall be described how the data is secured during transfer according to FTP_ITC.1 or FPT_ITT.1.

For each TOE component which does not store audit events locally by itself, the evaluator confirms that the TSS describes how the audit information is buffered before sending to another TOE component for local storage.

Section 6.1, Table 18 (FAU_GEN.1) in the ST states that each TOE component generates its own audit messages, and stores its own audit messages internally.

Component Guidance Assurance Activities: The evaluator shall examine the guidance documentation to ensure that it describes how the link between different TOE components is established if audit data is exchanged between TOE components for local storage. The guidance documentation shall describe all possible configuration options for local storage of audit data and provide all instructions how to perform the related configuration of the TOE components.

The evaluator shall also ensure that the guidance documentation describes for every TOE component which does not store audit information locally how audit information is buffered before transmission to other TOE components.

Section 6.1, Table 18 (FAU_GEN.1) in the ST states that each TOE component generates its own audit messages, and stores its own audit messages locally. None of the TOE components transmit their audit messages to other TOE components. The **Admin Guide** describes how to configure each TOE component to send their own audit messages to an external syslog server as described in FAU_GEN.1.

Component Testing Assurance Activities: For at least one of each type of distributed TOE components (sensors, central nodes, etc.), the following tests shall be performed using distributed TOEs.

Test 1: For each type of TOE component, the evaluator shall perform a representative subset of auditable actions and ensure that these actions cause the generation of appropriately formed audit records. Generation of such records can be observed directly on the distributed TOE component (if there is appropriate interface), or indirectly after transmission to a central location.

Test 2: For each type of TOE component that, in the evaluated configuration, is capable of transmitting audit information to the external audit server (as specified in FTP_ITC.1), the evaluator shall configure a trusted channel and confirm that audit records generated as a result of actions taken by the evaluator are securely transmitted. It is sufficient to observe negotiation and establishment of the secure channel with the TOE component and the subsequent transmission of encrypted data to confirm this functionality. Alternatively, the following steps shall be performed: The evaluator induces audit record transmission, then reviews the packet capture around the time of transmission and verifies that no audit data is transmitted in the clear.

Test 3: For each type of TOE component that, in the evaluated configuration, is capable of transmitting audit information to another TOE component (as specified in FTP_ITT.1 or FTP_ITC.1, respectively), the evaluator shall configure a secure channel and confirm that audit records generated as a result of actions taken by the evaluator are securely transmitted. It is sufficient to observe negotiation and establishment of the secure channel with the TOE component and the subsequent transmission of encrypted data to confirm this functionality. Alternatively, the following steps shall be performed: The evaluator induces audit record transmission, then reviews the packet capture around the time of transmission and verifies that no audit data is transmitted in the clear.

While performing these tests, the evaluator shall verify that the TOE behaviour observed during testing is consistent with the descriptions provided in the TSS and the Guidance Documentation. Depending on the TOE configuration, there might be a large number of different possible configurations. In such cases, it is acceptable to perform subset testing, accompanied by an equivalency argument describing the evaluator's sampling methodology.

Test 1 - The verification of TOE audits for each of the distributed TOE components was demonstrated as part of FAU_GEN.1.

Test 2 - The successful configuration and implementation of the TLS channel to an external syslog server was demonstrated in FTP_ITC.1 test 1. All syslog data was sent encrypted/not in plaintext.

Test 3 - All TOE components store audits locally and remotely on their own. They do not send their audit data to another TOE component.

2.2 Communication (FCO)

2.2.1 Component Registration Channel Definition (NDcPP22e:FCO_CPC_EXT.1)

2.2.1.1 NDcPP22e:FCO_CPC_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.1.2 NDcPP22e:FCO_CPC_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.2.1.3 NDcPP22e:FCO_CPC_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: If the TOE is not a distributed TOE then no evaluator action is necessary. For a distributed TOE the evaluator carries out the activities below. In carrying out these activities the evaluator shall determine answers to the following questions based on a combination of documentation analysis and testing (possibly also using input from carrying out the Evaluation Activities for the relevant registration channel, such as FTP_TRP.1(2)/Join), and shall report the answers.

Questions for which the evaluator needs to determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities:

a) What stops [The intent of the phrasing 'what stops...' as opposed to 'what secures...' is for the evaluator to pursue the answer to its lowest level of dependency, i.e. a level at which the security can clearly be seen to depend on things that are under appropriate control. For example, a channel may be protected by a public key that is provided to the relying party in a self-signed certificate. This enables cryptographic mechanisms to be applied to provide authentication (and therefore invites an answer that 'the check on the public key certificate secures...'), but does not ultimately stop an attacker from apparently authenticating because the attacker can produce their own self-signed certificate. The question 'what stops an unauthorised component from successfully communicating...' focuses attention on what an attacker needs to do, and therefore pushes the answer down to the level of whether a self-signed certificate could be produced by an attacker. Similarly a well-known key, or a key that is common to a type of device rather than an individual device, may be used in a confidentiality mechanism but does not provide confidentiality because an attacker can find the well-known key or obtain his own instance of a device containing the non-unique key.] a component from successfully communicating with TOE components (in a way that enables it to participate as part of the TOE) before it has properly authenticated and joined the TOE?

b) What is the enablement step? (Describe what interface it uses, with a reference to the relevant section and step in the operational guidance).

1) What stops anybody other than a Security Administrator from carrying out this step?

2) How does the Security Administrator know that they are enabling the intended component to join? (Identification of the joiner might be part of the enablement action itself or might be part of secure channel establishment, but it must prevent unintended joining of components)

c) What stops a component successfully joining if the Security Administrator has not carried out the enablement step; or, equivalently, how does the TOE ensure that an action by an authentic Security Administrator is required before a component can successfully join?

d) What stops a component from carrying out the registration process over a different, insecure channel?

e) If the FTP_TRP.1(2)/Join channel type is selected in FCO_CPC_EXT.1.2 then how do the registration process and its secure channel ensure that the data is protected from disclosure and provides detection of modification?

f) Where the registration channel does not rely on protection of the registration environment, does the registration channel provide a sufficient level of protection (especially with regard to confidentiality) for the data that passes over it?

g) Where the registration channel is subsequently used for normal internal communication between TOE components (i.e. after the joiner has completed registration), do any of the authentication or encryption features of the registration channel result in use of a channel that has weaker protection than the normal FPT_ITT.1 requirements for such a channel?

h) What is the disablement step? (Describe what interface it uses, with a reference to the relevant section and step in the operational guidance).

i) What stops a component successfully communicating with other TOE components if the Security Administrator has carried out the disablement step?

The evaluator shall examine the TSS to confirm that it:

a) Describes the method by which a Security Administrator enables and disables communications between pairs of TOE components.

b) Describes the relevant details according to the type of channel in the main selection made in FCO_CPC_EXT.1.2:

- First type: the TSS identifies the relevant SFR iteration that specifies the channel used

- Second type: the TSS (with support from the operational guidance if selected in FTP_TRP.1.3/Join) describes details of the channel and the mechanisms that it uses (and describes how the process ensures that the key is unique to the pair of components) - see also the Evaluation Activities for FTP_TRP.1(2)/Join.

The evaluator shall confirm that if any aspects of the registration channel are identified as not meeting FTP_ITC.1 or FPT_ITT.1, then the ST has also selected the FTP_TRP.1(2)/Join option in the main selection in FCO_CPC_EXT.1.2.

Section 6.1, Table 18 (FCO_CPC_EXT.1) in the ST states that in order for a new TOE component to become part of the distributed TOE it must successfully complete a registration process. The SMC is the management server within this distributed TOE and each other appliance type (FC, FS, and UDPD) must individually register with the SMC in order to become part of the TOE. Registration of a new appliance cannot be initiated by the SMC. Registration is initiated by the administrator performing the initial configuration of the FC, FS, or UDPD. During the initial configuration process the administrator is asked whether the new appliance will be managed by an SMC, and after answering affirmatively that administrator is prompted to provide the IP address or hostname of the SMC and a valid SMC administrator username and password to authenticate to the SMC. During that initial TLS session, the SMC and the new appliance exchange their unique X.509 certificates. Once the certificates have been exchanged, the new appliance has been joined with the SMC and all subsequent (automated) communications between the SMC and managed appliances use X.509 certificates for authentication in accordance with FPT_ITT.1 and FIA_X509_EXT.1/ITT.

The SMC administrator can de-register an appliance by selecting "Remove This Appliance" from the Central Management page of the WebUI on SMC. A local administrator of the managed appliance can remove the registration by selecting “RemoveAppliance” from the System Configuration menu. Either of these actions, whether initiated from SMC or from the managed appliance, will result in each TOE component deleting the device association from its local configuration, and no further TSF data will be exchanged between the appliances without completing a new registration process.

Section 6.1, Table 18 (FPT_ITT.1) in the ST states that the distributed components of the TOE communicate with each other using TLSv1.2. The SMC (SNA Management Console) communicates with each other appliance (FC, FS, and UDPD), and the TLS sessions can be initiated in either direction.

Section 6.1, Table 18 (FTP_TRP.1/Join) in the ST states that during installation of the TOE the SMC is configured first, then other appliances are joined to the SMC and after successful registration those appliances are managed via the SMC. When TOE components are joined to the SMC, the administrator installing the non-SMC appliance (FC, FS, or UDPD) inputs the IP address or DNS-resolvable hostname of the SMC, then the new appliance initiates a TLS session to the SMC. This initial TLS session used for joining is authenticated using a valid SMC administrator username and password that is manually entered by the administrator performing the initial configuration of the new appliance. If authentication succeeds, , the SMC server certificate fingerprint is displayed for the administrator to confirm, and if the administrator approves the certificate the SMC and the joining appliance exchange certificates, and subsequent TLS connections between the TOE components are authenticated using X.509 certificate-based authentication.

No supplemental support is required from the operational environment to secure the TLS connections between the SMC and other TOE components that are being joined to the SMC.

If attempts to join SNA appliances to an SMC fail:

- No FPT_ITT.1 TLS session (using certificate-based authentication per FIA_X509_EXT.1/ITT) can be established between the devices, and thus no TSF data can be exchanged between the devices until registration successfully completes.
- If the connection failed due to network connectivity error, resolve the connectivity, and reattempt joining.
- If the connection fails due to incorrect information provided by the administrator of the joining component (e.g. incorrect hostname/IP address of the SMC, or incorrect account name or password for the SMC), correct the information, and reattempt joining.

Component Guidance Assurance Activities: (Note: paragraph 274 lists questions for which the evaluator needs to determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.)

The evaluator shall examine the guidance documentation to confirm that it contains instructions for enabling and disabling communications with any individual component of a distributed TOE. The evaluator shall confirm that the method of disabling is such that all other components can be prevented from communicating with the component that is being removed from the TOE (preventing the remaining components from either attempting to initiate communications to the disabled component, or from responding to communications from the disabled component).

The evaluator shall examine the guidance documentation to confirm that it includes recovery instructions should a connection be unintentionally broken during the registration process.

If the TOE uses a registration channel for registering components to the TOE (i.e. where the ST author uses the FTP_ITC.1/FPT_ITT.1 or FTP_TRP.1(2)/Join channel types in the main selection for FCO_CPC_EXT.1.2) then the evaluator shall examine the Preparative Procedures to confirm that they:

a) describe the security characteristics of the registration channel (e.g. the protocol, keys and authentication data on which it is based) and shall highlight any aspects which do not meet the requirements for a steady-state inter-component channel (as in FTP_ITC.1 or FPT_ITT.1)

b) identify any dependencies between the configuration of the registration channel and the security of the subsequent inter-component communications (e.g. where AES-256 inter-component communications depend on transmitting 256 bit keys between components and therefore rely on the registration channel being configured to use an equivalent key length)

c) identify any aspects of the channel can be modified by the operational environment in order to improve the channel security, and shall describe how this modification can be achieved (e.g. generating a new key pair, or replacing a default public key certificate).

As background for the examination of the registration channel description, it is noted that the requirements above are intended to ensure that administrators can make an accurate judgement of any risks that arise from the default registration process. Examples would be the use of self-signed certificates (i.e. certificates that are not chained to an external or local Certification Authority), manufacturer-issued certificates (where control over aspects such as revocation, or which devices are issued with recognised certificates, is outside the control of the operational environment), use of generic/non-unique keys (e.g. where the same key is present on more than one instance of a device), or well-known keys (i.e. where the confidentiality of the keys is not intended to be strongly protected - note that this need not mean there is a positive action or intention to publicise the keys).

In the case of a distributed TOE for which the ST author uses the FTP_TRP.1(2)/Join channel type in the main selection for FCO_CPC_EXT.1.2 and the TOE relies on the operational environment to provide security for some aspects of the registration channel security then there are additional requirements on the Preparative Procedures as described in section 3.5.1.2.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode after the certificates are updated, to ensure that only compliant algorithms are used.

Section *Secure Installation* in the **Admin Guide** provides general guidance for the initial configuration of the appliances, the registration process and specific guidance for the evaluated configuration. This section also refers to the SNA x210 Series Hardware Installation Guide [SNA-HIG], the SNA Virtual Edition (VE) Appliance Installation Guide [SNA-VEIG], and the Secure Network Analytics System Configuration Guide [SNA-CG] for detailed instructions including the configuration order of the appliances. During the initial configuration process the administrator is asked whether the new appliance will be managed by an SMC, and after answering affirmatively, that administrator is prompted to trust the SMC certificate, provide the IP address or hostname of the SMC and a valid SMC administrator username and password to authenticate to the SMC. If authentication succeeds, the SMC and the joining appliance exchange certificates, and subsequent TLS connections between the TOE components are authenticated using X.509 certificate-based authentication.

The *Configuration for Common Criteria Compliance* section of the **Admin Guide** indicates that the first step in configuring the TOE for compliance is to complete all the certificates requirements and procedures. Section *Certificates* of the **Admin Guide** indicates that the communication between TOE appliances is authenticated using x509v3 certificates. This section defines the best practices and procedures for replacing the certificates of the TOE appliances to update the initial self-signed certificates that were used as part of registration, thus increasing the security of the distributed TOE channel.

Section *Secure Installation* section in the **Admin Guide** describes how to disable communication between distributed components. This section also provides recovery instructions If a connection fails between the SMC and the managed appliances, the connections will automatically retry and restore connectivity. Connections will always be temporarily unavailable when appliances are rebooting.

Component Testing Assurance Activities: (Note: paragraph 274 lists questions for which the evaluator needs to determine and report answers through the combination of the TSS, Guidance Documentation, and Tests Evaluation Activities.)

The evaluator shall carry out the following tests:

a) Test 1.1: the evaluator shall confirm that an IT entity that is not currently a member of the distributed TOE cannot communicate with any component of the TOE until the non-member entity is enabled by a Security Administrator for each of the non-equivalent TOE components [An 'equivalent TOE component' is a type of distributed TOE component that exhibits the same security characteristics, behaviour and role in the TSF as some other TOE component. In principle a distributed TOE could operate with only one instance of each equivalent TOE component, although the minimum configuration of the distributed TOE may include more than one instance (see discussion of the minimum configuration of a distributed TOE, in section B.4). In practice a deployment of the TOE may include more than one instance of some equivalent TOE components for practical reasons, such as performance or the need to have separate instances for separate subnets or VLANs.] that it is required to communicate with (non-equivalent TOE components are as defined in the minimum configuration for the distributed TOE)

b) Test 1.2: the evaluator shall confirm that after enablement, an IT entity can communicate only with the components that it has been enabled for. This includes testing that the enabled communication is successful for the enabled component pair, and that communication remains unsuccessful with any other component for which communication has not been explicitly enabled.

Some TOEs may set up the registration channel before the enablement step is carried out, but in such a case the channel must not allow communications until after the enablement step has been completed.

The evaluator shall repeat Tests 1.1 and 1.2 for each different type of enablement process that can be used in the TOE.

c) Test 2: The evaluator shall separately disable each TOE component in turn and ensure that the other TOE components cannot then communicate with the disabled component, whether by attempting to initiate communications with the disabled component or by responding to communication attempts from the disabled component.

d) Test 3: The evaluator shall carry out the following tests according to those that apply to the values of the main (outer) selection made in the ST for FCO_CPC_EXT.1.2.

1) If the ST uses the first type of communication channel in the selection in FCO_CPC_EXT.1.2 then the evaluator tests the channel via the Evaluation Activities for FTP_ITC.1 or FPT_ITT.1 according to the second selection - the evaluator shall ensure that the test coverage for these SFRs includes their use in the registration process.

2) If the ST uses the second type of communication channel in the selection in FCO_CPC_EXT.1.2 then the evaluator tests the channel via the Evaluation Activities for FTP_TRP.1(2)/Join.

3) If the ST uses the 'no channel' selection then no test is required.

e) Test 4: The evaluator shall perform one of the following tests, according to the TOE characteristics identified in its TSS and operational guidance:

1) If the registration channel is not subsequently used for inter-component communication, and in all cases where the second selection in FCO_CPC_EXT.1.2 is made (i.e. using FTP_TRP.1(2)/Join) then the evaluator shall confirm that the registration channel can no longer be used after the registration process has completed, by attempting to use the channel to communicate with each of the endpoints after registration has completed.

2) If the registration channel is subsequently used for intercomponent communication then the evaluator shall confirm that any aspects identified in the operational guidance as necessary to meet the requirements for a steady-state intercomponent channel (as in FTP_ITC.1 or FPT_ITT.1) can indeed be carried out (e.g. there might be a requirement to replace the default key pair and/or public key certificate).

f) Test 5: For each aspect of the security of the registration channel that operational guidance states can be modified by the operational environment in order to improve the channel security (cf. AGD_PRE.1 refinement item 2 in (cf. the requirements on Preparative Procedures in 3.5.1.2), the evaluator shall confirm, by following the procedure described in the operational guidance, that this modification can be successfully carried out.

Test 1.1 and 1.2 - The evaluator viewed via wireshark that no communication was occurring between the TOE components before registration. Since there was no communication, the resulting packet capture was empty. The registration of the distributed TOE components to the SMC was demonstrated in FTP_TRP.1/Join. After the join the evaluator saw the distributed TOE communication which was demonstrated in FPT_ITT.1.

Test 2 - The evaluator deleted a component from the SMC and viewed that the device was no longer registered and that all distributed communications between the SMC and the TOE component had stopped as a result of the deletion. The evaluator repeated this by deleting each component in turn.

Test 3 – This test was met by FTP_TRP.1/Join.

Test 4 – This test was performed after the registration in FTP_TRP.1/Join and before the deletion in FCO_CPC_EXT.1, Test 2. Once a Distributed TOE component has been registered to the SMC, there exists no interface to reinitiate the registration channel from the component as the central management settings are already configured. It falls on the SMC to ensure that it rejects any attempt by a device to use a previously existing registration channel. In order to attempt to reuse a registration channel, the evaluator configured a brand new device matching the configuration settings of a currently registered device. The evaluator then attempted to register the new device to the SMC and viewed that the SMC rejected the registration attempt due to detecting a reuse attempt of a previously registered device.

Test 5 – The Admin Guide describes how to update the identity certs used by each TOE component, which are used in the distributed TOE TLS communication. The evaluator demonstrated the successful updating of these certificates as part of FIA_X509_EXT.3.

2.3 Cryptographic support (FCS)

2.3.1 Cryptographic Key Generation (NDcPP22e:FCS_CKM.1)

2.3.1.1 NDcPP22e:FCS_CKM.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall ensure that the TSS identifies the key sizes supported by the TOE. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

Section 6.1, Table 18 (FCS_CKM.1) in the ST indicates that the TOE generates asymmetric keys in accordance with RSA schemes using key sizes of 4096 bit. In addition, ECC schemes are used with P-256, P-384 and P-521.

The TOE can create a RSA public-private key pair that can be used to generate a Certificate Signing Request (CSR). Via offline CSR transfer the TOE can provide its CSR for a Certificate Authority (CA) to generate a certificate, and the TOE's signed certificate can be imported. RSA, ECC and FFC schemes are used for TLS communications with syslog, LDAP and between distributed TOE components. RSA, ECC, and FFC are also used for HTTPS remote Administration. The TOE implements FFC key establishment schemes in TLS. ECC is used for ECDH key exchange for TLS.

Component Guidance Assurance Activities: The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used. This includes limiting the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target.

Component Testing Assurance Activities: Note: The following tests require the developer to provide access to a test platform that provides the evaluator with tools that are typically not found on factory products.

Generation of long-term cryptographic keys (i.e. keys that are not ephemeral keys/session keys) might be performed automatically (e.g. during initial start-up). Testing of key generation must cover not only administrator invoked key generation but also automated key generation (if supported).

Key Generation for FIPS PUB 186-4 RSA Schemes

The evaluator shall verify the implementation of RSA Key Generation by the TOE using the Key Generation test. This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d.

Key Pair generation specifies 5 ways (or methods) to generate the primes p and q. These include:

a) Random Primes:

- Provable primes

- Probable primes

b) Primes with Conditions:

- Primes p1, p2, q1, q2, p and q shall all be provable primes

- Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes

- Primes p1, p2, q1, q2, p and q shall all be probable primes

To test the key generation method for the Random Provable primes method and for all the Primes with Conditions methods, the evaluator must seed the TSF key generation routine with sufficient data to deterministically generate the RSA key pair. This includes the random seed(s), the public exponent of the RSA key, and the desired key length. For each key length supported, the evaluator shall have the TSF generate 25 key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation.

Key Generation for Elliptic Curve Cryptography (ECC)

FIPS 186-4 ECC Key Generation Test

For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall require the implementation under test (IUT) to generate 10 private/public key pairs. The private key shall be generated using an approved random bit generator (RBG). To determine correctness, the evaluator shall submit the generated key pairs to the public key verification (PKV) function of a known good implementation.

FIPS 186-4 Public Key Verification (PKV) Test

For each supported NIST curve, i.e., P-256, P-384 and P-521, the evaluator shall generate 10 private/public key pairs using the key generation function of a known good implementation and modify five of the public key values so that they are incorrect, leaving five values unchanged (i.e., correct). The evaluator shall obtain in response a set of 10 PASS/FAIL values.

Key Generation for Finite-Field Cryptography (FFC)

The evaluator shall verify the implementation of the Parameters Generation and the Key Generation for FFC by the TOE using the Parameter Generation and Key Generation test. This test verifies the ability of the TSF to correctly produce values for the field prime p, the cryptographic prime q (dividing p-1), the cryptographic group generator g, and the calculation of the private key x and public key y.

The Parameter generation specifies 2 ways (or methods) to generate the cryptographic prime q and the field prime p:

- Primes q and p shall both be provable primes

- Primes q and field prime p shall both be probable primes

and two ways to generate the cryptographic group generator g:

- Generator g constructed through a verifiable process

- Generator g constructed through an unverifiable process.

The Key generation specifies 2 ways to generate the private key x:

- len(q) bit output of RBG where 1 <=x <= q-1

- len(q) + 64 bit output of RBG, followed by a mod q-1 operation and a +1 operation, where 1<= x<=q-1.

The security strength of the RBG must be at least that of the security offered by the FFC parameter set.

To test the cryptographic and field prime generation method for the provable primes method and/or the group generator g for a verifiable process, the evaluator must seed the TSF parameter generation routine with sufficient data to deterministically generate the parameter set.

For each key length supported, the evaluator shall have the TSF generate 25 parameter sets and key pairs. The evaluator shall verify the correctness of the TSF's implementation by comparing values generated by the TSF with those generated from a known good implementation. Verification must also confirm

- g != 0,1

- q divides p-1

- g^q mod p = 1

- g^x mod p = y

for each FFC parameter set and key pair.

FFC Schemes using 'safe-prime' groups

Testing for FFC Schemes using safe-prime groups is done as part of testing in CKM.2.1.

(TD0580 applied)

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.1.2.

2.3.2 Cryptographic Key Establishment (NDcPP22e:FCS_CKM.2)

2.3.2.1 NDcPP22e:FCS_CKM.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall ensure that the supported key establishment schemes correspond to the key generation schemes identified in FCS_CKM.1.1. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme. It is sufficient to provide the scheme, SFR, and service in the TSS.

The intent of this activity is to be able to identify the scheme being used by each service. This would mean, for example, one way to document scheme usage could be:

```
Scheme          |          SFR          |        Service

-------------------------------------------------------------------------------------------

RSA             |  FCS_TLSS_EXT.1  | Administration

-------------------------------------------------------------------------------------------

ECDH            |  FCS_SSHC_EXT.1  | Audit Server

-------------------------------------------------------------------------------------------

ECDH            |  FCS_IPSEC_EXT.1 | Authentication Server

-------------------------------------------------------------------------------------------
```

The information provided in the example above does not necessarily have to be included as a table but can be presented in other ways as long as the necessary data is available.

(TD0580 applied)

Section 6.1, Table 18 (FCS_CKM.1) in the ST indicates that the TOE generates asymmetric keys in accordance with RSA schemes using key sizes of 4096 bit. In addition, ECC schemes are used with P-256, P-384 and P-521.

The TOE can create a RSA public-private key pair that can be used to generate a Certificate Signing Request (CSR). Via offline CSR transfer the TOE can provide its CSR for a Certificate Authority (CA) to generate a certificate, and the TOE's signed certificate can be imported. RSA, ECC and FFC schemes are used for TLS communications with syslog, LDAP and between distributed TOE components. RSA, ECC, and FFC are also used for HTTPS remote Administration. The TOE implements FFC key establishment schemes in TLS. ECC is used for ECDH key exchange for TLS.

The claims for FCS_CKM.1 and FCS_CKM.2 match as expected.

Component Guidance Assurance Activities: The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected key establishment scheme(s).

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used. This includes limiting the selected key establishment schemes as defined in the Security Target.

Component Testing Assurance Activities: Key Establishment Schemes

The evaluator shall verify the implementation of the key establishment schemes of the supported by the TOE using the applicable tests below.

SP800-56A Key Establishment Schemes

The evaluator shall verify a TOE's implementation of SP800-56A key agreement schemes using the following Function and Validity tests. These validation tests for each key agreement scheme verify that a TOE has implemented the components of the key agreement scheme according to the specifications in the Recommendation. These components include the calculation of the DLC primitives (the shared secret value Z) and the calculation of the derived keying material (DKM) via the Key Derivation Function (KDF). If key confirmation is supported, the evaluator shall also verify that the components of key confirmation have been implemented correctly, using the test procedures described below. This includes the parsing of the DKM, the generation of MACdata and the calculation of MACtag.

Function Test

The Function test verifies the ability of the TOE to implement the key agreement schemes correctly. To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each supported key agreement scheme-key agreement role combination, KDF type, and, if supported, key confirmation role- key confirmation type combination, the tester shall generate 10 sets of test vectors. The data set consists of one set of domain parameter values (FFC) or the NIST approved curve (ECC) per 10 sets of public keys. These keys are static, ephemeral or both depending on the scheme being tested.

The evaluator shall obtain the DKM, the corresponding TOE's public keys (static and/or ephemeral), the MAC tag(s), and any inputs used in the KDF, such as the Other Information field OI and TOE id fields.

If the TOE does not use a KDF defined in SP 800-56A, the evaluator shall obtain only the public keys and the hashed value of the shared secret.

The evaluator shall verify the correctness of the TSF's implementation of a given scheme by using a known good implementation to calculate the shared secret value, derive the keying material DKM, and compare hashes or MAC tags generated from these values.

If key confirmation is supported, the TSF shall perform the above for each implemented approved MAC algorithm.

Validity Test

The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOE's public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE id fields.

The evaluator shall inject an error in some of the test vectors to test that the TOE recognizes invalid key agreement results caused by the following fields being incorrect: the shared secret value Z, the DKM, the other information field OI, the data to be MACed, or the generated MACTag. If the TOE contains the full or partial (only ECC) public key validation, the evaluator will also individually inject errors in both parties' static public keys, both parties' ephemeral public keys and the TOE's static private key to assure the TOE detects errors in the public key validation function and/or the partial key validation function (in ECC only). At least two of the test vectors shall remain unmodified and therefore should result in valid key agreement results (they should pass).

The TOE shall use these modified test vectors to emulate the key agreement scheme using the corresponding parameters. The evaluator shall compare the TOE's results with the results using a known good implementation verifying that the TOE detects these errors.

RSA-based key establishment

The evaluator shall verify the correctness of the TSF's implementation of RSAES-PKCS1-v1_5 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses RSAES-PKCS1-v1_5.

FFC Schemes using 'safe-prime' groups

The evaluator shall verify the correctness of the TSF's implementation of safe-prime groups by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and FPT_ITT.1 that uses safe-prime groups. This test must be performed for each safe-prime group that each protocol uses.

(TD0580 applied)

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.1.2.

2.3.3 Cryptographic Key Destruction (NDcPP22e:FCS_CKM.4)

2.3.3.1 NDcPP22e:FCS_CKM.4.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator examines the TSS to ensure it lists all relevant keys (describing the origin and storage location of each), all relevant key destruction situations (e.g. factory reset or device wipe function, disconnection of trusted channels, key change as part of a secure channel protocol), and the destruction method used in each case. For the purpose of this Evaluation Activity the relevant keys are those keys that are relied upon to support any of the SFRs in the Security Target. The evaluator confirms that the description of keys and storage locations is consistent with the functions carried out by the TOE (e.g. that all keys for the TOE-specific secure channels and protocols, or that support FPT_APW.EXT.1 and FPT_SKP_EXT.1, are accounted for2). In particular, if a TOE claims not to store plaintext keys in non-volatile memory then the evaluator checks that this is consistent with the operation of the TOE.

The evaluator shall check to ensure the TSS identifies how the TOE destroys keys stored as plaintext in non-volatile memory, and that the description includes identification and description of the interfaces that the TOE uses to destroy keys (e.g., file system APIs, key store APIs).

Note that where selections involve 'destruction of reference' (for volatile memory) or 'invocation of an interface' (for non-volatile memory) then the relevant interface definition is examined by the evaluator to ensure that the interface supports the selection(s) and description in the TSS. In the case of non-volatile memory the evaluator includes in their examination the relevant interface description for each media type on which plaintext keys are stored. The presence of OS-level and storage device-level swap and cache files is not examined in the current version of the Evaluation Activity.

Where the TSS identifies keys that are stored in a non-plaintext form, the evaluator shall check that the TSS identifies the encryption method and the key-encrypting-key used, and that the key-encrypting-key is either itself stored in an encrypted form or that it is destroyed by a method included under FCS_CKM.4.

The evaluator shall check that the TSS identifies any configurations or circumstances that may not conform to the key destruction requirement (see further discussion in the Guidance Documentation section below). Note that reference may be made to the Guidance Documentation for description of the detail of such cases where destruction may be prevented or delayed.

Where the ST specifies the use of 'a value that does not contain any CSP' to overwrite keys, the evaluator examines the TSS to ensure that it describes how that pattern is obtained and used, and that this justifies the claim that the pattern does not contain any CSPs.

Section 6.1, Table 18 (FCS_CKM.4) in the ST states that the TOE meets all requirements specified in FIPS 140-2 for destruction of keys and Critical Security Parameters (CSPs). Section 7.1, Table 18 describes the key zeroization as identified in FCS_CKM.4 and provides a list of all of the relevant keys, how they are generated, how they are zeroized and when this occurs.

Component Guidance Assurance Activities: A TOE may be subject to situations that could prevent or delay key destruction in some cases. The evaluator shall check that the guidance documentation identifies configurations or circumstances that may not strictly conform to the key destruction requirement, and that this description is consistent with the relevant parts of the TSS (and any other supporting information used). The evaluator shall check that the guidance documentation provides guidance on situations where key destruction may be delayed at the physical layer.

For example, when the TOE does not have full access to the physical memory, it is possible that the storage may be implementing wear-levelling and garbage collection. This may result in additional copies of the key that are logically inaccessible but persist physically. Where available, the TOE might then describe use of the TRIM command [Where TRIM is used then the TSS and/or guidance documentation is also expected to describe how the keys are stored such that they are not inaccessible to TRIM, (e.g. they would need not to be contained in a file less than 982 bytes which would be completely contained in the master file table)] and garbage collection to destroy these persistent copies upon their deletion (this would be explained in TSS and Operational Guidance).

The ST and the **Admin Guide** do not identify any circumstances or configurations that do not strictly conform to the key destruction requirements.

Component Testing Assurance Activities: None Defined

2.3.4 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP22e:FCS_COP.1/DataEncryption)

2.3.4.1 NDcPP22e:FCS_COP.1.1/DataEncryption

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to ensure it identifies the key size(s) and mode(s) supported by the TOE for data encryption/decryption.

Section 6.1, Table 18 (FCS_COP.1/DataEncryption) in the ST states that the TOE provides symmetric encryption and decryption capabilities using AES in CBC and GCM mode (128 bits, 256 bits) as described in ISO 18033-3 and ISO

10116. AES is implemented in TLS. Through the implementation of the FIPS validated cryptographic module, the TOE provides AES encryption and decryption in support of TLS for secure communications.

Component Guidance Assurance Activities: The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected mode(s) and key size(s) defined in the Security Target supported by the TOE for data encryption/decryption.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used. This includes limiting the selected key generation scheme(s) and key size(s) for all cryptographic protocols defined in the Security Target.

Component Testing Assurance Activities: AES-CBC Known Answer Tests

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

KAT-1. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key.

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.

KAT-2. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys.

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.

KAT-3. To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N].

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

KAT-4. To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.

AES-CBC Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.

The evaluator shall also test the decrypt functionality for each mode by decrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.

AES-CBC Monte Carlo Tests

The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

```
# Input: PT, IV, Key

for i = 1 to 1000:

if i == 1:

CT[1] = AES-CBC-Encrypt(Key, IV, PT)

PT = IV

else:

CT[i] = AES-CBC-Encrypt(Key, PT)

PT = CT[i-1]
```

The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AES-CBC-Decrypt.

AES-GCM Test

The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:

128 bit and 256 bit keys

a) Two plaintext lengths. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.

a) Three AAD lengths. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.

b) Two IV lengths. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.

The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.

The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.

The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

AES-CTR Known Answer Tests

The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. Due to the fact that Counter Mode does not specify the counter that is used, it is not possible to implement an automated test for this mode. The generation and management of the counter is tested through FCS_SSH*_EXT.1.4. If CBC and/or GCM are selected in FCS_COP.1/DataEncryption, the test activities for those modes sufficiently demonstrate the correctness of the AES algorithm. If CTR is the only selection in FCS_COP.1/DataEncryption, the AES-CBC Known Answer Test, AES-GCM Known Answer Test, or the following test shall be performed (all of these tests demonstrate the correctness of the AES algorithm):

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

KAT-1 To test the encrypt functionality, the evaluator shall supply a set of 5 plaintext values for each selected keysize and obtain the ciphertext value that results from encryption of the given plaintext using a key value of all zeros.

KAT-2 To test the encrypt functionality, the evaluator shall supply a set of 5 key values for each selected keysize and obtain the ciphertext value that results from encryption of an all zeros plaintext using the given key value.

KAT-3 To test the encrypt functionality, the evaluator shall supply a set of key values for each selected keysize as described below and obtain the ciphertext values that result from AES encryption of an all zeros plaintext using the given key values. A set of 128 128-bit keys, a set of 192 192-bit keys, and/or a set of 256 256-bit keys. Key_i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1, N].

KAT-4 To test the encrypt functionality, the evaluator shall supply the set of 128 plaintext values described below and obtain the ciphertext values that result from encryption of the given plaintext using each selected keysize with a key value of all zeros (e.g. 256 ciphertext values will be generated if 128 bits and 256 bits are selected and 384 ciphertext values will be generated if all keysizes are selected). Plaintext value i in each set shall have the leftmost bits be ones and the rightmost 128-i bits be zeros, for i in [1, 128].

AES-CTR Multi-Block Message Test

The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 less-than i less-than-or-equal to 10 (test shall be performed using AES-ECB mode). For each i the evaluator shall choose a key and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key using a known good implementation. The evaluator shall perform this test using each selected keysize.

AES-CTR Monte-Carlo Test

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

# Input: PT, Key

for i = 1 to 1000:

CT[i] = AES-ECB-Encrypt(Key, PT) PT = CT[i]

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

There is no need to test the decryption engine.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.1.2.

2.3.5 Cryptographic Operation (Hash Algorithm) (NDcPP22e:FCS_COP.1/Hash)

2.3.5.1 NDcPP22e:FCS_COP.1.1/Hash

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS.

Section 6.1, Table 18 (FCS_COP.1/Hash) in the ST states that the TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384 and SHA-512 as specified in ISO/IEC 10118-3:2004. Through the implementation of the FIPS validated cryptographic module, the TOE provides Secure Hash Standard (SHS) hashing in support of TLS, for secure communications. Management of the cryptographic algorithms is provided through the Web UI with auditing of those commands.

Section 6.1, Table 18 (FPT_TUD_EXT.1) in the ST indicates that when an administrator initiates an update process from SMC, each appliance will use digital signature verification to ensure the integrity of the software update prior to installation of the update.

Component Guidance Assurance Activities: The evaluator checks the AGD documents to determine that any configuration that is required to configure the required hash sizes is present.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used. This includes limiting the required hash sizes.

Component Testing Assurance Activities: The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented testmacs.

The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP.

Short Messages Test - Bit-oriented Mode

The evaluators devise an input set consisting of m+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m bits. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Short Messages Test - Byte-oriented Mode

The evaluators devise an input set consisting of m/8+1 messages, where m is the block length of the hash algorithm. The length of the messages range sequentially from 0 to m/8 bytes, with each message being an integral number of bytes. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test - Bit-oriented Mode

The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 99*i, where 1 <= i <= m. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test - Byte-oriented Mode

The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm (e.g. 512 bits for SHA-256). The length of the ith message is m + 8*99*i, where 1 <= i <= m/8. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Pseudorandomly Generated Messages Test

This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is n bits long, where n is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.1.2.

2.3.6 Cryptographic Operation (Keyed Hash Algorithm) (NDcPP22e:FCS_COP.1/KeyedHash)

2.3.6.1 NDcPP22e:FCS_COP.1.1/KeyedHash

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key length, hash function used, block size, and output MAC length used.

Section 6.1, Table 18 (FCS_COP.1/KeyedHash) in the ST states that the TOE provides keyed-hashing message authentication services using HMAC-SHA1, HMAC-SHA-256, and HMAC-SHA-384, with key sizes 160, 256 and 384-bits, and message digest sizes 160, 256, and 384-bits as specified in ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”.

Through the implementation of the FIPS validated cryptographic module, the TOE provides SHS hashing and HMAC message authentication in support of TLS for secure communications. Management of the cryptographic algorithms is provided through the GUI with auditing of those commands. SHS hashing and HMAC message authentication (SHA-1) is used in the establishment of TLS/HTTPS sessions.

Component Guidance Assurance Activities: The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the values used by the HMAC function: key length, hash function used, block size, and output MAC length used defined in the Security Target supported by the TOE for keyed hash function.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used. This includes limiting the selected key generation scheme(s) and key size(s), and limiting the required hash sizes for all cryptographic protocols defined in the Security Target.

Component Testing Assurance Activities: For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and message data using a known good implementation.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.1.2.

2.3.7 Cryptographic Operation (Signature Generation and Verification) (NDcPP22e:FCS_COP.1/SigGen)

2.3.7.1 NDcPP22e:FCS_COP.1.1/SigGen

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that it specifies the cryptographic algorithm and key size supported by the TOE for signature services.

Section 6.1, Table 18 (FCS_COP.1/SigGen) in the ST states that the TOE provides cryptographic signature services using RSA Digital Signature Algorithm with key sizes of 2048 and 4096 as specified in FIPS PUB 186-4, "Digital Signature Standard".

Through the implementation of the FIPS validated cryptographic module, the TOE provides cryptographic signatures in support of TLS for secure communications. Management of the cryptographic algorithms is provided through the GUI with auditing of those commands. The TOE provides the RSA option in support of TLS key establishment.

Component Guidance Assurance Activities: The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected cryptographic algorithm and key size defined in the Security Target supported by the TOE for signature services.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used. This includes limiting the selected key generation scheme(s) and key size(s), and limiting the required hash sizes for all cryptographic protocols defined in the Security Target.

Component Testing Assurance Activities: ECDSA Algorithm Tests

ECDSA FIPS 186-4 Signature Generation Test

For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate 10 1024-bit long messages and obtain for each message a public key and the resulting signature values R and S. To determine correctness, the evaluator shall use the signature verification function of a known good implementation.

ECDSA FIPS 186-4 Signature Verification Test

For each supported NIST curve (i.e., P-256, P-384 and P-521) and SHA function pair, the evaluator shall generate a set of 10 1024-bit message, public key and signature tuples and modify one of the values (message, public key or signature) in five of the 10 tuples. The evaluator shall obtain in response a set of 10 PASS/FAIL values.

RSA Signature Algorithm Tests

Signature Generation Test

The evaluator generates or obtains 10 messages for each modulus size/SHA combination supported by the TOE. The TOE generates and returns the corresponding signatures.

The evaluator shall verify the correctness of the TOE's signature using a trusted reference implementation of the signature verification algorithm and the associated public keys to verify the signatures.

Signature Verification Test

For each modulus size/hash algorithm selected, the evaluator generates a modulus and three associated key pairs, (d, e). Each private key d is used to sign six pseudorandom messages each of 1024 bits using a trusted reference implementation of the signature generation algorithm. Some of the public keys, e, messages, or signatures are altered so that signature verification should fail. For both the set of original messages and the set of altered messages: the modulus, hash algorithm, public key e values, messages, and signatures are forwarded to the TOE, which then attempts to verify the signatures and returns the verification results.

The evaluator verifies that the TOE confirms correct signatures on the original messages and detects the errors introduced in the altered messages.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.1.2.

2.3.8 HTTPS Protocol (NDcPP22e:FCS_HTTPS_EXT.1)

2.3.8.1 NDcPP22e:FCS_HTTPS_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.3.8.2 NDcPP22e:FCS_HTTPS_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.3.8.3 NDcPP22e:FCS_HTTPS_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS and determine that enough detail is provided to explain how the implementation complies with RFC 2818.

Section 6.1, Table 18 (FCS_HTTPS_EXT.1) in the ST states that the TOE (SMC only) implements HTTPS over TLS as specified in RFC 2818 and FCS_TLSS_EXT.1. The TSF HTTPS implementation authenticates the TOE to the remote client with an X.509 certificate. System Administrators manage the TOE identity certificates using the TOE GUI.

The TSF HTTPS implementation performs server-based authentication using a server X.509v3 certificate to establish the TLS session. The TSF HTTPS implementation does not require client authentication at the TLS level but presents the Web interface logon page for administrative users to authenticate using their name and password.

Component Guidance Assurance Activities: The evaluator shall examine the guidance documentation to verify it instructs the Administrator how to configure TOE for use as an HTTPS client or HTTPS server.

Section *Secure Installation* in the **Admin Guide** provides general guidance for the initial configuration of the appliances, including how to configure the IP address for access. Section *Certificates* of the **Admin Guide** describes how to replace the identity certificates of the TOE which are used for HTTPs.

Component Testing Assurance Activities: This test is now performed as part of FIA_X509_EXT.1/Rev testing.

Tests are performed in conjunction with the TLS evaluation activities.

If the TOE is an HTTPS client or an HTTPS server utilizing X.509 client authentication, then the certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.

Administration via HTTPS was successfully demonstrated in NDcPP22e:FTP_TRP.1/Admin and the TLS protocol was tested in NDcPP22e:FCS_TLSS_EXT.1.

2.3.9 NTP Protocol (NDcPP22e:FCS_NTP_EXT.1)

2.3.9.1 NDcPP22e:FCS_NTP_EXT.1.1

TSS Assurance Activities: The evaluator shall examine the TSS to ensure it identifies the version of NTP supported, how it is implemented and what approach the TOE uses to ensure the timestamp it receives from an NTP timeserver (or NTP peer) is from an authenticated source and the integrity of the time has been maintained. The TOE must support at least one of the methods or may use multiple methods, as specified in the SFR element 1.2. The evaluator shall ensure that each method selected in the ST is described in the TSS, including the version of NTP supported in element 1.1, the message digest algorithms used to verify the authenticity of the timestamp and/or the protocols used to ensure integrity of the timestamp.

Section 6.1, Table 18 (FCS_NTP_EXT.1) in the ST states that NTP functionality can be enabled using the Central Management (WebUI) on SMC to configure each appliance (via Appliance Manager > Appliance Configuration > Network Services > NTP Server). Each appliance can be configured to use up to three NTP servers (auto-negotiating use of NTPv4), and authenticating each server with a unique Key ID and Key Value (SHA1). The TOE will not use any NTP updates from broadcast or multicast addresses.

Guidance Assurance Activities: The evaluator shall examine the guidance documentation to ensure it provides the Security Administrator instructions as how to configure the version of NTP supported, how to configure multiple NTP servers for the TOE's time source and how to configure the TOE to use the method(s) that are selected in the ST.

Section *NTP* of the **Admin Guide** states during initial setup, when using the Appliance Setup Tool to join appliances to SMC, each appliance was configured with at least one NTP server. To satisfy requirements for the CC-evaluated configuration, each NTP server must be configured to use authentication. Instructions are provided to configure NTPv authentication using SHA1 which can include more than 1 NTP server.

Testing Assurance Activities: The version of NTP selected in element 1.1 and specified in the ST shall be verified by observing establishment of a connection to an external NTP server known to be using the specified version(s) of NTP. This may be combined with tests of other aspects of FCS_NTP_EXT.1 as described below.

The evaluator configured the TOE to get NTP time updates from the evaluator's NTP Server. Inspection of the network traffic between the TOE and the evaluator's NTP server revealed that NTPv4 is supported by the TOE.

2.3.9.2 NDcPP22e:FCS_NTP_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: For each of the secondary selections made in the ST, the evaluator shall examine the guidance document to ensure it instructs the Security Administrator how to configure the TOE to use the algorithms that support the authenticity of the timestamp and/or how to configure the TOE to use the protocols that ensure the integrity of the timestamp.

Assurance Activity Note:

Each primary selection in the SFR contains selections that specify a cryptographic algorithm or cryptographic protocol. For each of these secondary selections made in the ST, the evaluator shall examine the guidance documentation to ensure that the documentation instructs the administrator how to configure the TOE to use the chosen option(s).

Section *NTP* of the **Admin Guide** states during initial setup, when using the Appliance Setup Tool to join appliances to SMC, each appliance was configured with at least one NTP server. To satisfy requirements for the CC-evaluated configuration, each NTP server must be configured to use authentication. Instructions are provided to configure NTPv authentication using SHA1 which can include more than 1 NTP server.

Testing Assurance Activities: The cryptographic algorithms selected in element 1.2 and specified in the ST will have been specified in an FCS_COP SFR and tested in the accompanying Evaluation Activity for that SFR. Likewise, the cryptographic protocol selected in in element 1.2 and specified in the ST will have been specified in an FCS SFR and tested in the accompanying Evaluation Activity for that SFR.

[Conditional] If the message digest algorithm is claimed in element 1.2, the evaluator will change the message digest algorithm used by the NTP server in such a way that the new value does not match the configuration on the TOE and confirms that the TOE does not synchronize to this time source.

The evaluator shall use a packet sniffer to capture the network traffic between the TOE and the NTP server. The evaluator uses the captured network traffic, to verify the NTP version, to observe time change of the TOE and uses the TOE's audit log to determine that the TOE accepted the NTP server's timestamp update.

The captured traffic is also used to verify that the appropriate message digest algorithm was used to authenticate the time source and/or the appropriate protocol was used to ensure integrity of the timestamp that was transmitted in the NTP packets.

The evaluator configured the TOE to get NTP time updates from the evaluator's NTP Server using SHA1 authentication, set the time on the NTP server ahead and noted that the TOE successfully synched its time with that of the NTP server.

Next, the evaluator maintained the NTP configuration on the TOE, but modified the NTP server's keys so that the TOE was no longer using the correct value. The time on the NTP server was set ahead once again. This time, the TOE was no longer synching its time with that of the NTP server.

After the key was set back to its original value, the TOE being able to authenticate using the correct key, successfully was able to synch its time with that of the NTP server.

2.3.9.3 NDcPP22e:FCS_NTP_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: The evaluator shall examine the guidance documentation to ensure it provides the Security Administrator instructions as how to configure the TOE to not accept broadcast and multicast NTP packets that would result in the timestamp being updated.

Section *NTP* of the **Admin Guide** states by default, the NTP client in SNA appliances will not accept broadcast or multicast NTP packets to update the clock. This default setting satisfies the CC requirement and is not configurable.

Testing Assurance Activities: The evaluator shall configure NTP server(s) to support periodic time updates to broadcast and multicast addresses. The evaluator shall confirm the TOE is configured to not accept broadcast and multicast NTP packets that would result in the timestamp being updated. The evaluator shall check that the time stamp is not updated after receipt of the broadcast and multicast packets.

The evaluator configured the TOE to get NTP time updates from the evaluator's NTP Server. The evaluator also alternately configured the NTP server to send broadcast and multicast time updates such that they would be visible to the TOE. The evaluator observed that the TOE did not accept the time updates from the NTP Server.

2.3.9.4 NDcPP22e:FCS_NTP_EXT.1.4

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: Test 1: The evaluator shall confirm the TOE supports configuration of at least three (3) NTP time sources. The evaluator shall configure at least three NTP servers to support periodic time updates to the TOE. The evaluator shall confirm the TOE is configured to accept NTP packets that would result in the timestamp being updated from each of the NTP servers. The evaluator shall check that the time stamp is updated after receipt of the NTP packets. The purpose of this test to verify that the TOE can be configured to synchronize with multiple NTP servers. It is up to the evaluator to determine that the multi- source update of the time information is appropriate and consistent with the behaviour prescribed by the RFC 1305 for NTPv3 and RFC 5905 for NTPv4.

Test 2: (The intent of this test is to ensure that the TOE would only accept NTP updates from configured NTP Servers).

The evaluator shall confirm that the TOE would not synchronize to other, not explicitly configured time sources by sending an otherwise valid but unsolicited NTP Server responses indicating different time from the TOE's current system time. This rogue time source needs to be configured in a way (e.g. degrade or disable valid and configured NTP servers) that could plausibly result in unsolicited updates becoming a preferred time source if they are not discarded by the TOE. The TOE is not mandated to respond in a detectable way or audit the occurrence of such unsolicited updates. The intent of this test is to ensure that the TOE would only accept NTP updates from configured NTP Servers. It is up to the evaluator to craft and transmit unsolicited updates in a way that would be consistent with the behaviour of a correctly-functioning NTP server.

(TD0528 applied)

Test 1 - The evaluator configured 3 NTP time sources on the TOE and then verified that the TOE was able to sync its time with at least one of the three configured sources.

Test 2 - The evaluator configured an NTP server to broadcast and advertise its time to the IP address of the TOE and confirmed that the TOE only synchronizes its time with configured NTP servers and does not respond to the broadcast packets sent by the NTP server.

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.3.10 Random Bit Generation (NDcPP22e:FCS_RBG_EXT.1)

2.3.10.1 NDcPP22e:FCS_RBG_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.3.10.2 NDcPP22e:FCS_RBG_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: Documentation shall be produced - and the evaluator shall perform the activities - in accordance with Appendix D of [NDcPP].

The evaluator shall examine the TSS to determine that it specifies the DRBG type, identifies the entropy source(s) seeding the DRBG, and state the assumed or calculated min-entropy supplied either separately by each source or the min-entropy contained in the combined seed value.

The Entropy description is provided in a separate (non-ST) document that has been delivered to NIAP for approval. Note that the entropy analysis has been accepted by NIAP/NSA.

Section 6.1, Table 18 (FCS_RBG_EXT.1) in the ST states that the TOE implements a NIST-approved AES-CTR Deterministic Random Bit Generator (DRBG) seeded by a hardware-based entropy source within the TOE. The DRBG is seeded with a minimum of 256 bits of full entropy, which is at least equal to the greatest security strength of the keys and hashes that the DRBG will generate.

Component Guidance Assurance Activities: Documentation shall be produced - and the evaluator shall perform the activities - in accordance with Appendix D of [NDcPP].

The evaluator shall confirm that the guidance documentation contains appropriate instructions for configuring the RNG functionality.

The Entropy description is provided in a separate (non-ST) document that has been delivered to NIAP for approval. Note that the entropy analysis has been accepted by NIAP/NSA.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used. This includes the switch to use the proper DRBG methods.

Component Testing Assurance Activities: The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration.

If the RNG has prediction resistance enabled, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. 'generate one block of random bits' means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP800-90A).

If the RNG does not have prediction resistance, each trial consists of (1) instantiate DRBG, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 - 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

Entropy input: the length of the entropy input value must equal the seed length.

Nonce: If a nonce is supported (CTR_DRBG with no Derivation Function does not use a nonce), the nonce bit length is one-half the seed length.

Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.

Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

The TOE has been CAVP tested. Refer to the CAVP certificates identified in Section 1.1.2.

2.3.11 TLS Client Protocol Without Mutual Authentication - per TD0634 (NDcPP22e:FCS_TLSC_EXT.1)

2.3.11.1 NDcPP22e:FCS_TLSC_EXT.1.1

TSS Assurance Activities: The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified include those listed for this component.

Section 6.1, Table 18 (FCS_TLSC_EXT.1) in the ST states that the TOE only supports TLSv1.2 with AES 128 or 256 bit symmetric ciphers in CBC and GCM modes, in conjunction with SHA (SHA1, SHA256, and SHA384), RSA, DHE (Diffie-Hellman Group 14), ECDHE (NIST curves supported are secp256r1, secp384r1, and secp521r1) and ECDSA. By default all the Supported Elliptic Curves Extension options are presented in the Client Hello and this behavior is not configurable.

The following TLS cipher suites are implemented by the TOE in CC mode:

- Ciphersuites relevant to FTP_ITC and FCS_TLSC_EXT.2, for syslog over TLS (client only, from each TOE component) are listed in section 5.3.3.11 of the ST.

- Ciphersuites relevant to FPT_ITT, FCS_TLSC_EXT.2, and FCS_TLSS_EXT.2 (client and server, where each TOE component can operate as the client and/or server) are as listed in sections 5.3.3.11 and 5.3.3.13 of the ST.

- Ciphersuites relevant to FTP_ITC and FCS_TLSC_EXT.1 for LDAP over TLS (client only, from the SMC TOE component only) are listed in section 5.3.3.11 of the ST.

While the FOM (see section 7.2) supports additional cipher suites (for example, RSA_3DES_EDE_CBC_SHA, RSA_DES_CBC_SHA, RSA_RC4_128_MD5, RSA_RC4_128_SHA, etc.), they are all disabled while operating in CC mode.

Guidance Assurance Activities: The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used for any TLS connection, including connections to LDAP, syslog, and Distributed TOE connections. It further identifies the specific TLS ciphersuites used when the TOE is operating as a TLS client and indicates that only TLSv1.2 is supported. For all TLS functionality, RSA key establishment uses 4096 bits. The only enabled EC curves are secp256r1, secp384r1, and secp521r1; and the Diffie-Hellman parameters are 2048 bits.

Section *Review External Service Certificate Requirements* of the **Admin Guide** describes the requirements for certificates of external services such as a remote syslog and LDAP. Section *Add External Service Certificates to the Trust Stores* describes how to add the relevant certificates for external services, including a remote syslog and LDAP, to the appliances trust store.

Section *Certificates* of the **Admin Guide** defines the best practices and procedures for updating the certificates of the TOE appliances to be used for the TLS distributed TOE channel.

Section *LDAP* of the **Admin Guide** describes the configuration process for enabling LDAP authentication over TLS on the SMC.

Testing Assurance Activities: Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

Test 2: The evaluator shall attempt to establish the connection using a server with a server certificate that contains the Server Authentication purpose in the extendedKeyUsage extension and verify that a connection is established. The evaluator will then verify that the client rejects an otherwise valid server certificate that lacks the Server Authentication purpose in the extendedKeyUsage field, and a connection is not established. Ideally, the two certificates should be identical except for the extendedKeyUsage field.

Test 3: The evaluator shall send a server certificate in the TLS connection that does not match the server-selected ciphersuite (for example, send an ECDSA certificate while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite). The evaluator shall verify that the TOE disconnects after receiving the server's Certificate handshake message.

Test 4: The evaluator shall perform the following 'negative tests':

a) The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.

b) Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.

c) [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension the evaluator shall configure the server to perform an ECDHE or DHE key exchange in the TLS connection using a non-supported curve/group (for example P-192) and shall verify that the TOE disconnects after receiving the server's Key Exchange handshake message.

Test 5: The evaluator shall perform the following modifications to the traffic:

a) Change the TLS version selected by the server in the Server Hello to a non-supported TLS version and verify that the client rejects the connection.

b) [conditional]: If using DHE or ECDH, modify the signature block in the Server's Key Exchange handshake message, and verify that the handshake does not finished successfully, and no application data flows. This test does not apply to cipher suites using RSA key exchange. If a TOE only supports RSA key exchange in conjunction with TLS, then this test shall be omitted.

Test 6: The evaluator performs the following 'scrambled message tests':

a) Modify a byte in the Server Finished handshake message and verify that the handshake does not finish successfully and no application data flows.

b) Send a garbled message from the server after the server has issued the ChangeCipherSpec message and verify that the handshake does not finish successfully and no application data flows.

c) Modify at least one byte in the server's nonce in the Server Hello handshake message and verify that the client rejects the Server Key Exchange handshake message (if using a DHE or ECDHE ciphersuite) or that the server denies the client's Finished handshake message.

These tests were repeated for 3 variations as follows: ITT Communication, Syslog, and LDAP. LDAP is only supported for the Manager and Virtual Manager devices.

Test 1- The evaluator configured a test server to accept each ciphersuite allowed by the PP, a single ciphersuite at a time. While the test server listened for the single configured ciphersuite, the evaluator caused the TOE to attempt a connection to the test Server. The evaluator confirmed that each claimed TLS ciphersuite resulted in a successful connection.

Test 2 - The evaluator configured the TOE to connect to a test server and attempted two connections. During the first TLS negotiation the test server sent a valid certificate chaining to a CA known by the TOE. The certificate included the Server Authentication extended key usage (EKU) field. The PCAP for this part of the test shows that the client issued the Change Cipher Spec and Encrypted Handshake messages, which signify that the TLS connection is successful.

During the second connection, the server presented a certificate chaining to a CA known by the TOE. However, the certificate did not include the Server Authentication extended key usage (EKU) field. In the PCAP for this part of the test, the client generates a fatal alert and closes the connection.

Test 3 - The evaluator configured the test server to require TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 and then used an RSA key in its key exchange. In each case, the TOE rejected the connection attempt.

Test 4 -

Test 4a - The evaluator configured the TOE to communicate with a test server that sends only a TLS_NULL_WITH_NULL_NULL ciphersuite in the server hello. The evaluator then attempted to establish a TLS session from the TOE to the test server and observed that the connection failed.

Test 4b - The evaluator configured the test server to select a ciphersuite not presented in the TOE's Client Hello handshake message. The evaluator attempted to establish a TLS session from the TOE to the test server and observed that the connection failed.

Test 4c - The evaluator configured the TOE to connect to a test server using TLS with a TOE supported ECDHE key exchange method. The evaluator also configured the GSS test server to accept that same ECHDE key exchange method, but to require a curve that was not supported by the TOE (i.e., P-192). The evaluator then attempted to establish a TLS session from the TOE to the test server and observed that the connection failed.

Test 5 -

Test 5a - The evaluator configured the TOE to connect to a test server using TLS. During the connection the evaluator caused the server to use a TLS version in the Server Hello that is a non-supported TLS version (version 1.4 represented by two bytes 0x0305). The evaluator verified that the TOE rejected the connection.

Test 5b - The evaluator configured the TOE to connect to a test server using TLS. During the connection the evaluator caused the server to modify the signature block in the Server's Key Exchange handshake message. The evaluator verified that the TOE rejected the connection.

Test 6 -

Test 6a - The evaluator attempted a connection to the TOE where the evaluator modified a byte in the Server Finished handshake message, and then verified that the client rejected the connection attempt after receiving the server Finished message and that no application data was exchanged.

Test 6b - The evaluator garbled a message between the TOE and its TLS peer. The modification occurred after the Server sent the ChangeCipherSpec message. The evaluator observed that the Client denies the connection. Due to the nature of the error, regardless of whether the TOE is the client or server, the client is always the first to recognize the error.

Test 6c - The evaluator configured the TOE to connect to a test server using TLS. During the connection the evaluator caused the server to modify one byte in the server's nonce in the Server Hello handshake message. The evaluator verified that the TOE rejected the connection.

2.3.11.2 NDcPP22e:FCS_TLSC_EXT.1.2

TSS Assurance Activities: The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the administrator/application-configured reference identifier, including which types of reference identifiers are supported (e.g. application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported.

Note that where a TLS channel is being used between components of a distributed TOE for FPT_ITT.1, the requirements to have the reference identifier established by the user are relaxed and the identifier may also be established through a 'Gatekeeper' discovery process. The TSS should describe the discovery process and highlight how the reference identifier is supplied to the 'joining' component. Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1 and the ST author selected attributes from RFC 5280, the evaluator shall ensure the TSS describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The evaluator shall ensure the TSS presents an argument how the attribute type, or combination of attribute types, uniquely identify the remote TOE component; and the evaluator shall verify the attribute type, or combination of attribute types, is sufficient to support unique identification of the maximum supported number of TOE components.

If IP addresses are supported in the CN as reference identifiers, the evaluator shall ensure that the TSS describes the TOE's conversion of the text representation of the IP address in the CN to a binary representation of the IP address in network byte order. The evaluator shall also ensure that the TSS describes whether canonical format (RFC5952 for IPv6, RFC 3986 for IPv4) is enforced.

Section 6.1, Table 18 (FCS_TLSC_EXT.1) in the ST states that when in CC mode and the TOE acts as a TLS client connecting to a syslog server (syslog-over-TLS) the TOE will verify the server Common Name (CN) and/or Subject Alternative Name (SAN) against the reference identity. For all LDAP-over-TLS connections and all TLS connections between TOE components the TOE will verify the reference identifier in accordance with RFC 6125 section 6.

When an IP address is used as the reference identifier (supported for LDAP-over-TLS and syslog-over-TLS) the TOE converts the text representations (dotted-decimal notation) of the locally-stored IP address and the IP addresses found in the CN to binary (hex-based) representations in network byte order, enforcing the canonical format per RFC 3986.

Guidance Assurance Activities: The evaluator shall ensure that the operational guidance describes all supported identifiers, explicitly states whether the TOE supports the SAN extension or not, and includes detailed instructions on how to configure the reference identifier(s) used to check the identity of peer(s). If the identifier scheme implemented by the TOE includes support for IP addresses, the evaluator shall ensure that the operational guidance provides a set of warnings and/or CA policy recommendations that would result in secure TOE use.

Where the secure channel is being used between components of a distributed TOE for FPT_ITT.1, the SFR selects attributes from RFC 5280, and FCO_CPC_EXT.1.2 selects 'no channel'; the evaluator shall verify the guidance provides instructions for establishing unique reference identifiers based on RFC5280 attributes.

Section *LDAP* of the **Admin Guide** describes the configuration process for enabling LDAP authentication over TLS on the SMC including defining the fully qualified domain name or IP address of the server which is used as the reference identifier.

Section *Review Appliance Identity Certificate Requirements* of the **Admin Guide** describes that the hostname and IP address of the appliances should be included in the Subject Alt Name of their certificates as they are used as reference identifiers.

Testing Assurance Activities: Note that the following tests are marked conditional and are applicable under the following conditions:

a) For TLS-based trusted channel communications according to FTP_ITC.1 where RFC 6125 is selected, tests 1-6 are applicable.

or

b) For TLS-based trusted path communications according to FTP_TRP where RFC 6125 is selected, tests 1-6 are applicable

or

c) For TLS-based trusted path communications according to FPT_ITT.1 where RFC 6125 is selected, tests 1-6 are applicable. Where RFC 5280 is selected, only test 7 is applicable.

Note that for some tests additional conditions apply.

IP addresses are binary values that must be converted to a textual representation when presented in the CN of a certificate. When testing IP addresses in the CN, the evaluator shall follow the following formatting rules:

- IPv4: The CN contains a single address that is represented a 32-bit numeric address (IPv4) is written in decimal as four numbers that range from 0-255 separated by periods as specified in RFC 3986.

- IPv6: The CN contains a single IPv6 address that is represented as eight colon separated groups of four lowercase hexadecimal digits, each group representing 16 bits as specified in RFC 4291. Note: Shortened addresses, suppressed zeros, and embedded IPv4 addresses are not tested.

The evaluator shall configure the reference identifier according to the AGD guidance and perform the following tests during a TLS connection:

a) Test 1 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the CN.

Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 1.

b) Test 2 [conditional]: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, URI). When testing IPv4 or IPv6 addresses, the evaluator shall modify a single decimal or hexadecimal digit in the SAN.

c) Test 3 [conditional]: If the TOE does not mandate the presence of the SAN extension, the evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each identifier type (e.g. IPv4, IPv6, FQDN) supported in the CN. If the TOE does mandate the presence of the SAN extension, this Test shall be omitted.

d) Test 4 [conditional]: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds. The evaluator shall repeat this test for each supported SAN type (e.g. IPv4, IPv6, FQDN, SRV).

e) Test 5 [conditional]: The evaluator shall perform the following wildcard tests with each supported type of reference identifier that includes a DNS name (i.e. CN-ID with DNS, DNS-ID, SRV-ID, URI-ID):

1) [conditional]: The evaluator shall present a server certificate containing a wildcard that is not in the left- most label of the presented identifier (e.g. foo.*.example.com) and verify that the connection fails.

2) [conditional]: The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g. *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g. foo.example.com) and verify that the connection succeeds if wildcards are supported or fails if wildcards are not supported. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g. example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g. bar.foo.example.com) and verify that the connection fails. (Remark: Support for wildcards was always intended to be optional. It is sufficient to state that the TOE does not support wildcards and observe rejected connection attempts to satisfy corresponding assurance activities.)

f) Objective: The objective of this test is to ensure the TOE is able to differentiate between IP address identifiers that are not allowed to contain wildcards and other types of identifiers that may contain wildcards.

Test 6 [conditional]: If IP address identifiers supported in the SAN or CN, the evaluator shall present a server certificate that contains a CN that matches the reference identifier, except one of the groups has been replaced with a wildcard asterisk (*) (e.g. CN=*.168.0.1 when connecting to 192.168.1.20, CN=2001:0DB8:0000:0000:0008:0800:200C:*when connecting to 2001:0DB8:0000:0000:0008:0800:200C:417A). The certificate shall not contain the SAN extension. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported IP address version (e.g. IPv4, IPv6).

This negative test corresponds to the following section of the Application Note 64: 'The exception being, the use of wildcards is not supported when using IP address as the reference identifier.'

Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass Test 6.

Test 7 [conditional]: If the secure channel is used for FPT_ITT, and RFC 5280 is selected, the evaluator shall perform the following tests. Note, when multiple attribute types are selected in the SFR (e.g. when multiple attribute types are combined to form the unique identifier), the evaluator modifies each attribute type in accordance with the matching criteria described in the TSS (e.g. creating a mismatch of one attribute type at a time while other attribute types contain values that will match a portion of the reference identifier):

1) The evaluator shall present a server certificate that does not contain an identifier in the Subject (DN) attribute type(s) that matches the reference identifier. The evaluator shall verify that the connection fails.

2) The evaluator shall present a server certificate that contains a valid identifier as an attribute type other than the expected attribute type (e.g. if the TOE is configured to expect id-atserialNumber=correct_identifier, the certificate could instead include id-at-name=correct_identifier), and does not contain the SAN extension. The evaluator shall verify that the connection fails. Remark: Some systems might require the presence of the SAN extension. In this case the connection would still fail but for the reason of the missing SAN extension instead of the mismatch of CN and reference identifier. Both reasons are acceptable to pass this test.

3) The evaluator shall present a server certificate that contains a Subject attribute type that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.

4) The evaluator shall confirm that all use of wildcards results in connection failure regardless of whether the wildcards are used in the left or right side of the presented identifier. (Remark: Use of wildcards is not addressed within RFC 5280.)

Test 1 - This test was repeated for 2 variations as follows: IPv4 addresses in CN or SAN (LDAP and Syslog) and Identifiers defined in RFC 6125 (LDAP, Syslog, and ITT communications). Note LDAP is only applicable to the Manager and virtual manager.

The evaluator configured the TOE to connect with the test server using TLS alternately configured with matching certificate identifiers and certificate identifiers that do not match the reference identifier in either the SAN or the CN. The evaluator confirmed that the connections were only successful when the identifier fulfilled the required rules.

Test 2-4 – These tests were demonstrated as part of Test 1.

Test 5 – For the ITT, syslog, and LDAP (Managers only) channels, the evaluator configured the TOE to connect with the test server using TLS, with the test server alternately configured with a DNS name certificate identifier using wildcards in various places for both the CN and SAN. The TOE does not support wildcards so all attempts were rejected by the TOE.

Test 6 – For the ITT and syslog channels the evaluator attempted to connect with the test server using TLS with the test server configured with a certificate identifier using an IP address with a wildcard. The TOE does not support wildcards so the attempt was rejected by the TOE.

Test 7 – Not applicable. RFC 5280 is not selected.

2.3.11.3 NDcPP22e:FCS_TLSC_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

Test 1: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds and a trusted channel can be established.

Test 2: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.

Test 3 : The purpose of this test to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.

Test 1 - This successful connection with a valid certificate chain was demonstrated in FCS_TLSC_EXT.2.1, Test 1.

Test 2 - This test has been performed in several other test activities. Specifically, this test repeats the assurance activities as described here.

match the reference identifier -- Corresponds to FCS_TLSC_EXT.1.2 Tests 1 through 7.

validate certificate path -- Corresponds to FIA_X509_EXT.1/REV.1 Test 1 and FIA_X509_EXT.1/ITT.1 Test 1

validate expiration date -- Corresponds to FIA_X509_EXT.1/REV.1 Test 2 and FIA_X509_EXT.1/ITT.1 Test 2

determine the revocation status -- Corresponds to FIA_X509_EXT.2 Test 1.

Test 3 – Not applicable. The TOE does not support administrator override of any of the certificate checks.

2.3.11.4 NDcPP22e:FCS_TLSC_EXT.1.4

TSS Assurance Activities: The evaluator shall verify that TSS describes the Supported Elliptic Curves/Supported Groups Extension and whether the required behavior is performed by default or may be configured.

Section 6.1, Table 18 (FCS_TLSC_EXT.1) in the ST states that the TOE only supports TLSv1.2 with AES 128 or 256 bit symmetric ciphers in CBC and GCM modes, in conjunction with SHA (SHA1, SHA256, and SHA384), RSA, DHE (Diffie-Hellman Group 14), ECDHE (NIST curves supported are secp256r1, secp384r1, and secp521r1) and ECDSA. By default, all the Supported Elliptic Curves Extension options are presented in the Client Hello and this behavior is not configurable.

Guidance Assurance Activities: If the TSS indicates that the Supported Elliptic Curves/Supported Groups Extension must be configured to meet the requirement, the evaluator shall verify that AGD guidance includes configuration of the Supported Elliptic Curves/Supported Groups Extension.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used. The only enabled EC curves are secp256r1, secp384r1, and secp521r1.

Testing Assurance Activities: Test 1 [conditional]: If the TOE presents the Supported Elliptic Curves/Supported Groups Extension, the evaluator shall configure the server to perform ECDHE or DHE (as applicable) key exchange using each of the TOE's supported curves and/or groups. The evaluator shall verify that the TOE successfully connects to the server.

The results below are iterated for 3 variations as follows: ITT Communication Results, LDAP Results and Syslog Results. LDAP is only supported for the Manager and Virtual Manager devices.

Test 1 - The evaluator attempted to establish a TLS session with the TOE when the evaluator's server specified only one key exchange method in the Server Hello from the list the TOE’s supported curves. The evaluator observed the connection established successfully for each supported curve.

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: For all tests in this chapter the TLS server used for testing of the TOE shall be configured not to require mutual authentication.

All testing performed for FCS_TLSC_EXT.1 was performed without mutual authentication.

2.3.12 TLS Client Support for Mutual Authentication (NDcPP22e:FCS_TLSC_EXT.2)

2.3.12.1 NDcPP22e:FCS_TLSC_EXT.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.

Section 6.1, Table 18 (FCS_TLSC_EXT.2) in the ST states that the TLS clients of each TOE component support mutual authentication (FCS_TLSC_EXT.2) for connections to the syslog server (FTP_ITC.1), and for interconnections between distributed TOE components (FPT_ITT.1). Mutual authentication is not supported for TLS connections to LDAP servers (FTP_ITC.1).

Component Guidance Assurance Activities: If the TSS indicates that mutual authentication using X.509v3 certificates is used, the evaluator shall verify that the AGD guidance includes instructions for configuring the client-side certificates for TLS mutual authentication.

Section *Certificates* of the **Admin Guide** defines procedures for updating the certificates of the TOE appliances to be used for the TLS distributed TOE channel. Section *Review Appliance Identity Certificate Requirements* describes all the requirements for the identity certificates so that the certificates are verified successfully.

Component Testing Assurance Activities: For all tests in this chapter the TLS server used for testing of the TOE shall be configured to require mutual authentication. (covered by FCS_TLSC_EXT.1.1 Test 1 and testing for FIA_X.509_EXT.*).

Refer to FCS_TLSC_EXT.1 and FIA_X509_EXT where mutual authentication was tested for both ITT and Syslog.

2.3.13 TLS Server Protocol Without Mutual Authentication - per TD0635 (NDcPP22e:FCS_TLSS_EXT.1)

2.3.13.1 NDcPP22e:FCS_TLSS_EXT.1.1

TSS Assurance Activities: The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component.

Section 6.1, Table 18 (FCS_TLSS_EXT.1) in the ST states that the TOE supports TLSv1.2 with AES 128 or 256 bit symmetric ciphers in CBC and GCM modes, in conjunction with SHA (SHA1, SHA256, and SHA384), RSA, DHE (Diffie-Hellman Group 14), and ECDHE (NIST curves supported are secp256r1, secp384r1, and secp521r1). The TOE will use secp521r1 if supported by the client, otherwise secp384r1, and lastly secp256r1.

The following TLS cipher suites are implemented by the TOE in CC mode:

- Ciphersuites relevant to FPT_ITT, FCS_TLSC_EXT.2, and FCS_TLSS_EXT.2 (client and server) are as listed in sections 5.3.3.11 and 5.3.3.13 in the ST.
- Ciphersuites relevant to FTP_TRP.1 and FCS_TLSS_EXT.1 (server only) are as listed in section 5.3.3.13 in the ST.

While the FOM (see section 7.2 of the ST) supports additional cipher suites (for example, RSA_3DES_EDE_CBC_SHA, RSA_DES_CBC_SHA, RSA_RC4_128_MD5, RSA_RC4_128_SHA, etc.), they are all disabled while operating in CC mode.

Guidance Assurance Activities: The evaluator shall check the guidance documentation to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used for any TLS connection, including HTTPS management on the SMC. It further identifies the specific TLS ciphersuites used when the TOE is operating as a TLS server and indicates that only TLSv1.2 is supported. For all TLS functionality, RSA key establishment uses 4096 bits. The only enabled EC curves are secp256r1, secp384r1, and secp521r1; and the Diffie-Hellman parameters are 2048 bits.

Section *Certificates* of the **Admin Guide** defines the best practices and procedures for updating the certificates of the TOE appliances to be used for the TLS distributed TOE channel and HTTPS management on the SMC.

Testing Assurance Activities: Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of an HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

Test 2: The evaluator shall send a Client Hello to the server with a list of ciphersuites that does not contain any of the ciphersuites in the server's ST and verify that the server denies the connection. Additionally, the evaluator shall send a Client Hello to the server containing only the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the server denies the connection.

Test 3: The evaluator shall perform the following modifications to the traffic:

a) Modify a byte in the Client Finished handshake message, and verify that the server rejects the connection and does not send any application data.

b) (Test Intent: The intent of this test is to ensure that the server's TLS implementation immediately makes use of the key exchange and authentication algorithms to: a) Correctly encrypt (D)TLS Finished message and b) Encrypt every (D)TLS message after session keys are negotiated.)

The evaluator shall use one of the claimed ciphersuites to complete a successful handshake and observe transmission of properly encrypted application data. The evaluator shall verify that no Alert with alert level Fatal (2) messages were sent.

The evaluator shall verify that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator shall examine the Finished message (encrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 11 22 33 44 55...) and confirm that it does not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c...), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14 for at least one of three test messages. There is a chance that an encrypted Finished message contains a hexadecimal value of '14' at the position where a plaintext Finished message would contain the message type code '14'. If the observed Finished message contains a hexadecimal value of '14' at the position where the plaintext Finished message would contain the message type code, the test shall be repeated three times in total. In case the value of '14' can be observed in all three tests it can be assumed that the Finished message has indeed been sent in plaintext and the test has to be regarded as 'failed'. Otherwise it has to be assumed that the observation of the value '14' has been due to chance and that the Finished message has indeed been sent encrypted. In that latter case the test shall be regarded as 'passed'.

Note: The TOE listens to both TLS traffic for HTTPS and ITT communication on the default TLS/HTTPS port 443. As a result, the TOE always sends a certificate request, and then determines the functionality by the provided certificate. If the TOE does not receive a certificate with an identifier matching an expected ITT certificate, including not receiving a certificate at all, the TLS connection succeeds but the TOE interprets the connection as a TLS/HTTPS session attempt. The authentication to the WebUI is then performed via password authentication thus certificate mutual authentication is not supported for the HTTPS Web UI regardless of a certificate request being sent.

Test 1 - The evaluator attempted to connect to the TOE using each of the ciphersuites identified by the Protection Profile. A packet capture was obtained for each connection attempt. The evaluator verified that successful connections were only established with the claimed ciphersuites. These tests were repeated for 2 variations as follows: RSA ciphers and ECDSA ciphers.

Test 2 - The evaluator attempted to connect to the TOE using all ciphers except those defined in the PP and observed that the connection failed. The evaluator used the openssl s_client to attempt to connect to the TOE using the TLS_NULL_WITH_NULL_NULL ciphersuite and observed that the connection failed.

Test 3a - The evaluator attempted a connection to the TOE where the evaluator modified a byte in the Client Finished handshake message, verified that the server rejected the connection attempt after receiving the client Finished message and no application data was exchanged.

Test 3b - The evaluator viewed a successful TLS connection and saw that the Finished message (Content type hexadecimal 16 and handshake message type hexadecimal 14) is sent immediately after the server's ChangeCipherSpec (Content type hexadecimal 14) message. The evaluator examined the Finished message and confirmed that it did not contain unencrypted data (unencrypted example in hexadecimal of a TLS record containing a Finished message, 16 03 03 00 40 14 00 00 0c...), by verifying that the first byte of the encrypted Finished message does not equal hexadecimal 14, which it did not.

2.3.13.2 NDcPP22e:FCS_TLSS_EXT.1.2

TSS Assurance Activities: The evaluator shall verify that the TSS contains a description of how the TOE technically prevents the use of old SSL and TLS versions.

Section 6.1, Table 18 (FCS_TLSS_EXT.1) in the ST states that an authorized administrator can initiate inbound TLSv1.2 connections using the web-based GUI for remote administration of the TOE. Any session where the client offers the following in the client hello: SSL 2.0, SSL 3.0, TLS 1.0, and TLS 1.1 will be rejected by the TOE's TLS server. The TOE supports TLSv1.2 with AES 128 or 256 bit symmetric ciphers in CBC and GCM modes, in conjunction with SHA (SHA1, SHA256, and SHA384), RSA, DHE (Diffie-Hellman Group 14), and ECDHE (NIST curves supported are secp256r1, secp384r1, and secp521r1).

Guidance Assurance Activities: The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used. This includes limiting the supported TLS versions. It further identifies the specific TLS ciphersuites used when the TOE is operating as a TLS server and indicates that only TLSv1.2 is supported. For all TLS functionality, RSA key establishment uses 4096 bits. The only enabled EC curves are secp256r1, secp384r1, and secp521r1; and the Diffie-Hellman parameters are 2048 bits.

Testing Assurance Activities: The evaluator shall send a Client Hello requesting a connection for all mandatory and selected protocol versions in the SFR (e.g. by enumeration of protocol versions in a test client) and verify that the server denies the connection for each attempt.

The evaluator alternately attempted to connect to the TOE using a control case (TLS not specified), SSL2.0, SSL3.0, TLSv1.0, TLSv1.1 and TLSv1.2 and confirmed that connections were accepted only when claimed TLS versions are used.

2.3.13.3 NDcPP22e:FCS_TLSS_EXT.1.3

TSS Assurance Activities: If using ECDHE and/or DHE ciphers, the evaluator shall verify that the TSS lists all EC Diffie-Hellman curves and/or Diffie-Hellman groups used in the key establishment by the TOE when acting as a TLS Server. For example, if the TOE supports TLS_DHE_RSA_WITH_AES_128_CBC_SHA cipher and Diffie-Hellman parameters with size 2048 bits, then list Diffie-Hellman Group 14.

Section 6.1, Table 18 (FCS_TLSS_EXT.1) in the ST states that the key agreement parameters of the server key exchange message are specified in the RFC 5246 (section 7.4.3) for TLSv1.2. The TOE conforms to the RFC.

The TOE supports TLSv1.2 with AES 128 or 256 bit symmetric ciphers in CBC and GCM modes, in conjunction with SHA (SHA1, SHA256, and SHA384), RSA, DHE (Diffie-Hellman Group 14), and ECDHE (NIST curves supported are secp256r1, secp384r1, and secp521r1). The TOE will use secp521r1 if supported by the client, otherwise secp384r1, and lastly secp256r1.

Guidance Assurance Activities: The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the AGD guidance.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used. This includes limiting the possible key agreement parameters. It further identifies the specific TLS ciphersuites used when the TOE is operating as a TLS server and indicates that only TLSv1.2 is supported. For all TLS functionality, RSA key establishment uses 4096 bits. The only enabled EC curves are secp256r1, secp384r1, and secp521r1; and the Diffie-Hellman parameters are 2048 bits.

Testing Assurance Activities: Test 1: [conditional] If ECDHE ciphersuites are supported:

a) The evaluator shall repeat this test for each supported elliptic curve. The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single supported elliptic curve specified in the Elliptic Curves Extension. The Evaluator shall verify (though a packet capture or instrumented client) that the TOE selects the same curve in the Server Key Exchange message and successfully establishes the connection.

b) The evaluator shall attempt a connection using a supported ECDHE ciphersuite and a single unsupported elliptic curve (e.g. secp192r1 (0x13)) specified in RFC4492, chap. 5.1.1. The evaluator shall verify that the TOE does not send a Server Hello message and the connection is not successfully established.

Test 2: [conditional] If DHE ciphersuites are supported, the evaluator shall repeat the following test for each supported parameter size. If any configuration is necessary, the evaluator shall configure the TOE to use a supported Diffie-Hellman parameter size. The evaluator shall attempt a connection using a supported DHE ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a Server Key Exchange Message where p Length is consistent with the message are the ones configured Diffie-Hellman parameter size(s).

Test 3: [conditional] If RSA key establishment ciphersuites are supported, the evaluator shall repeat this test for each RSA key establishment key size. If any configuration is necessary, the evaluator shall configure the TOE to perform RSA key establishment using a supported key size (e.g. by loading a certificate with the appropriate key size). The evaluator shall attempt a connection using a supported RSA key establishment ciphersuite. The evaluator shall verify (through a packet capture or instrumented client) that the TOE sends a certificate whose modulus is consistent with the configured RSA key size.

Test 1 - The evaluator attempted to establish a TLS session with the TOE when the evaluator's client specified only one key exchange method in the Client Hello. The evaluator observed that the TOE successfully established connections using all of the claimed key exchanges.

Test 2 - The evaluator attempted to establish a TLS session with the TOE when the evaluator's client specified only one key exchange method in the Client Hello. The evaluator observed that the TOE successfully established connections using all of the claimed key exchanges.

Test 3 - The evaluator attempted to establish a TLS session with the TOE when the evaluator's client specified only one key exchange method in the Client Hello. The evaluator observed that the TOE successfully established connections using all of the claimed key exchanges.

2.3.13.4 NDcPP22e:FCS_TLSS_EXT.1.4

TSS Assurance Activities: The evaluator shall verify that the TSS describes if session resumption based on session IDs is supported (RFC 4346 and/or RFC 5246) and/or if session resumption based on session tickets is supported (RFC 5077).

If session tickets are supported, the evaluator shall verify that the TSS describes that the session tickets are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption. The evaluator shall verify that the TSS identifies the key lengths and algorithms used to protect session tickets.

If session tickets are supported, the evaluator shall verify that the TSS describes that session tickets adhere to the structural format provided in section 4 of RFC 5077 and if not, a justification shall be given of the actual session ticket format.

Section 6.1, Table 18 (FCS_TLSS_EXT.1) in the [ST] states The SMC WebUI supports session resumption based on session resumption based on session tickets according to RFC 5077.

Session tickets are encrypted using the symmetric algorithms and key lengths associated with the negotiated ciphersuite and which are consistent with the selections from FCS_COP.1/DataEncryption. Session tickets adhere to the structural format provided in section 4 of RFC 5077.

Guidance Assurance Activities: None Defined

Testing Assurance Activities: Test Objective: To demonstrate that the TOE will not resume a session for which the client failed to complete the handshake (independent of TOE support for session resumption).

Test 1 [conditional]: If the TOE does not support session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2) or session tickets according to RFC5077, the evaluator shall perform the following test:

a) The client sends a Client Hello with a zero-length session identifier and with a SessionTicket extension containing a zero-length ticket.

b) The client verifies the server does not send a NewSessionTicket handshake message (at any point in the handshake).

c) The client verifies the Server Hello message contains a zero-length session identifier or passes the following steps: Note: The following steps are only performed if the ServerHello message contains a non-zero length SessionID.

d) The client completes the TLS handshake and captures the SessionID from the ServerHello.

e) The client sends a ClientHello containing the SessionID captured in step d). This can be done by keeping the TLS session in step d) open or start a new TLS session using the SessionID captured in step d).

f) The client verifies the TOE (1) implicitly rejects the SessionID by sending a ServerHello containing a different SessionID and by performing a full handshake (as shown in Figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

Test 2 [conditional]: If the TOE supports session resumption using session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

a) The evaluator shall conduct a successful handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then initiate a new TLS connection and send the previously captured session ID to show that the TOE resumed the previous session by responding with ServerHello containing the same SessionID immediately followed by ChangeCipherSpec and Finished messages (as shown in Figure 2 of RFC 4346 or RFC 5246).

b) The evaluator shall initiate a handshake and capture the TOE-generated session ID in the Server Hello message. The evaluator shall then, within the same handshake, generate or force an unencrypted fatal Alert message immediately before the client would otherwise send its ChangeCipherSpec message thereby disrupting the handshake. The evaluator shall then initiate a new Client Hello using the previously captured session ID, and verify that the server (1) implicitly rejects the session ID by sending a ServerHello containing a different SessionID and performing a full handshake (as shown in figure 1 of RFC 4346 or RFC 5246), or (2) terminates the connection in some way that prevents the flow of application data.

Test 3 [conditional]: If the TOE supports session tickets according to RFC5077, the evaluator shall carry out the following steps (note that for each of these tests, it is not necessary to perform the test case for each supported version of TLS):

a) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator shall then attempt to correctly reuse the previous session by sending the session ticket in the ClientHello. The evaluator shall confirm that the TOE responds with a ServerHello with an empty SessionTicket extension, NewSessionTicket, ChangeCipherSpec and Finished messages (as seen in figure 2 of RFC 5077).

b) The evaluator shall permit a successful TLS handshake to occur in which a session ticket is exchanged with the non-TOE client. The evaluator will then modify the session ticket and send it as part of a new Client Hello message.

The evaluator shall confirm that the TOE either (1) implicitly rejects the session ticket by performing a full handshake (as shown in figure 3 or 4 of RFC 5077), or (2) terminates the connection in some way that prevents the flow of application data.

Test 1: This test is not applicable as the TOE supports session resumption with session tickets.

Test 2: This test is not applicable as session resumption with session IDs is not supported.

Test 3 - The evaluator first attempted to resume a session using a valid session ticket. The server correctly reuses the client's proposed session ticket and the session is successfully resumed.

The evaluator then attempted a second connection in which the session ticket is modified to be different from the server provided session ticket. The server correctly performs a full handshake and issues a new session ticket to the client.

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.3.14 TLS Server Support for Mutual Authentication (NDcPP22e:FCS_TLSS_EXT.2)

2.3.14.1 NDcPP22e:FCS_TLSS_EXT.2.1

TSS Assurance Activities: The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.

The evaluator shall verify the TSS describes how the TSF uses certificates to authenticate the TLS client. The evaluator shall verify the TSS describes if the TSF supports any fallback authentication functions (e.g. username/password, challenge response) the TSF uses to authenticate TLS clients that do not present a certificate. If fallback authentication functions are supported, the evaluator shall verify the TSS describes whether the fallback authentication functions can be disabled.

Section 6.1, Table 18 (FCS_TLSS_EXT.2) in the ST states that the TLS servers of each TOE component support mutual authentication (FCS_TLSS_EXT.2) for interconnections between distributed TOE components (FPT_ITT.1). Mutual authentication is not supported for TLS connections used for remote administration (FTP_TRP.1). When a TLS session is initiated between two TOE components, the server side of the TLS session will check the client certificate and if the SAN within the client certificate does not match the expected identifier on the server, the connection will be rejected by the server.

Guidance Assurance Activities: If the TSS indicates that mutual authentication using X.509v3 certificates is used, the evaluator shall verify that the AGD guidance includes instructions for configuring the client-side certificates for TLS mutual authentication.

The evaluator shall verify the guidance describes how to configure the TLS client certificate authentication function. If the TSF supports fallback authentication functions, the evaluator shall verify the guidance provides instructions for configuring the fallback authentication functions. If fallback authentication functions can be disabled, the evaluator shall verify the guidance provides instructions for disabling the fallback authentication functions.

Section *Certificates* of the **Admin Guide** indicates that the communication between TOE appliances is authenticated using x509v3 certificates. This section defines the best practices and procedures for replacing the certificates of the TOE appliances to update the initial self-signed certificates that were used as part of registration, thus increasing the security of the distributed TOE channel.

Section *Review Appliance Identity Certificate Requirements* in the **Admin Guide** provides the appliance identity certificate requirements that must be met.

The TOE supports no fallback authentication functions

Testing Assurance Activities: Test 1a [conditional]: If the TOE requires or can be configured to require a client certificate, the evaluator shall configure the TOE to require a client certificate and send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify that the handshake is not finished successfully and no application data flows.

Test 1b [conditional]: If the TOE supports fallback authentication functions and these functions cannot be disabled. The evaluator shall configure the fallback authentication functions on the TOE and configure the TOE to send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify the TOE authenticates the connection using the fallback authentication functions as described in the TSS.

Note: Testing the validity of the client certificate is performed as part of X.509 testing.

Test 2 [conditional]: If TLS 1.2 is claimed for the TOE, the evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify that the connection is denied.

Test 3: The aim of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To carry out this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognised by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not terminate in the claimed CA certificate). The evaluator shall verify that the attempted connection is denied.

Test 4: The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that the server denies the connection. Ideally, the two certificates should be identical except for the Client Authentication purpose.

Test 5: The evaluator shall perform the following modifications to the traffic:

a) Configure the server to require mutual authentication and then connect to the server with a client configured to send a client certificate that is signed by a Certificate Authority trusted by the TOE. The evaluator shall verify that the server accepts the connection.

b) Configure the server to require mutual authentication and then modify a byte in the signature block of the client's Certificate Verify handshake message (see RFC5246 Sec 7.4.8). The evaluator shall verify that the server rejects the connection.

Note: Testing the validity of the client certificate is performed as part of X.509 testing.

The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

Test 6: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.

Test 7: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.

Test 8 [conditional]: The purpose of this test is to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.

Test 1a – This test is not applicable as the TOE's TLS server implementation does not require a client certificate.

Test 1b – The TOE handles both the communication for ITT and HTTPS (Web UI) on the same port. When a TLS handshake is initiated, and no certificate is presented, a Web UI connection is assumed and the TOE falls back to user authentication. The administrator initiated a TLS connection to the TOE without a cert and demonstrated the TOE prompted for user and password authentication.

Test 2 - The evaluator configured a test client to connect to the TOE using TLS with an unsupported signature algorithm (md5). The connection failed.

Test 3 - The evaluator configured a test client to connect to the TOE using TLS with a certificate issued by an imposter root CA (i.e, one with an issuer CA distinguished name matching a trusted CA specified by the TOE in its Certificate Request message, but where the imposter CA's signing key differs). The evaluator observed the TOE reject the connection attempt.

Test 4 - The evaluator configured a test client to connect to the TOE using TLS. The first connection attempt is made with a valid certificate including the client auth EKU. This connection is accepted. During the second connection attempt the testy client provides a certificate without the ClientAuthentication EKU. This connection attempt is rejected.

Test 5a – This test has been performed in FIA_X509_EXT.1.1, Test 5 where multiple certificate modifications were tested.

Test 5b - The evaluator used an openssl s_client to attempt to connect to the TOE, but a modified openssl library causes the client to modify the client Certificate Verify handshake message. The evaluator observed the TOE reject the connection attempt.

Test 6 - This test has been performed in FIA_X509_EXT.1.1 Test 1 where a successful connection with a valid cert chain is tested.

Test 7 – This test has been performed in FIA_X509_EXT.1 Test 1 where certificate failure cases are tested.

Test 8 - Not applicable. The TOE does not support administrative override of certificate validation failures.

2.3.14.2 NDcPP22e:FCS_TLSS_EXT.2.2

TSS Assurance Activities: The evaluator shall ensure that the TSS description required per FIA_X509_EXT.2.1 includes the use of client-side certificates for TLS mutual authentication.

The evaluator shall verify the TSS describes how the TSF uses certificates to authenticate the TLS client. The evaluator shall verify the TSS describes if the TSF supports any fallback authentication functions (e.g. username/password, challenge response) the TSF uses to authenticate TLS clients that do not present a certificate. If fallback authentication functions are supported, the evaluator shall verify the TSS describes whether the fallback authentication functions can be disabled.

See FCS_TLSS_EXT.2.1 above.

Guidance Assurance Activities: If the TSS indicates that mutual authentication using X.509v3 certificates is used, the evaluator shall verify that the AGD guidance includes instructions for configuring the client-side certificates for TLS mutual authentication.

The evaluator shall verify the guidance describes how to configure the TLS client certificate authentication function. If the TSF supports fallback authentication functions, the evaluator shall verify the guidance provides instructions for configuring the fallback authentication functions. If fallback authentication functions can be disabled, the evaluator shall verify the guidance provides instructions for disabling the fallback authentication functions.

See FCS_TLSS_EXT.2.1 above.

Testing Assurance Activities: Test 1a [conditional]: If the TOE requires or can be configured to require a client certificate, the evaluator shall configure the TOE to require a client certificate and send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify that the handshake is not finished successfully and no application data flows.

Test 1b [conditional]: If the TOE supports fallback authentication functions and these functions cannot be disabled. The evaluator shall configure the fallback authentication functions on the TOE and configure the TOE to send a Certificate Request to the client. The evaluator shall attempt a connection while sending a certificate_list structure with a length of zero in the Client Certificate message. The evaluator shall verify the TOE authenticates the connection using the fallback authentication functions as described in the TSS.

Note: Testing the validity of the client certificate is performed as part of X.509 testing.

Test 2 [conditional]: If TLS 1.2 is claimed for the TOE, the evaluator shall configure the server to send a certificate request to the client without the supported_signature_algorithm used by the client's certificate. The evaluator shall attempt a connection using the client certificate and verify that the connection is denied.

Test 3: The aim of this test is to check the response of the server when it receives a client identity certificate that is signed by an impostor CA (either Root CA or intermediate CA). To carry out this test the evaluator shall configure the client to send a client identity certificate with an issuer field that identifies a CA recognised by the TOE as a trusted CA, but where the key used for the signature on the client certificate does not correspond to the CA certificate trusted by the TOE (meaning that the client certificate is invalid because its certification path does not terminate in the claimed CA certificate). The evaluator shall verify that the attempted connection is denied.

Test 4: The evaluator shall configure the client to send a certificate with the Client Authentication purpose in the extendedKeyUsage field and verify that the server accepts the attempted connection. The evaluator shall repeat this test without the Client Authentication purpose and shall verify that the server denies the connection. Ideally, the two certificates should be identical except for the Client Authentication purpose.

Test 5: The evaluator shall perform the following modifications to the traffic:

a) Configure the server to require mutual authentication and then connect to the server with a client configured to send a client certificate that is signed by a Certificate Authority trusted by the TOE. The evaluator shall verify that the server accepts the connection.

b) Configure the server to require mutual authentication and then modify a byte in the signature block of the client's Certificate Verify handshake message (see RFC5246 Sec 7.4.8). The evaluator shall verify that the server rejects the connection.

Note: Testing the validity of the client certificate is performed as part of X.509 testing.

The evaluator shall demonstrate that using an invalid certificate results in the function failing as follows:

Test 6: Using the administrative guidance, the evaluator shall load a CA certificate or certificates needed to validate the presented certificate used to authenticate an external entity and demonstrate that the function succeeds, and a trusted channel can be established.

Test 7: The evaluator shall then change the presented certificate(s) so that validation fails and show that the certificate is not automatically accepted. The evaluator shall repeat this test to cover the selected types of failure defined in the SFR (i.e. the selected ones from failed matching of the reference identifier, failed validation of the certificate path, failed validation of the expiration date, failed determination of the revocation status). The evaluator performs the action indicated in the SFR selection observing the TSF resulting in the expected state for the trusted channel (e.g. trusted channel was established) covering the types of failure for which an override mechanism is defined.

Test 8 [conditional]: The purpose of this test is to verify that only selected certificate validation failures could be administratively overridden. If any override mechanism is defined for failed certificate validation, the evaluator shall configure a new presented certificate that does not contain a valid entry in one of the mandatory fields or parameters (e.g. inappropriate value in extendedKeyUsage field) but is otherwise valid and signed by a trusted CA. The evaluator shall confirm that the certificate validation fails (i.e. certificate is rejected), and there is no administrative override available to accept such certificate.

See FCS_TLSS_EXT.2.1 above.

2.3.14.3 NDcPP22e:FCS_TLSS_EXT.2.3

TSS Assurance Activities: The evaluator shall verify that the TSS describes which types of identifiers are supported during client authentication (e.g. Fully Qualified Domain Name (FQDN)). If FQDNs are supported, the evaluator shall verify that the TSS describes that corresponding identifiers are matched according to RFC6125. For all other types of identifiers, the evaluator shall verify that the TSS describes how these identifiers are parsed from the certificate, what the expected identifiers are and how the parsed identifiers from the certificate are matched against the expected identifiers.

Section 6.1, Table 18 (FCS_TLSS_EXT.2) in the ST states that the TLS servers of each TOE component support mutual authentication (FCS_TLSS_EXT.2) for interconnections between distributed TOE components (FPT_ITT.1). Mutual authentication is not supported for TLS connections used for remote administration (FTP_TRP.1). When a TLS session is initiated between two TOE components, the server side of the TLS session will check the client certificate and if the SAN within the client certificate does not match the expected identifier on the server, the connection will be rejected by the server.

Guidance Assurance Activities: The evaluator shall ensure that the AGD guidance describes the configuration of expected identifier(s) for X.509 certificate-based authentication of TLS clients. The evaluator ensures this description includes all types of identifiers described in the TSS and, if claimed, configuration of the TOE to use a directory server.

Section *Review Appliance Identity Certificate Requirements* of the **Admin Guide** describes that the hostname and IP address of the appliances should be included in the Subject Alt Name of their certificates as they are used as reference identifiers.

Testing Assurance Activities: The evaluator shall send a client certificate with an identifier that does not match an expected identifier and verify that the server denies the connection.

The evaluator configured a test client to connect to the TOE using TLS. The evaluator alternately used a valid client certificate and a certificate with an identifier not configured on the TOE but is otherwise valid (e.g., chains to the root configured on the TOE) and the control connection succeeded while the second connection failed to yield a usable user session.

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: For all tests in this chapter the TLS client used for testing of the TOE shall support mutual authentication.

Refer to FCS_TLSS_EXT.2.1 where mutual authentication was tested.

2.4 Identification and authentication (FIA)

2.4.1 Authentication Failure Management (NDcPP22e:FIA_AFL.1)

2.4.1.1 NDcPP22e:FIA_AFL.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.4.1.2 NDcPP22e:FIA_AFL.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that it contains a description, for each supported method for remote administrative actions, of how successive unsuccessful authentication attempts are detected and tracked. The TSS shall also describe the method by which the remote administrator is prevented from successfully logging on to the TOE, and the actions necessary to restore this ability.

The evaluator shall examine the TSS to confirm that the TOE ensures that authentication failures by remote administrators cannot lead to a situation where no administrator access is available, either permanently or temporarily (e.g. by providing local logon which is not subject to blocking).

Section 6.1, Table 18 (FIA_AFL.1) in the ST states that the administrator can configure the maximum number of failed login attempts (configurable from 1-10 consecutive failed attempts) before the account is locked. Accounts that become locked by this feature will remain locked for an administratively defined number of minutes (configurable from 1-720 minutes). By default, this feature is disabled.

The predefined 'admin' account on each appliance is exempt from being locked out, but in the CC-evaluated configuration that account is disabled. All remote administration is performed via the SMC using non-default administrative accounts on the SMC (local accounts or LDAP accounts), and no remote authentication is permitted to any other TOE component. If all non-default remote administration accounts become locked due to consecutive failed login attempts, the default 'sysadmin' account (which can only login via console and is exempt from lockout) can unlock the default 'admin' account and the default 'admin' account can be used to unlock any locally stored (not LDAP) remote administrative account.

Component Guidance Assurance Activities: The evaluator shall examine the guidance documentation to ensure that instructions for configuring the number of successive unsuccessful authentication attempts and time period (if implemented) are provided, and that the process of allowing the remote administrator to once again successfully log on is described for each 'action' specified (if that option is chosen). If different actions or mechanisms are implemented depending on the secure protocol employed (e.g., TLS vs. SSH), all must be described.

The evaluator shall examine the guidance documentation to confirm that it describes, and identifies the importance of, any actions that are required in order to ensure that administrator access will always be maintained, even if remote administration is made permanently or temporarily unavailable due to blocking of accounts as a result of FIA_AFL.1.

Section *Session Settings (Account Lockout)* of the **Admin Guide** includes information on configuring the number of unsuccessful login attempts before security lockout occurs as well as the duration of the lockout before the user can login again. The Security Lockout rules do not apply to the admin, root, and sysadmin users. Root and sysadmin can only be accessed locally where FIA_AFL.1 does not apply. To ensure lockout cannot be circumvented via the remote admin account, section *Local Authentication* in the **Admin Guide** defines steps to disable the default admin user account.

Component Testing Assurance Activities: The evaluator shall perform the following tests for each method by which remote administrators access the TOE (e.g. any passwords entered as part of establishing the connection protocol or the remote administrator application):

a) Test 1: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE (and, if the time period selection in FIA_AFL.1.2 is included in the ST, then the evaluator shall also use the operational guidance to configure the time period after which access is re-enabled). The evaluator shall test that once the authentication attempts limit is reached, authentication attempts with valid credentials are no longer successful.

b) Test 2: After reaching the limit for unsuccessful authentication attempts as in Test 1 above, the evaluator shall proceed as follows.

If the administrator action selection in FIA_AFL.1.2 is included in the ST then the evaluator shall confirm by testing that following the operational guidance and performing each action specified in the ST to re-enable the remote administrator's access results in successful access (when using valid credentials for that administrator).

If the time period selection in FIA_AFL.1.2 is included in the ST then the evaluator shall wait for just less than the time period configured in Test 1 and show that an authorisation attempt using valid credentials does not result in successful access. The evaluator shall then wait until just after the time period configured in Test 1 and show that an authorisation attempt using valid credentials results in successful access.

Test 1 and Test 2 – The evaluator first set the maximum number of failed login attempts to 3, and lockout duration to 1 minute on the SMC. The evaluator then logged in 3 times with an incorrect password and confirmed that the account was locked as expected. After 1 minute, the evaluator successfully logged in with the correct password. The evaluator then repeated this test with the maximum number of failed login attempts set to 5, and the lockout duration set to 3 minutes on the SMC.

2.4.2 Password Management (NDcPP22e:FIA_PMG_EXT.1)

2.4.2.1 NDcPP22e:FIA_PMG_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that it contains the lists of the supported special character(s) and minimum and maximum number of charters supported for administrator passwords.

Section 6.1, Table 18 (FIA_PMG_EXT.1) in the ST states that the TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: [“!”, “@”, “#”, “$”, “%”,“(“, “)”, [“<”, “>”, “.”, “?”, “/”, “’”, “””, “\”, “|”, “:”, “;”, “`”, “~”, “-“, “_”, “+”, “=”, “{”, “}”, “[“, and “]”.

Minimum password length is settable by the Authorized Administrator, and the minimum number of characters can be set from 8 to 30 inclusive. Password composition rules specifying the types and number of required characters that comprise the password are settable by the Authorized Administrator.

Component Guidance Assurance Activities: The evaluator shall examine the guidance documentation to determine that it:

a) identifies the characters that may be used in passwords and provides guidance to security administrators on the composition of strong passwords, and

b) provides instructions on setting the minimum password length and describes the valid minimum password lengths supported.

Section *Password Policy* of the **Admin Guide** describes the configuration for the password policy that applies to all user interfaces. This section identifies the characters that may be used in passwords, including, uppercase, lowercase, numbers, and the following special characters: < > . , ? / ' " \ | : ; ` ~ ! @ # $ % ^ & * ( ) - _ + = { } [ ]. The section also describes how many characters of each type should be used to ensure a sufficiently complex password. Although the TOE supports passwords of 8 characters or more, guidance recommends a password length between 15 to 30 characters to ensure sufficient password strength.

Component Testing Assurance Activities: The evaluator shall perform the following tests.

Test 1: The evaluator shall compose passwords that meet the requirements in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, and a minimum length listed in the requirement are supported and justify the subset of those characters chosen for testing.

Test 2: The evaluator shall compose passwords that do not meet the requirements in some way. For each password, the evaluator shall verify that the TOE does not support the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that the TOE enforces the allowed characters and the minimum length listed in the requirement and justify the subset of those characters chosen for testing.

Test 1 - The evaluator first configured a password policy for the device for a minimum length of 8 characters and several complexity requirements. The evaluator attempted to change the password, for both the local and remote accounts, to a variety of passwords that either met or did not meet the requirements and observed that only the ones that met the password requirements were successfully implemented. The evaluator configured another password policy to demonstrate other settings were possible. In this case the minimum length was set to 30. The evaluator tested several 30-character passwords to ensure that all supported characters were tested. All password attempts were successful.

Test 2 – Refer to Test 1 where the evaluator tested passwords that did not meet the requirements.

2.4.3 Protected Authentication Feedback (NDcPP22e:FIA_UAU.7)

2.4.3.1 NDcPP22e:FIA_UAU.7.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: None Defined

Component Guidance Assurance Activities: The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps to ensure authentication data is not revealed while entering for each local login allowed.

There are not any necessary preparatory steps to ensure authentication data is obscured. By default, the authentication data is always obscured. This behavior is consistent with the testing where the evaluator found that no password feedback is provided when logging into the local console.

Component Testing Assurance Activities: The evaluator shall perform the following test for each method of local login allowed:

a) Test 1: The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.

The evaluator found that no password feedback is provided when logging into the local console.

2.4.4 Password-based Authentication Mechanism (NDcPP22e:FIA_UAU_EXT.2)

2.4.4.1 NDcPP22e:FIA_UAU_EXT.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: Evaluation Activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.

See FIA_UIA_EXT.1

Component Guidance Assurance Activities: Evaluation Activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.

See FIA_UIA_EXT.1

Component Testing Assurance Activities: Evaluation Activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.

See FIA_UIA_EXT.1

2.4.5 User Identification and Authentication (NDcPP22e:FIA_UIA_EXT.1)

2.4.5.1 NDcPP22e:FIA_UIA_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.4.5.2 NDcPP22e:FIA_UIA_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a 'successful logon'.

The evaluator shall examine the TSS to determine that it describes which actions are allowed before user identification and authentication. The description shall cover authentication and identification for local and remote TOE administration.

For distributed TOEs the evaluator shall examine that the TSS details how Security Administrators are authenticated and identified by all TOE components. If not all TOE components support authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the TSS shall describe how the overall TOE functionality is split between TOE components including how it is ensured that no unauthorized access to any TOE component can occur.

For distributed TOEs, the evaluator shall examine the TSS to determine that it describes for each TOE component which actions are allowed before user identification and authentication. The description shall cover authentication and identification for local and remote TOE administration. For each TOE component that does not support authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 the TSS shall describe any unauthenticated services/services that are supported by the component.

Section 6.1, Table 18 (FIA_UIA_EXT.1) in the ST states that the TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed except for acknowledging the pre-login warning banner. The TOE mediates all administrative actions through its GUI and CLI. Once a potential administrative user attempts to access the TOE through either a directly connected console or remotely through TLS, the TOE prompts the user for a user name and password. Only after the administrative user presents the correct authentication credentials will access to the TOE administrative functionality be granted. No access is allowed to the administrative functionality of the TOE until an administrator is successfully identified and authenticated.

In this distributed TOE all remote administration is performed via the WebUI (GUI) of the SMC. The default 'admin' remote administration account is disabled on all TOE components and only the SMC has non-default remote administrative accounts (defined locally, and/or as LDAP accounts). Each non-SMC (FC, FS, and UDPD) TOE component continues to have a remotely accessible login page, but all remote login attempts will fail because the only existing remote account (the default 'admin' account) is disabled. Every TOE component (SMC, FC, FS, and UDPD) allow local console login using the default 'sysadmin' account.

Section 6.1, Table 18 (FIA_UIA_EXT.2) in the ST states that the TOE provides a local password-based authentication mechanism at its console CLI and its remote GUI. The TOE also supports AAA LDAP authentication at its GUI.

The administrator authentication policies include authentication to the local user database or, optionally for the GUI, redirection to a remote authentication server (LDAP over TLS). Interfaces can be configured to try one or more remote authentication servers, and then fail back to the local user database if the remote authentication servers are inaccessible.

The process for authentication is the same for administrative access whether administration is occurring via a directly connected console cable or remotely via TLS. At initial login in the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative password associated with the user account. The TOE then either grants administrative access (if the combination of username and password is correct) or indicates that the login was unsuccessful. The TOE does not provide a reason for failure in the cases of a login failure.

Component Guidance Assurance Activities: The evaluator shall examine the guidance documentation to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the guidance documentation provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the guidance documentation provides sufficient instruction on limiting the allowed services.

Section *Compliance Mode (FIPS and Common Criteria*) of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used for any TLS connection, including HTTPS management on the SMC.

Section *Certificates* of the **Admin Guide** defines the best practices and procedures for updating the certificates of the TOE appliances to be used for HTTPS management on the SMC.

Section *Secure Installation* in the **Admin Guide** states that initial configuration requires use of the console port to configure networking and refers to the SNA Hardware Installation Guide which provides instructions for how to connect to the local console ports on all physical appliances. The SNA Installation and Configuration Guide provides instructions for console access on all virtual appliances.

Section Log in to the SNA Web App of the Admin Guide describes how to log into the Web UI over HTTPS. Section **Local Authentication** describes how to log into the local console and includes references to the SNA Hardware Installation Guide and the SNA Installation and Configuration Guide for how to access the local console on both physical and virtual appliances.

Component Testing Assurance Activities: The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method:

a) Test 1: The evaluator shall use the guidance documentation to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

b) Test 2: The evaluator shall configure the services allowed (if any) according to the guidance documentation, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement.

c) Test 3: For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.

d) Test 4: For distributed TOEs where not all TOE components support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2, the evaluator shall test that the components authenticate Security Administrators as described in the TSS.

Test 1 – The SMC supports remote authentication using local password or LDAP. All TOE components support local authentication, but local authentication does not support LDAP. The evaluator first attempted to log into the SMC’s Web UI using local authentication with invalid credentials and viewed that the attempt failed. The evaluator then logged in with valid credentials and viewed that the attempt was successful. Next the evaluator attempted to log into the SMC’s Web UI using LDAP authentication with invalid credentials and viewed that the attempt failed. The evaluator then logged in with valid credentials and viewed that the attempt was successful. Lastly, the evaluator attempted to login to the SMC’s local console with invalid credentials and viewed the attempt failed. The evaluator then logged in to the local console with valid credentials and the attempt was successful.

Test 2 – The evaluator observed during Test 1 that the only service available to an administrator prior to login was viewing the warning banner and initiating a login. This is demonstrated in FTA_TAB.1.

Test 3 - The evaluator observed during Test 1 that the only service available to an administrator prior to login was viewing the warning banner and initiating a login.

Test 4 – This test was demonstrated in test 1. The SMC supports remote authentication using local password or LDAP. The other Distributed TOE components do not support remote authentication as they are managed remotely by the SMC. All TOE components support local authentication but local authentication does not support LDAP. This was tested for all TOE components as part of FIA_UIA_EXT.1, Test 1.

2.4.6 X.509 Certificate Validation (NDcPP22e:FIA_X509_EXT.1/ITT)

2.4.6.1 NDcPP22e:FIA_X509_EXT.1.1/ITT

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step. It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the device. The evaluator shall perform the following tests for FIA_X509_EXT.1.1/ITT:

These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols.:

a) Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function, and shall use this chain to demonstrate that the function succeeds.

Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).

Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.

b) Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.

c) Test 3: The evaluator shall test that the TOE can properly handle revoked certificates - conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the TOE certificate and revocation of the TOE intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. No testing is required if no revocation method is selected. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore, the revoked certificate(s) used for testing shall not be a trust anchor.

d) Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.

e) Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)

f) Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

g) Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.)

h) The following tests are run when a minimum certificate path length of three certificates is implemented.

Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall conduct the following tests:

Test 8a: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

Test 8c: The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.

(TD0527 12/2020 update applied)

These tests were repeated for 2 variations as follows: ITT Client and ITT Server.

Test 1 - For this test, the evaluator configured the TOE to have the trusted root CA used by either test server/client to anchor all of its certificates. In each case, the evaluator then attempted to connect the TOE to either the test server/client expecting TOE to accept the first TLS connection (where the test server/client presents a complete chain) and reject the second TLS connection (where the test server/client presents only when the trusted root CA was properly configured forming a valid certificate chain.

Test 2 - The evaluator alternately configured a test server/client to send an authentication certificate 1) that is valid, 2) that is expired, and 3) issued by an intermediate CA that is expired. In each case, the evaluator then attempted to connect the TOE to the test server/client and confirmed that the connection only succeeded if there were no expired certificates.

Test 3 – Not applicable. The TOE does not support revocation checking for ITT communication.

Test 4 - Not applicable. The TOE does not support revocation checking for ITT communication.

Test 5 - The evaluator alternately configured a test server/client to send an authentication certificate 1) that is valid, 2) that has one byte in the ASN1 field changed, 3) that has one byte in the certificate signature changed, and 4) that has one byte in the certificate public key changed. In each case, the evaluator then attempted to connect the TOE to the test server/client and confirmed that the connection only succeeded if the certificate was not modified/corrupted.

Test 6 - This test has been performed in FIA_X509_EXT.1.1/ITT, Test 5.

Test 7 - This test has been performed in FIA_X509_EXT.1.1/ITT, Test 5.

Test 8 - This test is not applicable as the TOE does not support EC certificates.

2.4.6.2 NDcPP22e:FIA_X509_EXT.1.2/ITT

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: The evaluator shall perform the following tests for FIA_X509_EXT.1.2/ITT. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/ITT. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted. The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation). For each of the following tests the evaluator shall create a chain of at least two certificates: a self-signed root CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

a) Test 1: The evaluator shall ensure that one CA in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

b) Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

These tests were repeated for 2 variations as follows: ITT Client and ITT Server.

Test 1 and Test 2- The evaluator alternately configured a test server/client to send an authentication certificate issued by a Sub CA with no BasicConstraints and with BasicConstraints but the CA Flag set to false. In each case, the evaluator then attempted to connect the TOE to the test server/client and confirmed that the connection was rejected in each case.

Component TSS Assurance Activities: The evaluator shall examine the TSS to ensure it describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). If selected, the TSS shall describe how certificate revocation checking is performed. It is not sufficient to verify the status of a X.509 certificate only when it's loaded onto the device.

Section 6.1, Table 18 (FIA_X509_EXT.1/ITT) states that when the distributed components initiate TLS connections to each other the client validates the server's certificates during session establishment and validate those certificates against the locally stored root certificate. Revocation checking (neither CRL nor OCSP checking) is not performed for the TLS connection between TOE components. The extendedKeyUsage field is validated according to the following rules - Server certificates have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field and the Client certificates have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field. Checking is done for the 'basicConstraints' extension and the 'cA' flag to determine whether they are present and set to TRUE. If they are not, the CA certificate is not accepted as a trust anchor.

Component Guidance Assurance Activities: The evaluator shall also ensure that the guidance documentation describes where the check of validity of the certificates takes place, describes any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) and describe how certificate revocation checking is performed.

Section *Review Appliance Identity Certificate Requirements* in the **Admin Guide** provides the appliance identity certificate requirements that must be met including details for each field in the certificate including CA signature, chain length, basic constraints, extended key usage, format, SAN, RSA Key length and date range.

The extendedKeyUsage field is validated according to the following rules - Server certificates have the Server Authentication purpose in the extendedKeyUsage field and the Client certificates have the Client Authentication purpose in the extendedKeyUsage field.

Revocation is not supported for inter-TOE communication.

Component Testing Assurance Activities: None Defined

2.4.7 X.509 Certificate Validation (NDcPP22e:FIA_X509_EXT.1/Rev)

2.4.7.1 NDcPP22e:FIA_X509_EXT.1.1/Rev

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: The evaluator shall demonstrate that checking the validity of a certificate is performed when a certificate is used in an authentication step or when performing trusted updates (if FPT_TUD_EXT.2 is selected). It is not sufficient to verify the status of a X.509 certificate only when it is loaded onto the TOE. It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected). The evaluator shall perform the following tests for FIA_X509_EXT.1.1/Rev. These tests must be repeated for each distinct security function that utilizes X.509v3 certificates. For example, if the TOE implements certificate-based authentication with IPSEC and TLS, then it shall be tested with each of these protocols:

a) Test 1a: The evaluator shall present the TOE with a valid chain of certificates (terminating in a trusted CA certificate) as needed to validate the leaf certificate to be used in the function, and shall use this chain to demonstrate that the function succeeds. Test 1a shall be designed in a way that the chain can be 'broken' in Test 1b by either being able to remove the trust anchor from the TOEs trust store, or by setting up the trust store in a way that at least one intermediate CA certificate needs to be provided, together with the leaf certificate from outside the TOE, to complete the chain (e.g. by storing only the root CA certificate in the trust store).

Test 1b: The evaluator shall then 'break' the chain used in Test 1a by either removing the trust anchor in the TOE's trust store used to terminate the chain, or by removing one of the intermediate CA certificates (provided together with the leaf certificate in Test 1a) to complete the chain. The evaluator shall show that an attempt to validate this broken chain fails.

b) Test 2: The evaluator shall demonstrate that validating an expired certificate results in the function failing.

c) Test 3: The evaluator shall test that the TOE can properly handle revoked certificates - conditional on whether CRL or OCSP is selected; if both are selected, then a test shall be performed for each method. The evaluator shall test revocation of the peer certificate and revocation of the peer intermediate CA certificate i.e. the intermediate CA certificate should be revoked by the root CA. The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator then attempts the test with a certificate that has been revoked (for each method chosen in the selection) to ensure when the certificate is no longer valid that the validation function fails. Revocation checking is only applied to certificates that are not designated as trust anchors. Therefore the revoked certificate(s) used for testing shall not be a trust anchor.

d) Test 4: If OCSP is selected, the evaluator shall configure the OCSP server or use a man-in-the-middle tool to present a certificate that does not have the OCSP signing purpose and verify that validation of the OCSP response fails. If CRL is selected, the evaluator shall configure the CA to sign a CRL with a certificate that does not have the cRLsign key usage bit set, and verify that validation of the CRL fails.

e) Test 5: The evaluator shall modify any byte in the first eight bytes of the certificate and demonstrate that the certificate fails to validate. (The certificate will fail to parse correctly.)

f) Test 6: The evaluator shall modify any byte in the last byte of the certificate and demonstrate that the certificate fails to validate. (The signature on the certificate will not validate.)

g) Test 7: The evaluator shall modify any byte in the public key of the certificate and demonstrate that the certificate fails to validate. (The hash of the certificate will not validate.)

h) The following tests are run when a minimum certificate path length of three certificates is implemented.

Test 8: (Conditional on support for EC certificates as indicated in FCS_COP.1/SigGen). The evaluator shall conduct the following tests:

Test 8a: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a valid chain of EC certificates (terminating in a trusted CA certificate), where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

Test 8b: (Conditional on TOE ability to process CA certificates presented in certificate message) The test shall be designed in a way such that only the EC root certificate is designated as a trust anchor, and by setting up the trust store in a way that the EC Intermediate CA certificate needs to be provided, together with the leaf certificate, from outside the TOE to complete the chain (e.g. by storing only the EC root CA certificate in the trust store). The evaluator shall present the TOE with a chain of EC certificates (terminating in a trusted CA certificate), where the intermediate certificate in the certificate chain uses an explicit format version of the Elliptic Curve parameters in the public key information field, and is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

Test 8c: The evaluator shall establish a subordinate CA certificate, where the elliptic curve parameters are specified as a named curve, that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is accepted into the TOE's trust store. The evaluator shall then establish a subordinate CA certificate that uses an explicit format version of the elliptic curve parameters, and that is signed by a trusted EC root CA. The evaluator shall attempt to load the certificate into the trust store and observe that it is rejected, and not added to the TOE's trust store.

(TD0527 12/2020 update applied)

These tests were repeated for 2 variations as follows: LDAP and Syslog.

Test 1 – For this test, the evaluator configured the TOE to have the trusted root CA used by the test server to anchor all of its certificates. In each case, the evaluator then attempted to connect the TOE to the test server expecting TOE to accept the first TLS connection (where the test server presents a complete chain) and reject the second TLS connection (where the test server presents only when the trusted root CA was properly configured forming a valid certificate chain.

Test 2 – The evaluator alternately configured a test server to send an authentication certificate 1) that is valid, 2) that is expired, and 3) issued by an intermediate CA that is expired. In each case, the evaluator then attempted to connect the TLSC TOE client to the test server and the TOE and confirmed that the connection only succeeded if there were no expired certificates.

Test 3 – The evaluator alternately configured a test server to send an authentication certificate 1) that is valid, 2) that is revoked, and 3) issued by an intermediate CA that is revoked. In each case, the evaluator then attempted to connect the TLSC TOE client between the test server and the TOE and confirmed that a connection only succeeded if there were no revoked certificates. These tests were performed for 2 variations as follows: CRL and OCSP.

Test 4 – CRL: The evaluator alternately configured a test server to send an authentication certificate 1) that is valid, 2) issued by an intermediate CA referring to a CRL revocation server where the signer lacks cRLSign, and 3) issued by an intermediate CA whose issuer CA refers to a CRL revocation server where the signer lacks cRLSign. In each case, the evaluator then attempted to connect the TLSC TOE client to the test server and confirmed that the connection only succeeded if all retreived CRLs are signed using certificates with cRLSign.

OCSP: The evaluator alternately configured a test server to send an authentication certificate 1) that is valid, 2) that has a root that refers to an OCSP revocation server where the signer lacks OCSPSigning, 3) issued by an intermediate CA whose issuer CA refers to an OCSP revocation server where the signer lacks OCSPSigning, and 4) issued by an intermediate CA referring to an OCSP revocation server where the signer lacks OCSPSigning. In each case, the evaluator then attempted to connect the TLSC TOE client to the test server and confirmed that the connection only succeeded if all retrieved OCSP responses are signed using certificates with OCSPSigning.

Test 5 – The evaluator alternately configured a test server to send an authentication certificate 1) that is valid, 2) that has one byte in the ASN1 field changed, 3) that has one byte in the certificate signature changed, and 4) that has one byte in the certificate public key changed. In each case, the evaluator then attempted to connect the TLSC TOE client to the test server and confirmed that a connection succeeded only if the certificate is not modified/corrupted.

Test 6 - This test was performed in FIA_X509_EXT.1.1/Rev, Test 5.

Test 7 – This test was performed in FIA_X509_EXT.1.1/Rev, Test 5.

Test 8 – This test is not applicable as the TOE does not support EC certificates.

2.4.7.2 NDcPP22e:FIA_X509_EXT.1.2/Rev

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: The evaluator shall perform the following tests for FIA_X509_EXT.1.2/Rev. The tests described must be performed in conjunction with the other certificate services assurance activities, including the functions in FIA_X509_EXT.2.1/Rev. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules. Where the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) then the associated extendedKeyUsage rule testing may be omitted.

The goal of the following tests is to verify that the TOE accepts a certificate as a CA certificate only if it has been marked as a CA certificate by using basicConstraints with the CA flag set to True (and implicitly tests that the TOE correctly parses the basicConstraints extension as part of X509v3 certificate chain validation). For each of the following tests the evaluator shall create a chain of at least three certificates: a self-signed root CA certificate, an intermediate CA certificate and a leaf (node) certificate. The properties of the certificates in the chain are adjusted as described in each individual test below (and this modification shall be the only invalid aspect of the relevant certificate chain).

a) Test 1: The evaluator shall ensure that at least one of the CAs in the chain does not contain the basicConstraints extension. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate without the basicConstraints extension to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

b) Test 2: The evaluator shall ensure that at least one of the CA certificates in the chain has a basicConstraints extension in which the CA flag is set to FALSE. The evaluator confirms that the TOE rejects such a certificate at one (or both) of the following points: (i) as part of the validation of the leaf certificate belonging to this chain; (ii) when attempting to add a CA certificate with the CA flag set to FALSE to the TOE's trust store (i.e. when attempting to install the CA certificate as one which will be retrieved from the TOE itself when validating future certificate chains).

The evaluator shall repeat these tests for each distinct use of certificates. Thus, for example, use of certificates for TLS connection is distinct from use of certificates for trusted updates so both of these uses would be tested. But there is no need to repeat the tests for each separate TLS channel in FTP_ITC.1 and FTP_TRP.1/Admin (unless the channels use separate implementations of TLS).

These tests were repeated for 2 variations as follows: LDAP and Syslog.

Test 1 and Test 2 –The evaluator alternately configured a test server to send an authentication certificate issued by a Sub CA with no BasicConstraints and with BasicConstraints but the CA Flag set to false. In each case, the evaluator then attempted to connect the TLSC TOE client to the test server and confirmed that the connection was rejected in each case.

Component TSS Assurance Activities: The evaluator shall ensure the TSS describes where the check of validity of the certificates takes place, and that the TSS identifies any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied). It is expected that revocation checking is performed when a certificate is used in an authentication step and when performing trusted updates (if selected). It is not necessary to verify the revocation status of X.509 certificates during power-up self-tests (if the option for using X.509 certificates for self-testing is selected).

The TSS shall describe when revocation checking is performed and on what certificates. If the revocation checking during authentication is handled differently depending on whether a full certificate chain or only a leaf certificate is being presented, any differences must be summarized in the TSS section and explained in the Guidance.

Section 6.1, Table 18 (FIA_X509_EXT.1/Rev) in the ST states that the TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS connections. The certificate validation checking takes place during the TLS session setup. Certificate revocation checking is supported by the TOE through use of OCSP and CRL when the TOE is validating server certificates when initiating outbound TLS connections to syslog and LDAP servers (for FTP_ITC only). The extendedKeyUsage field is validated according to the following rules: Server certificates have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field; the Client certificates have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field; and the OCSP responder certificates have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field. Certificate validation includes verification of the basicConstraints extension and the CA flag to determine whether they are present and set to TRUE. The local CA certificate that was imported must contain the basic constraints extension with the CA flag set to true, the check also ensure that the key usage extension is present.

In all use cases (syslog-over-TLS and LDAP-over-TLS, whether using CRL or OCSP) if the connection to determine the certificate validity cannot be established, the TOE will not accept the certificate.

There are no differences in how revocation checking is handled for a leaf cert or full chain.

Component Guidance Assurance Activities: The evaluator shall also ensure that the guidance documentation describes where the check of validity of the certificates takes place, describes any of the rules for extendedKeyUsage fields (in FIA_X509_EXT.1.1) that are not supported by the TOE (i.e. where the ST is therefore claiming that they are trivially satisfied) and describes how certificate revocation checking is performed and on which certificate.

Section *Review Appliance Identity Certificate Requirements* in the **Admin Guide** provides the appliance identity certificate requirements that must be met including details for each field in the certificate including CA signature, chain length, basic constraints, extended key usage, format, SAN, RSA Key length and date range.

The extendedKeyUsage field is validated according to the following rules - Server certificates have the Server Authentication purpose in the extendedKeyUsage field and the Client certificates have the Client Authentication purpose in the extendedKeyUsage field.

Sections *Configure the LDAP Settings* and *Configure the Audit Log Destination Settings* describe how to enable revocation checking of the server’s certificate using either CRL or OCSP, depending on which is defined within the server’s certificate.

Component Testing Assurance Activities: None Defined

2.4.8 X.509 Certificate Authentication (NDcPP22e:FIA_X509_EXT.2)

2.4.8.1 NDcPP22e:FIA_X509_EXT.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.4.8.2 NDcPP22e:FIA_X509_EXT.2.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall check the TSS to ensure that it describes how the TOE chooses which certificates to use, and any necessary instructions in the administrative guidance for configuring the operating environment so that the TOE can use the certificates.

The evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel. The evaluator shall verify that any distinctions between trusted channels are described. If the requirement that the administrator is able to specify the default action, then the evaluator shall ensure that the guidance documentation contains instructions on how this configuration action is performed.

Section 6.1, Table 18 (FIA_X509_EXT.2) in the ST states that the TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS connections. The certificate validation checking takes place during the TLS session setup. Certificate revocation checking is supported by the TOE through use of OCSP and CRL when the TOE is validating server certificates when initiating outbound TLS connections to syslog and LDAP servers (for FTP_ITC only). Certificate validation includes verification of the basicConstraints extension and the CA flag to determine whether they are present and set to TRUE. The local CA certificate that was imported must contain the basic constraints extension with the CA flag set to true, the check also ensure that the key usage extension is present. In all use cases (syslog-over-TLS and LDAP-over-TLS, whether using CRL or OCSP) if the connection to determine the certificate validity cannot be established, the TOE will not accept the certificate.

Component Guidance Assurance Activities: The evaluator shall also ensure that the guidance documentation describes the configuration required in the operating environment so the TOE can use the certificates. The guidance documentation shall also include any required configuration on the TOE to use the certificates. The guidance document shall also describe the steps for the Security Administrator to follow if the connection cannot be established during the validity check of a certificate used in establishing a trusted channel.

Section *Certificates* of the **Admin Guide** describes how to generate a Certificate Signing Request (CSR) with the following optional fields: Organization, Organizational Unit, Locality or City, State or Province, Country Code, and Email Address. This certificate is downloaded to be signed by an external CA. This section further describes how to add the external signing CA to the TOE's trust store and then replace the existing appliance certificate with the newly signed CSR.

Section *Review Appliance Identity Certificate Requirements* in the **Admin Guide** provides the appliance identity certificate requirements that must be met including details for each field in the certificate including CA signature, chain length, basic constraints, extended key usage, format, SAN, RSA Key length and date range.

Sections *Troubleshooting LDAP Connectivity* and *Troubleshooting Syslog Connectivity* describe what actions an administrator should take in the event the connection cannot be established during the validity check of a certificate.

Component Testing Assurance Activities: The evaluator shall perform the following test for each trusted channel:

The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is administrator-configurable, then the evaluator shall follow the guidance documentation to determine that all supported administrator-configurable options behave in their documented manner.

This test was repeated for 2 variations as follows: LDAP and Syslog. The ITT communication channels do not support revocation.

The evaluator alternately configured a test server to send an authentication certificate with valid/accessible revocation servers and an authentication certificate with revocation information referring to an inaccessible revocation server. In each case, the evaluator then attempted to connect the TLSC TOE client to the test server and confirmed that the connection was successful when the revocation server is accessible and when the revocation server is not accessible the connection was rejected.

2.4.9 X.509 Certificate Requests (NDcPP22e:FIA_X509_EXT.3)

2.4.9.1 NDcPP22e:FIA_X509_EXT.3.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.4.9.2 NDcPP22e:FIA_X509_EXT.3.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: If the ST author selects 'device-specific information', the evaluator shall verify that the TSS contains a description of the device-specific fields used in certificate requests.

Not applicable. The ST does not select 'device-specific information'.

Component Guidance Assurance Activities: The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certification Request. If the ST author selects 'Common Name', 'Organization', 'Organizational Unit', or 'Country', the evaluator shall ensure that this guidance includes instructions for establishing these fields before creating the Certification Request.

Section *Certificates* of the **Admin Guide** describes how to generate a Certificate Signing Request (CSR) with the following optional fields: Organization, Organizational Unit, Locality or City, State or Province, Country Code, and Email Address. This certificate is downloaded to be signed by an external CA. This section further describes how to add the external signing CA to the TOE's trust store and then replace the existing appliance certificate with the newly signed CSR.

Component Testing Assurance Activities: The evaluator shall perform the following tests:

a) Test 1: The evaluator shall use the guidance documentation to cause the TOE to generate a Certification Request. The evaluator shall capture the generated request and ensure that it conforms to the format specified. The evaluator shall confirm that the Certification Request provides the public key and other required information, including any necessary user-input information.

b) Test 2: The evaluator shall demonstrate that validating a response message to a Certification Request without a valid certification path results in the function failing. The evaluator shall then load a certificate or certificates as trusted CAs needed to validate the response message, and demonstrate that the function succeeds.

Test 1 – The evaluator generated a CSR and then viewed that the CSR conformed to the format specified and provided the public key and other required information.

Test 2 – The evaluator first attempted to validate a response message to a Certification Request without a valid certification path and verified that the function failed. The evaluator then loaded certificates as trusted CAs needed to validate the response message, and found that the function succeeds.

2.5 Security management (FMT)

2.5.1 Management of security functions behaviour (NDcPP22e:FMT_MOF.1/ManualUpdate)

2.5.1.1 NDcPP22e:FMT_MOF.1.1/ManualUpdate

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: For distributed TOEs it is required to verify the TSS to ensure that it describes how every function related to security management is realized for every TOE component and shared between different TOE components. The evaluator shall confirm that all relevant aspects of each TOE component are covered by the FMT SFRs.

There are no specific requirements for non-distributed TOEs.

Section 6.1, Table 18 (FMT_MOF.1/ManualUpdate) in the ST states that manual software updates can only be initiated by the authorized administrator through the SMC GUI. Updates for all TOE components are managed through the GUI of SMC. All update files are first uploaded by an administrator to the SMC, then an SMC administrator manually initiates installing of updates to each appliance (SMC, FC, FS, and UDPD). When updating appliances other than the SMC itself, the update files are transmitted from the SMC to each other TOE component over the same TLS connection used for all other distributed TOE (FPT_ITT.1) communications.

Section 9 in the ST provides a mapping of SFRs to TOE components. Manual updates are managed by the SMC, but otherwise, manual updates, the management of TSF data and the specification of management functions are implemented by all components. These functions are primarily performed via the SMC, but there is a limited set of functions implemented via the CLI on all components (e.g. change password and change time). All security management functions and the corresponding configuration information are identified or referenced throughout this AAR with the requirement to which they apply.

Component Guidance Assurance Activities: The evaluator shall examine the guidance documentation to determine that any necessary steps to perform manual update are described. The guidance documentation shall also provide warnings regarding functions that may cease to operate during the update (if applicable).

For distributed TOEs the guidance documentation shall describe all steps how to update all TOE components. This shall contain description of the order in which components need to be updated if the order is relevant to the update process. The guidance documentation shall also provide warnings regarding functions of TOE components and the overall TOE that may cease to operate during the update (if applicable).

Section *Product Updates* of the **Admin Guide** describes how the admin can view the installed version of the software of each of the TOE components through Central Management > Update Manager. The section further describes the steps for performing manual updates for all the TOE components. Software updates can be downloaded from the Cisco software download site at https://software.cisco.com.A command can be run on the administrator's local workstation to calculate the checksum. All updates for all TOE components are uploaded and deployed via Central Management > Update Manager on the SMC. The section provides guidance on determining the order in which the appliances should be updated and indicates that if the software update fails, an error message is shown.

Component Testing Assurance Activities: The evaluator shall try to perform the update using a legitimate update image without prior authentication as Security Administrator (either by authentication as a user with no administrator privileges or without user authentication at all - depending on the configuration of the TOE). The attempt to update the TOE should fail.

The evaluator shall try to perform the update with prior authentication as Security Administrator using a legitimate update image. This attempt should be successful. This test case should be covered by the tests for FPT_TUD_EXT.1 already.

For the legitimate image update refer to FPT_TUD_EXT.1 where the evaluator tested an update using a valid image. Additionally, FIA_UIA_EXT.1-t1 demonstrates that no services are available prior to login other than the warning banner which is displayed prior to attempting to login.

2.5.2 Management of TSF Data (NDcPP22e:FMT_MTD.1/CoreData)

2.5.2.1 NDcPP22e:FMT_MTD.1.1/CoreData

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that, for each administrative function identified in the guidance documentation; those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.

If the TOE supports handling of X.509v3 certificates and implements a trust store, the evaluator shall examine the TSS to determine that it contains sufficient information to describe how the ability to manage the TOE's trust store is restricted.

Section 6.1, Table 18 (FMT_MTD.1/CoreData) in the ST states that the TOE restricts the access to manage TSF data that can affect the security functions of the TOE to Security Administrators (i.e., administrator roles). The TSF data here includes user accounts and roles, login banner, inactivity timeout values, password length settings, TOE updates, X.509 certificates, audit records, and audit server information. Access to TSF data is disallowed for all non-administrative users because all accounts are administrative accounts.

Each TOE component contains a trust store of X.509v3 certificates. The trust store contains certificates for the local TOE component, and certificates for all other TOE components with which the local component communicates, and certificates for remote syslog servers. The trust store on the SMC additionally contains X.509v3 certificates of remote LDAP servers. Access to trust store data on each component is restricted to authorized administrators only. Remote administrators using the SMC WebUI can manage trust stores on each TOE component via the SMC WebUI. Local 'sysadmin' accounts on each appliance could regenerate a new self-signed X.509v3 device certificate, but doing so would break communication with other TOE components and would result in removing that component from the TOE.

Component Guidance Assurance Activities: The evaluator shall review the guidance documentation to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of the cPP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

If the TOE supports handling of X.509v3 certificates and provides a trust store, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to configure and maintain the trust store in a secure way. If the TOE supports loading of CA certificates, the evaluator shall review the guidance documentation to determine that it provides sufficient information for the administrator to securely load CA certificates into the trust store. The evaluator shall also review the guidance documentation to determine that it explains how to designate a CA certificate a trust anchor.

The ST indicates that only security administrators can login to manage TSF data and configure TOE services. The TSF data manipulating functions and the corresponding configuration information are identified or referenced throughout this AAR with the requirement to which they apply.

Section *Certificates* of the **Admin Guide** describes how administrators can upload certificates to each appliance's trust store, including appliance identity certificates and external CA's. The instructions include certificate requirements that specify the chain must include the root certificate and specific steps for adding all required certificates to the trust store.

Component Testing Assurance Activities: No separate testing for FMT_MTD.1/CoreData is required unless one of the management functions has not already been exercised under any other SFR.

No separate testing for FMT_MTD.1/CoreData.

2.5.3 Management of TSF Data (NDcPP22e:FMT_MTD.1/CryptoKeys)

2.5.3.1 NDcPP22e:FMT_MTD.1.1/CryptoKeys

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: For distributed TOEs see chapter 2.4.1.1.

For non-distributed TOEs, the evaluator shall ensure the TSS lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed.

Section 6 in the ST details how each TOE security function including all security management functions are realized for every TOE component. The SMC component in the TOE manages all TOE components. Section 9 in the ST provides a mapping of the distributed TOE components to the SFRs in the ST. This TOE is a distributed TOE consistent with Use Case 3 as defined in the NDcPP22e.

All security management functions and the corresponding configuration information are identified or referenced throughout this AAR with the requirement to which they apply.

Component Guidance Assurance Activities: For distributed TOEs see chapter 2.4.1.2.

For non-distributed TOEs, the evaluator shall also ensure the Guidance Documentation lists the keys the Security Administrator is able to manage to include the options available (e.g. generating keys, importing keys, modifying keys or deleting keys) and how that how those operations are performed.

This TOE is a distributed TOE consistent with Use Case 3 as defined in the NDcPP22e.

All security management functions and the corresponding configuration information are identified or referenced throughout this AAR with the requirement to which they apply.

Component Testing Assurance Activities: The evaluator shall try to perform at least one of the related actions (modify, delete, generate/import) without prior authentication as security administrator (either by authentication as a non-administrative user, if supported, or without authentication at all). Attempts to perform related actions without prior authentication should fail. According to the implementation no other users than the Security Administrator might be defined and without any user authentication the user might not be able to get to the point where the attempt to manage cryptographic keys can be executed. In that case it shall be demonstrated that access control mechanisms prevent execution up to the step that can be reached without authentication as Security Administrator. The evaluator shall try to perform at least one of the related actions with prior authentication as security administrator. This attempt should be successful.

The operations to modify, delete, generate/import cryptographic keys and certificates are only available to authorized administrators who can login to the TOE. As specified by FIA_UIA_EXT.1, the set of functions available to a user prior to login do not include operations to modify, delete, generate/import cryptographic keys and certificates.

Refer to the results of FIA_UIA_EXT.1 that demonstrate the limited functions available to users prior to login.

The successful management of the crypto keys was performed to successfully configure the trusted channels that were tested for FTP_ITC.1 and FPT_ITT.1.

2.5.4 Specification of Management Functions - per TD0631 (NDcPP22e:FMT_SMF.1)

2.5.4.1 NDcPP22e:FMT_SMF.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The security management functions for FMT_SMF.1 are distributed throughout the cPP and are included as part of the requirements in FTA_SSL_EXT.1, FTA_SSL.3, FTA_TAB.1, FMT_MOF.1(1)/ManualUpdate, FMT_MOF.1(4)/AutoUpdate (if included in the ST), FIA_AFL.1, FIA_X509_EXT.2.2 (if included in the ST), FPT_TUD_EXT.1.2 & FPT_TUD_EXT.2.2 (if included in the ST and if they include an administrator-configurable action), FMT_MOF.1(2)/Services, and FMT_MOF.1(3)/Functions (for all of these SFRs that are included in the ST), FMT_MTD, FPT_TST_EXT, and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FMT_SMF.1.

(containing also requirements on Guidance Documentation and Tests)

The evaluator shall examine the TSS, Guidance Documentation and the TOE as observed during all other testing and shall confirm that the management functions specified in FMT_SMF.1 are provided by the TOE. The evaluator shall confirm that the TSS details which security management functions are available through which interface(s) (local administration interface, remote administration interface).

The evaluator shall examine the TSS and Guidance Documentation to verify they both describe the local administrative interface. The evaluator shall ensure the Guidance Documentation includes appropriate warnings for the administrator to ensure the interface is local.

For distributed TOEs with the option 'ability to configure the interaction between TOE components' the evaluator shall examine that the ways to configure the interaction between TOE components is detailed in the TSS and Guidance Documentation. The evaluator shall check that the TOE behaviour observed during testing of the configured SFRs is as described in the TSS and Guidance Documentation.

See the other requirements in this AAR as referenced. All security management functions and the corresponding configuration information are identified or referenced throughout this AAR with the requirement to which they apply.

Section 6.1, Table 18 (FMT_SMF.1) in the ST states that the TOE includes the functions necessary to administer the TOE locally and remotely. Most TOE administrative actions are performed via the WebUI/GUI of the SMC (SNA Management Console), but each TOE component also has a local serial console interface through which some administrative actions can be performed. Each TOE component has an active WebUI accessible via TLS, but the SMC is the only TOE component that allows login via its WebUI.

The following administrative actions are performed via the SMC WebUI:

- Configuring the pre-login access banner. The banner for each TOE component is configured separately, and each TOE component uses a single banner for its local console and for its remote WebUI.
- Configuring session inactivity time limits. The same idle timeout values apply to the WebUI and console interfaces.
- Management and verification of software updates for each TOE component. (Note: The local console of each appliance can be used to verify the running version, but the local console cannot be used to install updates.)
- Configuration of authentication failure parameters for FIA_AFL.1 including number of successive failed login attempts prior to lockout, and the duration of lockout.
- Configuring audit behavior such as adding/removing syslog servers, and configuring use of TLS for those connections.
- Configuring cryptographic functionality, such as enabling CC mode and FIPS mode, and enabling file integrity checking.
- Configuring interactions between TOE components such as removing/deleting a managed TOE component, or rebooting a TOE component.
- Configuring use of one or more NTP servers, including NTP authentication.

- Configuring the reference identifier for peers such as adding IP addresses for syslog servers, and adding FQDN or IP addresses for LDAP servers.
- Managing the trust store of each TOE component, importing trust anchors, and importing certificates of multiple TOE components, syslog servers, and LDAP servers.

The following administrative actions can be performed via the local console on each appliance:

- Changing administrator passwords for the sysadmin account (the sysadmin account cannot login via the WebUI).
- Verifying the TOE version.

Component Guidance Assurance Activities: See TSS Assurance Activities

See the other requirements in this AAR as referenced. All security management functions and the corresponding configuration information are identified or referenced throughout this AAR with the requirement to which they apply.

Component Testing Assurance Activities: The evaluator tests management functions as part of testing the SFRs identified in section 2.4.4. No separate testing for FMT_SMF.1 is required unless one of the management functions in FMT_SMF.1.1 has not already been exercised under any other SFR.

The evaluator tests management functions as part of testing the SFRs identified in section 2.4.4. No separate testing for FMT_SMF.1 is required unless one of the management functions in FMT_SMF.1.1 has not already been exercised under any other SFR.

2.5.5 Restrictions on Security Roles (NDcPP22e:FMT_SMR.2)

2.5.5.1 NDcPP22e:FMT_SMR.2.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.5.5.2 NDcPP22e:FMT_SMR.2.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.5.5.3 NDcPP22e:FMT_SMR.2.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that it details the TOE supported roles and any restrictions of the roles involving administration of the TOE.

Section 6.1, Table 18 (FMT_SMR.2) in the ST states that the TOE platform maintains both privileged and semi-privileged administrator roles. The terms "Authorized Administrator" and "Security Administrator" are used interchangeable in this ST to refer to any user that has been assigned to a privilege level that is permitted to perform the relevant action; therefore, has the appropriate privileges to perform the requested functions. The assigned role determines the functions the user can perform; hence the authorized administrator with the appropriate privileges.

Component Guidance Assurance Activities: The evaluator shall review the guidance documentation to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration.

Section *Secure Configuration* in the **Admin Guide** describes how Central Management through the Web App is used to configure all of the TOE appliances. The *User Management* section of the **Admin Guide** defines how to create a user with the master admin role to access the SMC through the SNA Web App. The *LDAP* section provides an example with the full set of steps for creating a user with the master admin role. Section *Local Authentication* describes how to log on to the local console of each TOE appliance for administration using the sysadmin account. User accounts that are associated with the master admin role or the sysadmin account are considered to fulfill the TOE Security Administrator role.

Component Testing Assurance Activities: In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this cPP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.

All administrative interfaces were used throughout the course of testing including local console for all physical and virtual devices along with WebUI for the SMC devices. The WebUI is only available to the SMC devices. Access to each of the TOE interfaces was demonstrated in FIA_UIA_EXT.1 test 1.

2.6 Protection of the TSF (FPT)

2.6.1 Protection of Administrator Passwords (NDcPP22e:FPT_APW_EXT.1)

2.6.1.1 NDcPP22e:FPT_APW_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.1.2 NDcPP22e:FPT_APW_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.

Section 6.1, Table 18 (FPT_APW_EXT.1) in the ST states that the TOE ensures that plaintext user passwords will not be disclosed, even to administrators. There are no administrative interfaces that allow viewing of stored passwords; there is no viewable configuration file and no viewable password file; when changing passwords the old password is not displayed in any form (plaintext or otherwise) and the new password is obscured with asterisks as it’s being typed; passwords are obscured by asterisks during login via WebUI and are not echoed at all (the cursor at the password prompt does not move) when logging via the console; passwords are not written to audit records.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.6.2 Basic internal TSF data transfer protection (NDcPP22e:FPT_ITT.1)

2.6.2.1 NDcPP22e:FPT_ITT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: If the TOE is not a distributed TOE, then no evaluator action is necessary. For a distributed TOE the evaluator carries out the activities below.

The evaluator shall examine the TSS to determine that, for all communications between components of a distributed TOE, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS for these inter-component communications are specified and included in the requirements in the ST.

Section 6.1, Table 18 (FPT_ITT.1) in the ST states that the distributed components of the TOE communicate with each other using TLSv1.2. The SMC (SNA Management Console) communicates with each other appliance (FC, FS, and UDPD), and the TLS sessions can be initiated in either direction. During most TOE activity the managed components (FC, FS, and UDPD) initiate the mutually-authenticated TLS (FCS_TLSC_EXT.2) connections to the SMC (FCS_TLSS_EXT.2) to query the SMC for any updates (configuration updates and software updates). The SMC will initiate mutually-authenticated TLS connections (FCS_TLSC_EXT.2) with other components (FCS_TLSS_EXT.2) when an SMC administrator initiates a query for data stored on another component (e.g. to query for network flow data) for the purpose of displaying query results/reports to the SMC administrator.

Component Guidance Assurance Activities: If the TOE is not a distributed TOE then no evaluator action is necessary. For a distributed TOE the evaluator carries out the activities below.

The evaluator shall confirm that the guidance documentation contains instructions for establishing the relevant allowed communication channels and protocols between each pair of authorized TOE components, and that it contains recovery instructions should a connection be unintentionally broken.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode after the certificates are updated, to ensure that only compliant algorithms are used.

Section *Secure Installation* in the **Admin Guide** provides general guidance for the initial configuration of the appliances, the registration process and specific guidance for the evaluated configuration. This section also refers to the SNA Installation and Configuration Guides*, "Configuring the Managed System"* and *"Finishing Appliance Configurations"* sections for detailed instructions including the configuration order of the appliances. During the initial configuration process the administrator is asked whether the new appliance will be managed by an SMC, and after answering affirmatively, that administrator is prompted to trust the SMC certificate, provide the IP address or hostname of the SMC and a valid SMC administrator username and password to authenticate to the SMC. If authentication succeeds, the SMC and the joining appliance exchange certificates, and subsequent TLS connections between the TOE components are authenticated using X.509 certificate-based authentication.

The *Configuration for Common Criteria Compliance* section of the **Admin Guide** indicates that the first step in configuring the TOE for compliance is to complete all the certificates requirements and procedures. Section *Certificates* of the **Admin Guide** indicates that the communication between TOE appliances is authenticated using x509v3 certificates. This section defines the best practices and procedures for replacing the certificates of the TOE appliances to update the initial self-signed certificates that were used as part of registration, thus increasing the security of the distributed TOE channel.

Section *Secure Installation* in the **Admin Guide** describes how to disable communication between distributed components. This can be initiated from the SMC or from a managed appliance and will result in each TOE component deleting the device association from its local configuration. After an appliance is removed, no further TSF data will be exchanged between the appliances without completing a new registration process. This section also provides recovery instructions if a connection fails between the SMC and the managed appliances, the connections will automatically retry and restore connectivity. Connections will always be temporarily unavailable when appliances are rebooting.

Component Testing Assurance Activities: If the TOE is not a distributed TOE then no evaluator action is necessary. For a distributed TOE the evaluator carries out the activities below.

The evaluator shall perform the following tests:

a) Test 1: The evaluator shall ensure that communications using each protocol between each pair of authorized TOE components is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

b) Test 2: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.

c) Test3: Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route between distributed components.

The evaluator shall ensure that, for each different pair of non-equivalent component types, the connection is physically interrupted for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) a duration that is shorter than the application layer timeout but is of sufficient length to interrupt the network link layer.

The evaluator shall ensure that when physical connectivity is restored, either communications are appropriately protected, or the secure channel is terminated and the registration process (as described in the FTP_TRP.1/Join) re-initiated, with the TOE generating adequate warnings to alert the Security Administrator.

In the case that the TOE is able to detect when the cable is removed from the device, another physical network device (e.g. a core switch) shall be used to interrupt the connection between the components. The interruption shall not be performed at the virtual node (e.g. virtual switch) and must be physical in nature.

Further assurance activities are associated with the specific protocols.

Test 1 - The TLS communication between the distributed TOE components is enabled by default after the registration process completes. The registration process was demonstrated as part of FCO_CPC_EXT.1 and FTP_TRP.1/Join. The successful connections for the TLS channel are demonstrated in Test 3 below.

Test 2 – This test is demonstrated in Test 3 below.

Test 3 – The evaluator registered the TOE components so that the TLS ITT communication was enabled. The evaluator then physically disrupted the TLS connections by temporarily unplugging a cable between two in-line switches. The evaluator observed that the channels remained secure during and after the disruption. The evaluator physically disconnected a switch between the TOE components for approximately 20-30 seconds. The TOE continued to use the existing connection. The evaluator physically disconnected a switch between the TOE components for approximately 10 minutes. The TOE then renegotiated a new connection. This test was repeated between the SMC and each of the other TOE devices.

Note that when the connection between TOE components is disrupted the evaluator observed that the central manager alerts the administrator by stating that the channel is down. Additionally, the FPT_ITT.1 audits captured show when the connections are terminated and established.

2.6.3 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) (NDcPP22e:FPT_SKP_EXT.1)

2.6.3.1 NDcPP22e:FPT_SKP_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that it details how any preshared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

Section 6.1, Table 18 (FPT_SKP_EXT.1) in the ST states that the TOE stores all private keys (associated with local X.509v3 certificates), and pre-shared/symmetric keys (associated with LDAP servers) such that they are not readable by administrators. The keys are stored in plaintext but there are no administrative interfaces that allow viewing of any stored keys.

Component Guidance Assurance Activities: None Defined

Component Testing Assurance Activities: None Defined

2.6.4 Reliable Time Stamps - per TD0632 (NDcPP22e:FPT_STM_EXT.1)

2.6.4.1 NDcPP22e:FPT_STM_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.4.2 NDcPP22e:FPT_STM_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time, and that it provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

If 'obtain time from the underlying virtualization system' is selected, the evaluator shall examine the TSS to ensure that it identifies the VS interface the TOE uses to obtain time. If there is a delay between updates to the time on the VS and updating the time on the TOE, the TSS shall identify the maximum possible delay.

Section 6.1, Table 18 (FPT_STM_EXT.1) in the ST states that the TOE provides a source of date and time information in the form of a software clock, which it uses for writing timestamps to audit messages (FAU_GEN.1), for tracking session idle time (FTA_SSL_EXT.1 and FTA_SSL.3), for unlocking accounts after consecutive failed login attempts (FIA_AFL.1), and for verifying validity times of X.509v3 certificates (FIA_X509_EXT.1).

When each TOE component boots, it will set its software clock to match the hardware clock of the underlying server. All SNA appliances operate in the UTC time zone and the time zone is not configurable, so the hardware clock of the underlying system must also be set to UTC. The software clock on each TOE component can be manually set by the 'sysadmin' account via the console, or can be configured via the WebUI to use one or more NTP servers.

Component Guidance Assurance Activities: The evaluator examines the guidance documentation to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the guidance documentation instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

If the TOE supports obtaining time from the underlying VS, the evaluator shall verify the Guidance Documentation specifies any configuration steps necessary. If no configuration is necessary, no statement is necessary in the Guidance Documentation. If there is a delay between updates to the time on the VS and updating the time on the TOE, the evaluator shall ensure the Guidance Documentation informs the administrator of the maximum possible delay.

Section *NTP Communication* of the **Admin Guide** describes the steps to disable NTP communication and allow manual setting of the time by an administrator on each of the TOE components.

Component Testing Assurance Activities: The evaluator shall perform the following tests:

a) Test 1: If the TOE supports direct setting of the time by the Security Administrator then the evaluator uses the guidance documentation to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

b) Test 2: If the TOE supports the use of an NTP server; the evaluator shall use the guidance documentation to configure the NTP client on the TOE, and set up a communication path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the guidance documentation.

If the audit component of the TOE consists of several parts with independent time information, then the evaluator shall verify that the time information between the different parts are either synchronized or that it is possible for all audit information to relate the time information of the different part to one base information unambiguously.

c) Test 3: [conditional] If the TOE obtains time from the underlying VS, the evaluator shall record the time on the TOE, modify the time on the underlying VS, and verify the modified time is reflected by the TOE. If there is a delay between the setting the time on the VS and when the time is reflected on the TOE, the evaluator shall ensure this delay is consistent with the TSS and Guidance.

Test 1 – The use of NTP is the default configuration, so the evaluator had to enable the use of manual time configuration via the local console. Upon reboot of the TOE, the evaluator was able to manually set the time via the local console and observed that the time change was successful.

Test 2 – Refer to FCS_NTP_EXT.1 where the evaluator followed the guidance documentation to configure an NTP client on the TOE and observed the TOE correctly responds to the NTP server.

Test 2 – Not applicable. TOE does not obtain the time from the underlying VS.

2.6.5 TSF TESTING (NDcPP22e:FPT_TST_EXT.1)

2.6.5.1 NDcPP22e:FPT_TST_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF; this description should include an outline of what the tests are actually doing (e.g., rather than saying 'memory is tested', a description similar to 'memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written' shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

For distributed TOEs the evaluator shall examine the TSS to ensure that it details which TOE component performs which self-tests and when these self-tests are run.

Section 6.1, Table 18 (FPT_TST_EXT.1) in the ST states that the TOE components include a number of built in diagnostic tests that are run during start-up to determine whether the TOE is operating properly.

The crypto module within the TOE runs power-on self-tests including known answer tests (KATs). The following self-tests are executed: AES encryption/decryption KAT, RSA pairwise consistency and sign/verify KAT, SHA hash KATs, HMAC-SHA hash KATs, and DRBG KATs. When any crypto-dependent service (such as the TLS web server) initiates startup of the crypto module and any self-test fails during the startup of the crypto module the crypto module returns an error to the service and the service will fail to start. The crypto module will not enter an operational state if any of its self-tests fail, thus the requesting service will also fail to start, and the TOE will not enter a fully operational state. The TOE will remain in a state where all crypto-dependent services remain disabled (all TLS services: remote administration, inter-TSF communication, syslog-over-TLS, and LDAP-over-TLS), so the only interactive interface available will be via the local serial console for troubleshooting purposes. When a crypto-dependent service initiates startup of the crypto module and all cryptographic self-tests pass during the startup of the crypto module the crypto module returns a successful response to the service and the services continue to load.

The TOE runs SHA-256 integrity tests daily to verify the integrity of the kernel, and all binaries and libraries. If the hash verification fails, the TOE will generate an audit message indicating which file changed.

The virtual appliance forms of TOE components perform the same self-tests as the physical appliances. These tests (periodically reverifying the integrity of the software files, periodically reverifying the permissions of those files, and verifying the correct operation of cryptographic operations prior to the TOE becoming operational) are sufficient to demonstrate that the TSF is operating correctly.

Component Guidance Assurance Activities: The evaluator shall also ensure that the guidance documentation describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

For distributed TOEs the evaluator shall ensure that the guidance documentation describes how to determine from an error message returned which TOE component has failed the self-test.

Section *Common Criteria* in the **Admin Guide** states that if the cryptographic library fails any of its start-up self-tests, the appliance will fail to boot completely and all TLS-dependent processes (TLS server and TLS client) will remain inactive. If the failure occurs on the SMC, the Web UI remains disabled. If the failure occurs on another appliance, its status displays in Central Management as a communication failure. The user is advised to log in through the console on the appliance that has failed to troubleshoot, then contact Cisco Support.

Section *AIDE* in the **Admin Guide** states that the Advanced Intrusion Detection Environment (AIDE) is a host base-lining system that detects modifications of critical files on a system. When it is enabled, AIDE runs an audit of the current system once a day. It compares the hash sum and permissions of each monitored file on the current file system against the values stored in the appliance database. Each SNA appliance runs AIDE independently and generates audit log messages separately. These messages can be viewed through Central Management for each appliance. Each appliance is uniquely identified by its hostname as recorded in the messages transmitted to the syslog servers. When the daily AIDE job completes, an audit log message that indicates whether any changes were detected is recorded in the audit log. If changes are detected, the guide includes a list of file types that do not apply to Common Criteria. If there are other file types on the list, the user is instructed to contact Cisco SNA Support to determine whether the file changes are impacting CC security functionality.

Component Testing Assurance Activities: It is expected that at least the following tests are performed:

a) Verification of the integrity of the firmware and executable software of the TOE

b) Verification of the correct operation of the cryptographic functions necessary to fulfill any of the SFRs.

Although formal compliance is not mandated, the self-tests performed should aim for a level of confidence comparable to:

a) FIPS 140-2, chap. 4.9.1, Software/firmware integrity test for the verification of the integrity of the firmware and executable software. Note that the testing is not restricted to the cryptographic functions of the TOE.

b) FIPS 140-2, chap. 4.9.1, Cryptographic algorithm test for the verification of the correct operation of cryptographic functions. Alternatively, national requirements of any CCRA member state for the security evaluation of cryptographic functions should be considered as appropriate.

The evaluator shall either verify that the self tests described above are carried out during initial start-up or that the developer has justified any deviation from this.

For distributed TOEs the evaluator shall perform testing of self-tests on all TOE components according to the description in the TSS about which self-test are performed by which component.

The TOE performs a suite of crypto self-tests at startup. These tests have been described in the ST. While the TOE does not actively provide feedback when the self-tests have been performed, the self-tests have been addressed in the firmware modules FIPS compliance letter.

The TOE runs SHA-256 integrity tests daily to verify the integrity of the kernel, and all binaries and libraries. If the hash verification fails, the TOE will generate an audit message indicating which file changed. In order for this to occur the evaluator had to enable the AIDE functionality on each appliance as described in guidance. For each appliance, the evaluator viewed that an audit was generated to determine any changes to the file system since the baseline was created. As the integrity checking verifies all files and file permissions, some changes were detected, but the evaluator verified via guidance that the changed file types were expected and acceptable.

2.6.6 Trusted update (NDcPP22e:FPT_TUD_EXT.1)

2.6.6.1 NDcPP22e:FPT_TUD_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.6.2 NDcPP22e:FPT_TUD_EXT.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.6.6.3 NDcPP22e:FPT_TUD_EXT.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall verify that the TSS describe how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the TSS needs to describe how and when the inactive version becomes active. The evaluator shall verify this description.

The evaluator shall verify that the TSS describes all TSF software update mechanisms for updating the system firmware and software (for simplicity the term 'software' will be used in the following although the requirements apply to firmware and software). The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. Alternatively an approach using a published hash can be used. In this case the TSS shall detail this mechanism instead of the digital signature verification mechanism. The evaluator shall verify that the TSS describes the method by which the digital signature or published hash is verified to include how the candidate updates are obtained, the processing associated with verifying the digital signature or published hash of the update, and the actions that take place for both successful and unsuccessful signature verification or published hash verification.

If the options 'support automatic checking for updates' or 'support automatic updates' are chosen from the selection in FPT_TUD_EXT.1.2, the evaluator shall verify that the TSS explains what actions are involved in automatic checking or automatic updating by the TOE, respectively.

For distributed TOEs, the evaluator shall examine the TSS to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component. Alternatively, this description can be provided in the guidance documentation. In that case the evaluator should examine the guidance documentation instead.

If a published hash is used to protect the trusted update mechanism, then the evaluator shall verify that the trusted update mechanism does involve an active authorization step of the Security Administrator, and that download of the published hash value, hash comparison and update is not a fully automated process involving no active authorization by the Security Administrator. In particular, authentication as Security Administration according to FMT_MOF.1/ManualUpdate needs to be part of the update process when using published hashes.

Section 6.1, Table 18 (FPT_TUD_EXT.1) in the ST states that an Authorized Administrator can query the software version running on the TOE (on each TOE component) and can initiate software updates to each TOE component. The software version can be queried in multiple ways: the version running on every TOE component can be queried via the SMC WebUI, and the console interface of each individual TOE component can be used to view the running version on that component.

When software updates are made available by Cisco, an administrator can download the update from Cisco to a local administrative workstation, then upload the software updates to the SMC. Uploading a software update to SMC will not automatically initiate installation of the update. Updates for all TOE components are first uploaded to the SMC by an SMC administrator via the SMC WebUI/GUI then using the same interface the SMC administrator initiates installation of the software updates to each TOE component until all TOE components have been updated to the same version. Updates for all TOE components other than SMC itself (FC, FS, UDPD) are transferred from the SMC to other TOE components over the same TLS channel that's used for all other FPT_ITT.1 communications. When an administrator initiates the update process from SMC, each appliance will use digital signature verification to ensure the integrity of the software update prior to installation of the update.

Component Guidance Assurance Activities: The evaluator shall verify that the guidance documentation describes how to query the currently active version. If a trusted update can be installed on the TOE with a delayed activation, the guidance documentation needs to describe how to query the loaded but inactive version.

The evaluator shall verify that the guidance documentation describes how the verification of the authenticity of the update is performed (digital signature verification or verification of published hash). The description shall include the procedures for successful and unsuccessful verification. The description shall correspond to the description in the TSS.

If a published hash is used to protect the trusted update mechanism, the evaluator shall verify that the guidance documentation describes how the Security Administrator can obtain authentic published hash values for the updates.

For distributed TOEs the evaluator shall verify that the guidance documentation describes how the versions of individual TOE components are determined for FPT_TUD_EXT.1, how all TOE components are updated, and the error conditions that may arise from checking or applying the update (e.g. failure of signature verification, or exceeding available storage space) along with appropriate recovery actions. The guidance documentation only has to describe the procedures relevant for the Security Administrator; it does not need to give information about the internal communication that takes place when applying updates.

If this was information was not provided in the TSS: For distributed TOEs, the evaluator shall examine the Guidance Documentation to ensure that it describes how all TOE components are updated, that it describes all mechanisms that support continuous proper functioning of the TOE during update (when applying updates separately to individual TOE components) and how verification of the signature or checksum is performed for each TOE component.

If this was information was not provided in the TSS: If the ST author indicates that a certificate-based mechanism is used for software update digital signature verification, the evaluator shall verify that the Guidance Documentation contains a description of how the certificates are contained on the device. The evaluator also ensures that the Guidance Documentation describes how the certificates are installed/updated/selected, if necessary.

Section *Product Updates* of the **Admin Guide** describes how the admin can view the installed version of the software of each of the TOE components through Central Management > Update Manager. The Update Manager also shows the last reboot, version ready to install, and the update status. The section further describes the steps for performing manual updates for all the TOE components. Software updates for each TOE component can be downloaded from the SN Cisco software download site at https://software.cisco.com. A command can be run on the administrator's local workstation to calculate the checksum. If the result does not match the checksum, the admin is directed to attempt to download the file and calculate the checksum again. If the results still do not match, the admin should contact Cisco SNA Support.

Updates for all TOE components are uploaded and deployed via Central Management > Update Manager on the SMC. The *Product Updates* section provides guidance on determining the order in which the appliances should be updated and it describes the possible error conditions that could arise during the upload and installation of the update files as well as troubleshooting steps for a failed installation.

Component Testing Assurance Activities: The evaluator shall perform the following tests:

a) Test 1: The evaluator performs the version verification activity to determine the current version of the product. If a trusted update can be installed on the TOE with a delayed activation, the evaluator shall also query the most recently installed version (for this test the TOE shall be in a state where these two versions match). The evaluator obtains a legitimate update using procedures described in the guidance documentation and verifies that it is successfully installed on the TOE. For some TOEs loading the update onto the TOE and activation of the update are separate steps ('activation' could be performed e.g. by a distinct activation step or by rebooting the device). In that case the evaluator verifies after loading the update onto the TOE but before activation of the update that the current version of the product did not change but the most recently installed version has changed to the new product version. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update and that current version of the product and most recently installed version match again.

b) Test 2 [conditional]: If the TOE itself verifies a digital signature to authorize the installation of an image to update the TOE the following test shall be performed (otherwise the test shall be omitted). The evaluator first confirms that no updates are pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test. The evaluator obtains or produces illegitimate updates as defined below, and attempts to install them on the TOE. The evaluator verifies that the TOE rejects all of the illegitimate updates. The evaluator performs this test using all of the following forms of illegitimate updates:

1) A modified version (e.g. using a hex editor) of a legitimately signed update

2) An image that has not been signed

3) An image signed with an invalid signature (e.g. by using a different key as expected for creating the signature or by manual modification of a legitimate signature)

4) If the TOE allows a delayed activation of updates the TOE must be able to display both the currently executing version and most recently installed version. The handling of version information of the most recently installed version might differ between different TOEs depending on the point in time when an attempted update is rejected. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify, that both, current version and most recently installed version, reflect the same version information as prior to the update attempt.

c) Test 3 [conditional]: If the TOE itself verifies a hash value over an image against a published hash value (i.e. reference value) that has been imported to the TOE from outside such that the TOE itself authorizes the installation of an image to update the TOE, the following test shall be performed (otherwise the test shall be omitted). If the published hash is provided to the TOE by the Security Administrator and the verification of the hash value over the update file(s) against the published hash is performed by the TOE, then the evaluator shall perform the following tests. The evaluator first confirms that no update is pending and then performs the version verification activity to determine the current version of the product, verifying that it is different from the version claimed in the update(s) to be used in this test.

1) The evaluator obtains or produces an illegitimate update such that the hash of the update does not match the published hash. The evaluator provides the published hash value to the TOE and calculates the hash of the update either on the TOE itself (if that functionality is provided by the TOE), or else outside the TOE. The evaluator confirms that the hash values are different, and attempts to install the update on the TOE, verifying that this fails because of the difference in hash values (and that the failure is logged). Depending on the implementation of the TOE, the TOE might not allow the Security Administrator to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE.

2) The evaluator uses a legitimate update and tries to perform verification of the hash value without providing the published hash value to the TOE. The evaluator confirms that this attempt fails. Depending on the implementation of the TOE it might not be possible to attempt the verification of the hash value without providing a hash value to the TOE, e.g. if the hash value needs to be handed over to the TOE as a parameter in a command line message and the syntax check of the command prevents the execution of the command without providing a hash value. In that case the mechanism that prevents the execution of this check shall be tested accordingly, e.g. that the syntax check rejects the command without providing a hash value, and the rejection of the attempt is regarded as sufficient verification of the correct behaviour of the TOE in failing to verify the hash. The evaluator then attempts to install the update on the TOE (in spite of the unsuccessful hash verification) and confirms that this fails. Depending on the implementation of the TOE, the TOE might not allow to even attempt updating the TOE after the verification of the hash value fails. In that case the verification that the hash comparison fails is regarded as sufficient verification of the correct behaviour of the TOE.

3) If the TOE allows delayed activation of updates, the TOE must be able to display both the currently executing version and most recently installed version. The handling of version information of the most recently installed version might differ between different TOEs. Depending on the point in time when the attempted update is

rejected, the most recently installed version might or might not be updated. The evaluator shall verify that the TOE handles the most recently installed version information for that case as described in the guidance documentation. After the TOE has rejected the update the evaluator shall verify, that both, current version and most recently installed version, reflect the same version information as prior to the update attempt.

If the verification of the hash value over the update file(s) against the published hash is not performed by the TOE, Test 3 shall be skipped.

The evaluator shall perform Test 1, Test 2 and Test 3 (if applicable) for all methods supported (manual updates, automatic checking for updates, automatic updates).

For distributed TOEs the evaluator shall perform Test 1, Test 2 and Test 3 (if applicable) for all TOE components.

Test 1 – The evaluator viewed the version of the TOE devices from the SMC and then installed a valid update. The signature verified and the update was successfully installed. The evaluator displayed the version again and confirmed that the new version was displayed. This test was performed for each TOE component.

Test 2 – The evaluator first queried the TOE version and then attempted to load and install each of the invalid updates. The updates failed to load. The evaluator further confirmed this by querying the TOE version again and confirming that it was the same version as prior to the update attempt. This test was performed for each TOE component.

Test 3 – Not applicable. The TOE does not support published hash.

2.7 TOE access (FTA)

2.7.1 TSF-initiated Termination (NDcPP22e:FTA_SSL.3)

2.7.1.1 NDcPP22e:FTA_SSL.3.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that it details the administrative remote session termination and the related inactivity time period.

Section 6.1, Table 18 (FTA_SSL.3) in the ST states that the TOE administrators can configure maximum inactivity times individually for both local and remote administrative sessions. These settings are not immediately activated for the current session, changes to inactivity limits are enforced for new sessions.

If a local user session is inactive for a configured period of time, the session will be terminated, and re-authentication is required to start a new session. If a remote user session is inactive for the configured idle time limit, the session will be terminated and will require authentication to establish a new session.

The configurable idle timeout value is the same for local and remote sessions, and configured individually for each appliance via the SMC GUI: Central Management > Appliance Manager > General > Protected Sessions Time-Out. The configurable inactivity timeout range for local console sessions and remote sessions is from 2 to 1440 minutes.

Component Guidance Assurance Activities: The evaluator shall confirm that the guidance documentation includes instructions for configuring the inactivity time period for remote administrative session termination.

Section *Protected Sessions Time-Out* of the **Admin Guide** defines how to configure the time threshold for when a user is logged out for user activity including for the Web App on the SMC and local access on all appliances. The default setting is 0 which means the administrator session never expires but this default value should be updated in the CC configuration. The TOE supports a range of 2 to 1440 minutes.

Component Testing Assurance Activities: For each method of remote administration, the evaluator shall perform the following test:

a) Test 1: The evaluator follows the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a remote interactive session with the TOE. The evaluator then observes that the session is terminated after the configured time period.

On the SMC Web UI, the evaluator alternately configured 3- and 5-minute timeouts and then observed that the session timed out after the configured timeout period.

2.7.2 User-initiated Termination (NDcPP22e:FTA_SSL.4)

2.7.2.1 NDcPP22e:FTA_SSL.4.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that it details how the local and remote administrative sessions are terminated.

Section 6.1, Table 18 (FTA_SSL.4) in the ST states that the TOE Administrators are able to logout of all local and remote administrative sessions. From the local console, the administrator selects "Exit". From the GUI, the administrator clicks "Logout".

Component Guidance Assurance Activities: The evaluator shall confirm that the guidance documentation states how to terminate a local or remote interactive session.

Section *Terminating User Sessions* of the **Admin Guide** describes how to terminate both a session with the Web App on the SMC and local sessions on all appliances.

Component Testing Assurance Activities: For each method of remote administration, the evaluator shall perform the following tests:

a) Test 1: The evaluator initiates an interactive local session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated.

b) Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the guidance documentation to exit or log off the session and observes that the session has been terminated.

Test 1 – The evaluator logged out of the local console and observed that the session was terminated.

Test 2 - The evaluator logged out of the remote interactive session via the Web UI and observed that the session was terminated.

2.7.3 TSF-initiated Session Locking (NDcPP22e:FTA_SSL_EXT.1)

2.7.3.1 NDcPP22e:FTA_SSL_EXT.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that it details whether local administrative session locking or termination is supported and the related inactivity time period settings.

Section 6.1, Table 18 (FTA_SSL_EXT.1) in the ST states that TOE administrators can configure maximum inactivity times individually for both local and remote administrative sessions. These settings are not immediately activated for the current session, changes to inactivity limits are enforced for new sessions.

If a local user session is inactive for a configured period of time, the session will be terminated, and re-authentication is required to start a new session. If a remote user session is inactive for the configured idle time limit, the session will be terminated and will require authentication to establish a new session.

The configurable idle timeout value is the same for local and remote sessions, and configured individually for each appliance via the SMC GUI: Central Management > Appliance Manager > General > Protected Sessions Time-Out. The configurable inactivity timeout range for local console sessions and remote sessions is from 2 to 1440 minutes.

Component Guidance Assurance Activities: The evaluator shall confirm that the guidance documentation states whether local administrative session locking or termination is supported and instructions for configuring the inactivity time period.

Section *Protected Sessions Time-Out* of the **Admin Guide** defines how to configure the time threshold for when a user is logged out for user activity including for the Web App on the SMC and local access on all appliances. The default setting is 0 which means the administrator session never expires but this default value should be updated in the CC configuration. The TOE supports a range of 2 to 1440 minutes.

Component Testing Assurance Activities: The evaluator shall perform the following test:

a) Test 1: The evaluator follows the guidance documentation to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that reauthentication is needed when trying to unlock the session.

The evaluator set the console inactivity time period to 3 minutes, logged onto the TOE and made note of the time and verified that the devices timed out after 3 minutes as expected. The evaluator then repeated this with an inactivity time period of 5 minutes.

2.7.4 Default TOE Access Banners (NDcPP22e:FTA_TAB.1)

2.7.4.1 NDcPP22e:FTA_TAB.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall check the TSS to ensure that it details each administrative method of access (local and remote) available to the Security Administrator (e.g., serial port, SSH, HTTPS). The evaluator shall check the TSS to ensure that all administrative methods of access available to the Security Administrator are listed and that the TSS states that the TOE is displaying an advisory notice and a consent warning message for each administrative method of access. The advisory notice and the consent warning message might be different for different administrative methods of access, and might be configured during initial configuration (e.g. via configuration file).

Section 6.1, Table 18 (FTA_TAB.1) in the ST states that the TOE displays a privileged Administrator specified banner on the CLI management interface and on the WebUI prior to allowing any administrative access to the TOE. The banner configured for each appliance (all configured via the SMC WebUI) is displayed prior to login at that appliance's local console and prior to login at that appliance's WebUI.

Note that once the managed appliances (FC, FS, or UDPD) are in their CC evaluated configuration, authentication to each of their WebUI is effectively disabled since the local 'admin' account on those appliances is disabled and no remote (LDAP) authentication will be enabled, but the login banner continues to be displayed prior displaying the login prompt, and failed authentication attempts are logged.

Component Guidance Assurance Activities: The evaluator shall check the guidance documentation to ensure that it describes how to configure the banner message.

Section *Login Banners* in the **Admin Guide** describes the configuration to display a warning banner on all TOE appliances for all user interfaces. The message can be in any language and contain any supported UNICODE character other than the single quote (') or arrows (< >) symbols. The message can be up to 120 characters long.

Component Testing Assurance Activities: The evaluator shall also perform the following test:

a) Test 1: The evaluator follows the guidance documentation to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

The evaluator followed the guidance documentation to set the login banner and confirmed that the login banner was displayed via the Web UI and the local console.

2.8 Trusted path/channels (FTP)

2.8.1 Inter-TSF trusted channel (NDcPP22e:FTP_ITC.1)

2.8.1.1 NDcPP22e:FTP_ITC.1.1

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.1.2 NDcPP22e:FTP_ITC.1.2

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.1.3 NDcPP22e:FTP_ITC.1.3

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each secure communication mechanism is identified in terms of the allowed protocols for that IT entity, whether the TOE acts as a server or a client, and the method of assured identification of the non-TSF endpoint. The evaluator shall also confirm that all secure communication mechanisms are described in sufficient detail to allow the evaluator to match them to the cryptographic protocol Security Functional Requirements listed in the ST.

Section 6.1, Table 18 (FTP_ITC.1) in the ST states that the TOE protects communications between itself and remote audit servers using TLS. This provides a secure channel to transmit the syslog messages. The TOE protects communications between itself and AAA (LDAP) servers using TLS.

Section 6.1, Table 18 (FCS_TLSC_EXT.1) in the ST indicates that the TOE uses TLS clients for transmitting syslog over TLS and for connecting to LDAP servers over TLS.

Component Guidance Assurance Activities: The evaluator shall confirm that the guidance documentation contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used for any TLS connection, including connections to LDAP, and syslog.

Section *Review External Service Certificate Requirements* of the **Admin Guide** describes the requirements for certificates of external services such as a remote syslog and LDAP. Section *Add External Service Certificates to the Trust Stores* describes how to add the relevant certificates for external services, including a remote syslog and LDAP, to the appliances trust store. This section states that the Root CA of the syslog server should be added to all of the SNA components.

Section *Audit Log Destination* of the **Admin Guide** describes the configuration process for enabling sending audits to a remote syslog server over TLS from each of the TOE appliances.

Section *LDAP* of the **Admin Guide** describes the configuration process for enabling LDAP authentication over TLS on the SMC.

Sections *Troubleshooting LDAP Connectivity* and *Troubleshooting Syslog Connectivity* describe what actions an administrator should take in the event the connection cannot be established during the validity check of a certificate.

Component Testing Assurance Activities: The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public-facing document or report.

The evaluator shall perform the following tests:

a) Test 1: The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

b) Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be initiated from the TOE.

c) Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.

d) Test 4: Objective: The objective of this test is to ensure that the TOE reacts appropriately to any connection outage or interruption of the route to the external IT entities.

The evaluator shall, for each instance where the TOE acts as a client utilizing a secure communication mechanism with a distinct IT entity, physically interrupt the connection of that IT entity for the following durations: i) a duration that exceeds the TOE's application layer timeout setting, ii) a duration shorter than the application layer timeout but of sufficient length to interrupt the network link layer.

The evaluator shall ensure that, when the physical connectivity is restored, communications are appropriately protected and no TSF data is sent in plaintext.

In the case where the TOE is able to detect when the cable is removed from the device, another physical network device (e.g. a core switch) shall be used to interrupt the connection between the TOE and the distinct IT entity. The interruption shall not be performed at the virtual node (e.g. virtual switch) and must be physical in nature.

Further assurance activities are associated with the specific protocols.

For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of external secure channels to TOE components in the Security Target.

The developer shall provide to the evaluator application layer configuration settings for all secure communication mechanisms specified by the FTP_ITC.1 requirement. This information should be sufficiently detailed to allow the evaluator to determine the application layer timeout settings for each cryptographic protocol. There is no expectation that this information must be recorded in any public- facing document or report.

Test 1 – The evaluator first configured communication to the LDAP server using TLS. The successful authentication of the LDAP user was demonstrated as part of FIA_UIA_EXT.1 test 1. The successful initiation of the encrypted TLS LDAP communication is demonstrated in Test 4 below. Next, the evaluator configured the TLS communication of audits to the remote syslog server. The successful initiation of the encrypted TLS Syslog communication is demonstrated in Test 4 below. The evaluator viewed that syslog messages were in fact being sent to the syslog server.

Test 2 - This test case was demonstrated as part of test 4.

Test 3 - This test case was demonstrated as part of test 4.

Test 4 – Syslog: The evaluator configured TLS communication between the TOE and a syslog server. The evaluator then physically disrupted the TLS connections by temporarily unplugging a cable between two in-line switches. The evaluator observed that the channels remained secure during and after the disruption. The evaluator physically disconnected a switch between the TOE components for approximately 30 seconds. The TOE continued to use the existing connection. The evaluator physically disconnected a switch between the TOE components for at least approximately 15 minutes. The TOE then renegotiated a new connection.

LDAP: The evaluator physically disconnected a switch between the LDAP server and the TOE for approximately 30 seconds. Due to the nature of the LDAP connection, being a near instant auth request and response instead of a continuous connection such as syslog demonstrated above, it was not possible for the physical disruption to occur during the TLS session. Therefore, despite the disruption being short enough for only a Mac Layer timeout, the TOE initiated a new TLS session post reconnection. The data was still encrypted and not sent in plaintext. Next, the evaluator physically disconnected a switch between the LDAP server and the TOE for at least approximately 15 minutes. The TOE renegotiates a new TLS connection upon the return of the physical connection.

2.8.2 Trusted Path (NDcPP22e:FTP_TRP.1/Admin)

2.8.2.1 NDcPP22e:FTP_TRP.1.1/Admin

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.2.2 NDcPP22e:FTP_TRP.1.2/Admin

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.2.3 NDcPP22e:FTP_TRP.1.3/Admin

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

Section 6.1, Table 18 (FTP_TRP.1/Admin) in the ST states that all remote administrative communications take place over a secure encrypted TLS (HTTPS) session, allowing remote administrators to initiate TLS (HTTPS) communications with the TOE. TLSv1.2 is supported, using the ciphersuites and RFCs listed under FCS_TLSS_EXT.1, and the RFC listed in FCS_HTTPS_EXT.1.

Component Guidance Assurance Activities: The evaluator shall confirm that the guidance documentation contains instructions for establishing the remote administrative sessions for each supported method.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms are used for any TLS connection, including HTTPS management on the SMC. Section *Certificates* of the **Admin Guide** defines the best practices and procedures for updating the certificates of the TOE appliances to be used for HTTPS management on the SMC. Section *IP Addressing* of the **Admin Guide** describes how to configure the IP address for management of the TOE.

Component Testing Assurance Activities: The evaluator shall perform the following tests:

a) Test 1: The evaluators shall ensure that communications using each specified (in the guidance documentation) remote administration method is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.

b) Test 2: The evaluator shall ensure, for each communication channel, the channel data is not sent in plaintext.

Further assurance activities are associated with the specific protocols.

For distributed TOEs the evaluator shall perform tests on all TOE components according to the mapping of trusted paths to TOE components in the Security Target.

Test 1 – The evaluator used the TLS/HTTPS Web UI interface on the SMC throughout the course of testing. The other Distributed TOE devices do not have a remote interface as they are managed remotely by the SMC.

Test 2 - FCS_TLSS_EXT.1 testing demonstrates that the TLS/HTTPS management connection is not in plaintext.

2.8.3 Trusted Path (NDcPP22e:FTP_TRP.1/Join)

2.8.3.1 NDcPP22e:FTP_TRP.1.1/Join

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.3.2 NDcPP22e:FTP_TRP.1.2/Join

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

2.8.3.3 NDcPP22e:FTP_TRP.1.3/Join

TSS Assurance Activities: None Defined

Guidance Assurance Activities: None Defined

Testing Assurance Activities: None Defined

Component TSS Assurance Activities: The evaluator shall examine the TSS to determine that the methods of joining components to the TOE are identified, along with how those communications are protected, including identification of whether the environment is required to provide confidentiality of the communications or whether the registration data exchanged does not require confidentiality. If the TSS asserts that registration data does not require confidentiality protection then the evaluator shall examine the justification provided to confirm that.

The evaluator shall also check that all protocols listed in the TSS in support of this process are included in the SFRs in the ST, and that if the ST uses FTP_TRP.1/Join for the registration channel then this channel cannot be reused as the normal inter-component communication channel (the latter channel must meet FTP_ITC.1 or FPT_ITT.1).

The evaluator shall examine the TSS to confirm that sufficient information is provided to determine the TOE actions in the case that the initial component joining attempt fails.

Section 6.1, Table 18 (FTP_TRP.1/Join) in the ST states that During installation of the TOE the SMC is configured first, then other appliances are joined to the SMC and after successful registration those appliances are managed via the SMC. When TOE components are joined to the SMC, the administrator installing the non-SMC appliance (FC, FS, or UDPD) inputs the IP address or DNS-resolvable hostname of the SMC, then the new appliance initiates a TLS session to the SMC. This initial TLS session used for joining is authenticated using a valid SMC administrator username and password that is manually entered by the administrator performing the initial configuration of the new appliance. If authentication succeeds, the SMC and the joining appliance exchange X.509 certificates, and the initial TLS session is closed. All subsequent connections between the TOE components also use TLS, but use X.509 certificate-based authentication in accordance with FPT_ITT.1 and FIA_X509_EXT.1/ITT.

No supplemental support is required from the operational environment to secure the TLS connections between the SMC and other TOE components that are being joined to the SMC.

If attempts to join SNA appliances to an SMC fail:

- No FPT_ITT.1 TLS session (using certificate-based authentication per FIA_X509_EXT.1/ITT) can be established between the devices, and thus no TSF data can be exchanged between the devices until registration successfully completes.

- If the connection failed due to network connectivity error, resolve the connectivity, and reattempt joining.
- If the connection fails due to incorrect information provided by the administrator of the joining component (e.g. incorrect hostname/IP address of the SMC, or incorrect account name or password for the SMC), correct the information, and reattempt joining.

Component Guidance Assurance Activities: The evaluator shall examine the guidance documentation to confirm that it contains instructions for establishing and using the enablement and registration channel. The evaluator shall confirm that the guidance documentation makes clear which component initiates the communication. The evaluator shall confirm that the guidance documentation contains recovery instructions should a connection be unintentionally broken during the registration process.

In the case of a distributed TOE that relies on the operational environment to provide security for some aspects of the registration channel security then there are particular requirements on the Preparative Procedures as listed below. (Reliance on the operational environment in this way is indicated in an ST by a reference to operational guidance in the assignment in FTP_TRP.1.3/Join.) In this case the evaluator shall examine the Preparative Procedures to confirm that they:

a) clearly state the strength of the authentication and encryption provided by the registration channel itself and the specific requirements on the environment used for joining components to the TOE (e.g. where the environment is relied upon to prevent interception of sensitive messages, IP spoofing attempts, man-in-the-middle attacks, or race conditions)

b) identify what confidential values are transmitted over the enablement channel (e.g. any keys, their lengths, and their purposes), use of any non-confidential keys (e.g. where a developer uses the same key for more than one device or across all devices of a type or family), and use of any unauthenticated identification data (e.g. IP addresses, self-signed certificates)

c) highlight any situation in which a secret value/key may be transmitted over a channel that uses a key of lower comparable strength than the transmitted value/key. Comparable strength is defined as the amount of work required to compromise the algorithm or key and is typically expressed as 'bits' of security. The ST author and evaluator should consult NIST 800-57 Table 2 for further guidance on comparable algorithm strength.

Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode after the certificates are updated, to ensure that only compliant algorithms are used.

Section *Secure Installation* in the **Admin Guide** provides general guidance for the initial configuration of the appliances, the registration process and specific guidance for the evaluated configuration. This section also refers to the SNA Installation and Configuration Guides*, "Configuring the Managed System"* and *"Finishing Appliance Configurations"* sections for detailed instructions including the configuration order of the appliances.

Adding an appliance is not initiated by the SMC. It must be initiated by the administrator performing the initial configuration of the other appliances (UDP Director, Flow Collector and/or Flow Sensor). During the initial configuration process the administrator is asked whether the new appliance will be managed by an SMC, and after answering affirmatively, that administrator is prompted to trust the SMC certificate, provide the IP address or hostname of the SMC and a valid SMC administrator username and password to authenticate to the SMC. Recovery instructions are provided if the connection fails during the registration process due to network connectivity errors or incorrect information provided. If authentication succeeds, the SMC and the joining appliance exchange certificates, and subsequent TLS connections between the TOE components are authenticated using X.509 certificate-based authentication.

The *Configuration for Common Criteria Compliance* section of the **Admin Guide** indicates that the first step in configuring the TOE for compliance is to complete all the certificates requirements and procedures. Section *Certificates* of the **Admin Guide** indicates that the communication between TOE appliances is authenticated using x509v3 certificates. This section defines the best practices and procedures for replacing the certificates of the TOE appliances to update the initial self-signed certificates that were used as part of registration, thus increasing the security of the distributed TOE channel.

Section *Secure Installation* in the **Admin Guide** describes how to disable communication between distributed components. This can be initiated from the SMC or from a managed appliance and will result in each TOE component deleting the device association from its local configuration. After an appliance is removed, no further TSF data will be exchanged between the appliances without completing a new registration process.

Section *Connection Recovery* in the **Admin Guide** provides recovery instructions if a connection fails between the SMC and the managed appliances during operation. The connections will automatically retry and restore connectivity. Connections will always be temporarily unavailable when appliances are rebooting.

Component Testing Assurance Activities: The evaluator shall perform the following tests:

a) Test 1: The evaluator shall ensure that the communications path for joining components to the TSF is tested for each distinct (non-equivalent) component type [The intention here is to cover all different software sections involved. For example, a single software image may be installed on different TOE components, but with different sections of the image executed according to the hardware platform or communications stack. In such as case tests should be carried out for each different software section.], setting up the connections as described in the guidance documentation and ensuring that communication is successful. In particular the evaluator shall confirm that requirements on environment protection for the registration process are consistent with observations made on the test configuration (for example, a requirement to isolate the components from the Internet during registration might be inconsistent with the need for a component to contact a license server). If no requirements on the registration environment are identified as necessary to protect confidentiality, then the evaluator shall confirm that the key used for registration can be configured (following the instructions in the guidance documentation) to be at least the same length as the key used for the internal TSF channel that is being enabled. The evaluator shall confirm that the key used for the channel is unique to the pair of components (this is done by identifying the relevant key during the registration test: it is not necessary to examine the key value).

b) Test 2: The evaluator shall follow the guidance documentation to ensure that in fact the communication channel can be enabled by a Security Administrator for all the TOE components identified in the guidance documentation as capable of initiation.

c) Test 3: The evaluator shall ensure that if the guidance documentation states that the channel data is encrypted then the data observed on the channel is not plaintext.

Further assurance activities are associated with the specific protocols.

Test 1 – The evaluator followed the guidance documentation to register the SMC with each of the other TOE devices as part of the initial setup of each device. In order for the registration to be successful the SMC required its admin credentials as well as the administrator confirming that the SMC trust certificate matched what was expected, thus confirming the identity of the SMC. After each of the TOE components were registered the evaluator viewed the devices via central management on the SMC.

Test 2 – This was performed as part of Test 1.

Test 3 – The evaluator captured the traffic during the joining of the TOE components to the SMC. The capture demonstrated that the traffic of the joining channel was encrypted via TLS.

3. Protection Profile SAR Assurance Activities

3.1 Development (ADV)

3.1.1 Basic Functional Specification (ADV_FSP.1)

Assurance Activities: The EAs for this assurance component focus on understanding the interfaces (e.g., application programing interfaces, command line interfaces, graphical user interfaces, network interfaces) described in the AGD documentation, and possibly identified in the TOE Summary Specification (TSS) in response to the SFRs. Specific evaluator actions to be performed against this documentation are identified (where relevant) for each SFR in Section 2, and in EAs for AGD, ATE and AVA SARs in other parts of Section 5.

The EAs presented in this section address the CEM work units ADV_FSP.1-1, ADV_FSP.1-2, ADV_FSP.1-3, and ADV_FSP.1-5.

The EAs are reworded for clarity and interpret the CEM work units such that they will result in more objective and repeatable actions by the evaluator. The EAs in this SD are intended to ensure the evaluators are consistently performing equivalent actions.

The documents to be examined for this assurance component in an evaluation are therefore the Security Target, AGD documentation, and any required supplementary information required by the cPP: no additional 'functional specification' documentation is necessary to satisfy the EAs. The interfaces that need to be evaluated are also identified by reference to the EAs listed for each SFR, and are expected to be identified in the context of the Security Target, AGD documentation, and any required supplementary information defined in the cPP rather than as a separate list specifically for the purposes of CC evaluation. The direct identification of documentation requirements and their assessment as part of the EAs for each SFR also means that the tracing required in ADV_FSP.1.2D (work units ADV_FSP.1-4, ADV_FSP.1-6 and ADV_FSP.1-7) is treated as implicit and no separate mapping information is required for this element.

The evaluator shall examine the interface documentation to ensure it describes the purpose and method of use for each TSFI that is identified as being security relevant.

In this context, TSFI are deemed security relevant if they are used by the administrator to configure the TOE, or to perform other administrative functions (e.g. audit review or performing updates). Additionally, those interfaces that are identified in the ST, or guidance documentation, as adhering to the security policies (as presented in the SFRs), are also considered security relevant. The intent is that these interfaces will be adequately tested, and having an understanding of how these interfaces are used in the TOE is necessary to ensure proper test coverage is applied.

The set of TSFI that are provided as evaluation evidence are contained in the Administrative Guidance and User Guidance.

The evaluator shall check the interface documentation to ensure it identifies and describes the parameters for each TSFI that is identified as being security relevant.

The evaluator shall examine the interface documentation to develop a mapping of the interfaces to SFRs.

The evaluator uses the provided documentation and first identifies, and then examines a representative set of interfaces to perform the EAs presented in Section 2, including the EAs associated with testing of the interfaces.

It should be noted that there may be some SFRs that do not have an interface that is explicitly 'mapped' to invoke the desired functionality. For example, generating a random bit string, destroying a cryptographic key that is no longer needed, or the TSF failing to a secure state, are capabilities that may be specified in SFRs, but are not invoked by an interface.

However, if the evaluator is unable to perform some other required EA because there is insufficient design and interface information, then the evaluator is entitled to conclude that an adequate functional specification has not been provided, and hence that the verdict for the ADV_FSP.1 assurance component is a 'fail'.

For this PP, the Evaluation Activities for this family focus on understanding the interfaces presented in the TSS in response to the functional requirements and the interfaces presented in the AGD documentation. No additional 'functional specification' documentation is necessary to satisfy the Evaluation Activities specified in the SD.

3.2 Guidance documents (AGD)

3.2.1 Operational User Guidance (AGD_OPE.1)

Assurance Activities: The documentation must describe the process for verifying updates to the TOE for each method selected for FPT_TUD_EXT.1.3 in the Security Target. The evaluator shall verify that this process includes the following steps (per TD0536):

The evaluator performs the CEM work units associated with the AGD_OPE.1 SAR. Specific requirements and EAs on the guidance documentation are identified (where relevant) in the individual EAs for each SFR.

In addition, the evaluator performs the EAs specified below.

The evaluator shall ensure the Operational guidance documentation is distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

The evaluator shall ensure that the Operational guidance is provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

The evaluator shall ensure that the Operational guidance contains instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

The evaluator shall ensure the Operational guidance makes it clear to an administrator which security functionality and interfaces have been assessed and tested by the EAs.

In addition the evaluator shall ensure that the following requirements are also met.

a) The guidance documentation shall contain instructions for configuring any cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.

b) The documentation must describe the process for verifying updates to the TOE by verifying a digital signature. The evaluator shall verify that this process includes the following steps:

1) Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

2) Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes instructions that describe at least one method of validating the hash/digital signature.

c) The TOE will likely contain security functionality that does not fall in the scope of evaluation under this cPP. The guidance documentation shall make it clear to an administrator which security functionality is covered by the Evaluation Activities.

The **Admin Guide** identifies the TOE as Cisco Secure Network Analytics (SNA) products v7.4. The *1.4 Hardware, Software, and Functionality*section in the **Admin Guide** identifies the TOE components, as well as required components in the operating environment of the TOE. Section *Cryptography for FIPS and CC* in the **Admin Guide** identifies the cryptographic module within the TOE which is the CiscoSSL FOM 7.2a. Section Sample Network Topology provides a diagram outlining the deployment configuration of the TOE.

Throughout the **Admin Guide**, there are notes clarifying the boundaries of the TOE and anything that is excluded in the evaluated configuration making it clear to an administrator which security functionality and interfaces have been assessed and tested in the evaluated configuration. Section *Excluded Functionality* in the **Admin Guide** identifies all of the features that are not supported in the evaluated configuration.

As identified throughout this AAR, the **Admin Guide** provides instructions for configuring the TOE's cryptographic security functions. Section *Compliance Mode (FIPS and Common Criteria)* of the **Admin Guide** describes the processes to enable FIPS and CC mode to ensure that only compliant algorithms and requirement are satisfied.

Section *Product Updates*of the **Admin Guide** describes how the admin can view the installed version of the software of each of the TOE components through Central Management > Update Manager. The Update Manager also shows the last reboot, version ready to install, and the update status. The section further describes the steps for performing manual updates for all the TOE components. The section first states that before updating the TOE should be removed from FIPS and CC modes, these will need to be reenabled once the update process is completed. Software updates for each TOE component can be downloaded from the Cisco software download site at https://software.cisco.com.and a SHA-512 checksum displayed on the download page. A command can be run on your local workstation to calculate the checksum. If the result does not match the checksum, the admin is directed to not upload the update file. All updates for all TOE components are uploaded and deployed via Central Management > Update Manager on the SMC. The section provides guidance on determining the order in which the appliances should be updated.

3.2.2 Preparative Procedures (AGD_PRE.1)

Assurance Activities: As with the operational guidance, the developer should look to the Evaluation Activities to determine the required content with respect to preparative procedures.

It is noted that specific requirements for Preparative Procedures are defined in [SD] for distributed TOEs as part of the Evaluation Activities for FCO_CPC_EXT.1 and FTP_TRP.1(2)/Join.

The evaluator performs the CEM work units associated with the AGD_PRE.1 SAR. Specific requirements and EAs on the preparative documentation are identified (and where relevant are captured in the Guidance Documentation portions of the EAs) in the individual EAs for each SFR.

Preparative procedures are distributed to administrators and users (as appropriate) as part of the TOE, so that there is a reasonable guarantee that administrators and users are aware of the existence and role of the documentation in establishing and maintaining the evaluated configuration.

In addition, the evaluator performs the EAs specified below.

The evaluator shall examine the Preparative procedures to ensure they include a description of how the administrator verifies that the operational environment can fulfil its role to support the security functionality (including the requirements of the Security Objectives for the Operational Environment specified in the Security Target).

The documentation should be in an informal style and should be written with sufficient detail and explanation that they can be understood and used by the target audience (which will typically include IT staff who have general IT experience but not necessarily experience with the TOE product itself).

The evaluator shall examine the Preparative procedures to ensure they are provided for every Operational Environment that the product supports as claimed in the Security Target and shall adequately address all platforms claimed for the TOE in the Security Target.

The evaluator shall examine the preparative procedures to ensure they include instructions to successfully install the TSF in each Operational Environment.

The evaluator shall examine the preparative procedures to ensure they include instructions to manage the security of the TSF as a product and as a component of the larger operational environment.

In addition the evaluator shall ensure that the following requirements are also met.

The preparative procedures must

a) include instructions to provide a protected administrative capability; and

b) identify TOE passwords that have default values associated with them and instructions shall be provided for how these can be changed.

The evaluation team had the following documents to use when configuring the TOE:

- Cisco Secure Network Analytics (SNA) 7.4 Preparative Procedures & Operational User Guide for the Common Criteria Certified Configuration, Version 0.3 (**Admin Guide**)

In some instances, the document referenced general Cisco manuals which the evaluation could find on the Cisco web site.   The completeness of the documentation is addressed by their use in the AA's carried out in the evaluation.

3.3 Life-cycle support (ALC)

3.3.1 Labelling of the TOE (ALC_CMC.1)

Assurance Activities: This component is targeted at identifying the TOE such that it can be distinguished from other products or versions from the same vendor and can be easily specified when being procured by an end user. A label could consist of a 'hard label' (e.g., stamped into the metal, paper label) or a 'soft label' (e.g., electronically presented when queried).

The evaluator performs the CEM work units associated with ALC_CMC.1.

When evaluating that the TOE has been provided and is labelled with a unique reference, the evaluator performs the work units as presented in the CEM.

The evaluator verified that the ST, TOE and Guidance are all labeled with the same hardware versions and software. The information is specific enough to procure the TOE and it includes hardware models and software versions. The evaluator checked the TOE software version and hardware identifiers during testing by examining the actual machines used for testing.

3.3.2 TOE CM Coverage (ALC_CMS.1)

Assurance Activities: Given the scope of the TOE and its associated evaluation evidence requirements, the evaluator performs the CEM work units associated with ALC_CMS.1.

When evaluating the developer's coverage of the TOE in their CM system, the evaluator performs the work units as presented in the CEM.

See section 3.3.1 above for an explanation of how all CM items are addressed.

3.4 Tests (ATE)

3.4.1 Independent Testing - Conformance (ATE_IND.1)

Assurance Activities: Testing is performed to confirm the functionality described in the TSS as well as the guidance documentation (includes 'evaluated configuration' instructions). The focus of the testing is to confirm that the requirements specified in Section 5.1.7 are being met. The Evaluation Activities in [SD] identify the specific testing activities necessary to verify compliance with the SFRs. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this cPP.

The focus of the testing is to confirm that the requirements specified in the SFRs are being met. Additionally, testing is performed to confirm the functionality described in the TSS, as well as the dependencies on the Operational guidance documentation is accurate.

The evaluator performs the CEM work units associated with the ATE_IND.1 SAR. Specific testing requirements and EAs are captured for each SFR in Sections 2, 3 and 4.

The evaluator should consult Appendix B when determining the appropriate strategy for testing multiple variations or models of the TOE that may be under evaluation.

Note that additional Evaluation Activities relating to evaluator testing in the case of a distributed TOE are defined in section B.4.3.1.

The evaluator created a Detailed Test Report (DTR) to address all aspects of this requirement. The DTR discusses the test configuration, test cases, expected results, and test results. The test configuration consisted of the following TOE platform along with supporting products.

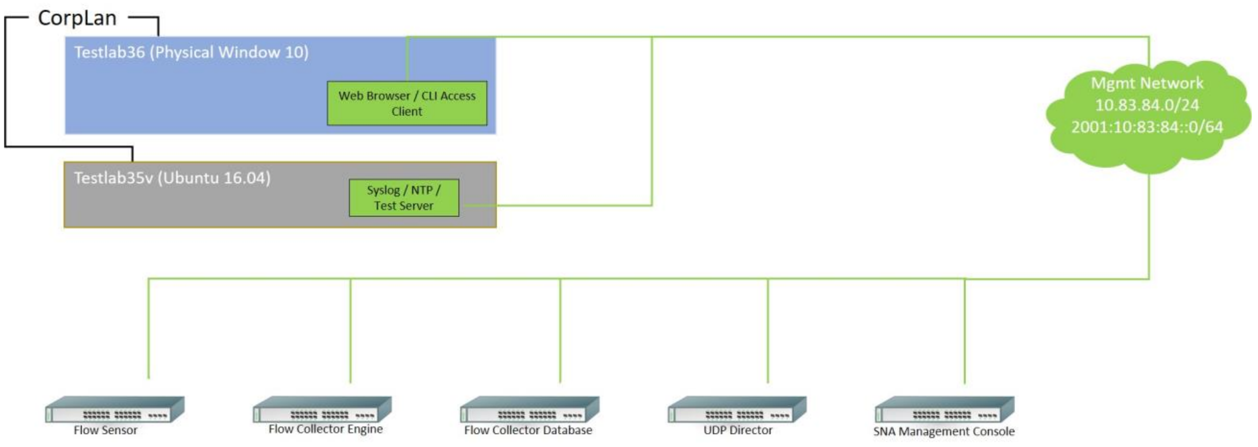
CorpLan
Testlab36 (Physical Window 10)
Web Browser / CLI Access Client
Testlab35v (Ubuntu 16.04)
Syslog / NTP / Test Server
Mgmt Network 10.83.84.0/24 2001:10:83:84::0/64
Flow Sensor
Flow Collector Engine
Flow Collector Database
UDP Director
SNA Management Console

Figure 1 Test Setup

TOE Platforms:

- SNA Management Console (SMC)
  - Physical
    - ST-SMC2210-K9 (UCSC-C220-M5 server)
  - Virtual
    - L-ST-SMC-VE-K9
- SNA Flow Collector
  - Physical
    - ST-FC5210-E (Engine) (UCSC-C220-M5 server)
    - ST-FC5210-D (Database) (UCSC-C240-M5 server)
  - Virtual
    - L-ST-FC-VE-K9
- SNA Flow Sensor

    - Physical
        - ST-FS4210-K9 (UCSC-C220-M5 server)
    - Virtual
        - L-ST-FS-VE-K9
- SNA UDP Director
    - Physical
        - ST-UDP2210-K9 (UCSC-C220-M5 server)
    - Virtual
        - L-ST-UDP-VE-K9

The virtual devices were installed on a UCS C220-M5 device running ESXi 7.0.

Supporting Platforms and Software:

- Windows 10, 64-bit
    - Standard Windows utilities (e.g., notepad, snip tool)
    - Wireshark version 3.6.5 (used to examine network packets)
    - Microsoft Hyper-V (part of Windows)
- Ubuntu version 16.04.7, 64-bit
    - Standard Linux commands (e.g., cat, grep, awk)
    - OpenSSL version 1.0.2g-fips (used to generate certificates)
    - Stunnel version 5.30 (used for tls client testing)
    - tcpdump (comes with Ubuntu – used to generate packet capture files for network traffic)
    - rsyslog version 8.16.0 (used to accept syslog logs)
    - ntpd version ntpd 4.2.8p4
    - Openldap version 2.4.42
    - Evaluator developed test scripts

3.5 Vulnerability assessment (AVA)

3.5.1 Vulnerability Survey (AVA_VAN.1)

Assurance Activities: While vulnerability analysis is inherently a subjective activity, a minimum level of analysis can be defined and some measure of objectivity and repeatability (or at least comparability) can be imposed on the vulnerability analysis process. In order to achieve such objectivity and repeatability it is important that the evaluator follows a set of well-defined activities, and documents their findings so others can follow their arguments and come to the same conclusions as the evaluator. While this does not guarantee that different evaluation facilities will identify exactly the same type of vulnerabilities or come to exactly the same conclusions, the approach defines the minimum level of analysis and the scope of that analysis, and provides Certification Bodies a measure of assurance that the minimum level of analysis is being performed by the evaluation facilities.

In order to meet these goals some refinement of the AVA_VAN.1 CEM work units is needed. The following table indicates, for each work unit in AVA_VAN.1, whether the CEM work unit is to be performed as written, or if it has been clarified by an Evaluation Activity. If clarification has been provided, a reference to this clarification is provided in the table.

Because of the level of detail required for the evaluation activities, the bulk of the instructions are contained in Appendix A, while an 'outline' of the assurance activity is provided below.

In addition to the activities specified by the CEM in accordance with Table 2, the evaluator shall perform the following activities.

The evaluator shall examine the documentation outlined below provided by the developer to confirm that it contains all required information. This documentation is in addition to the documentation already required to be supplied in response to the EAs listed previously.

The developer shall provide documentation identifying the list of software and hardware components7 that compose the TOE. Hardware components should identify at a minimum the processors used by the TOE. Software components include applications, the operating system and other major components that are independently identifiable and reusable (outside the TOE) such as a web server and protocol or cryptographic libraries. This additional documentation is merely a list of the name and version number of the components, and will be used by the evaluators in formulating hypotheses during their analysis.

If the TOE is a distributed TOE then the developer shall provide:

a) documentation describing the allocation of requirements between distributed TOE components as in [NDcPP, 3.4]

b) a mapping of the auditable events recorded by each distributed TOE component as in [NDcPP, 6.3.3]

c) additional information in the Preparative Procedures as identified in the refinement of AGD_PRE.1 in additional information in the Preparative Procedures as identified in 3.5.1.2 and 3.6.1.2.

The evaluator formulates hypotheses in accordance with process defined in Appendix A. The evaluator documents the flaw hypotheses generated for the TOE in the report in accordance with the guidelines in Appendix A.3. The evaluator shall perform vulnerability analysis in accordance with Appendix A.2. The results of the analysis shall be documented in the report according to Appendix A.3.

The vulnerability analysis is in the Detailed Test Report (DTR) prepared by the evaluator. The vulnerability analysis includes a public search for vulnerabilities and fuzz testing. None of the public search for vulnerabilities, or the fuzz testing uncovered any residual vulnerability.

The evaluation team performed a public search for vulnerabilities in order to ensure there are no publicly known and exploitable vulnerabilities in the TOE from the following sources:

- National Vulnerability Database (https://web.nvd.nist.gov/vuln/search)
- Vulnerability Notes Database (http://www.kb.cert.org/vuls/)
- Rapid7 Vulnerability Database (https://www.rapid7.com/db/vulnerabilities)
- Tipping Point Zero Day Initiative (http://www.zerodayinitiative.com/advisories)
- Exploit / Vulnerability Search Engine (http://www.exploitsearch.net)
- SecurITeam Exploit Search (http://www.securiteam.com)
- Tenable Network Security (http://nessus.org/plugins/index.php?view=search)
- Offensive Security Exploit Database (https://www.exploit-db.com/)

The search was performed on 2/20/2023 with the following search terms: "Cisco", "Cisco Secure Network Analytics", "SNA 7.4", "Cisco SSL FOM", "SNA Management Console", "SNA Flow Collector", "SNA Flow Sensor", "SNA UDP Director", "ESXi", "Intel Xeon Scalable Processor", "Intel Xeon E5", "Intel Xeon Gold", "Intel Xeon Bronze", "Stealthwatch", "Cisco SMC".

3.5.2 Additional Flaw Hypotheses (AVA_VLA.1)

Assurance Activities: The following additional tests shall be performed:1.) [Conditional]: If the TOE is a TLS server and supports ciphersuites that use RSA transport (e.g. supporting TLS_RSA_WITH_* ciphers) the following test shall be performed. Where RSA Key Establishment schemes are claimed and especially when PKCS#1 v1.5* padding is used, the evaluators shall test for implementation flaws allowing Bleichenbacher and Klima et al. style attacks, including Bock et al's ROBOT attacks of 2017 in the flaw analysis. Even though Bleichenbacher's original paper is two decades old, Bock et al. found these attacks to still be effective in weakening the security of RSA key establishment in current products. Bleichenbacher and Klima et al. style attacks are complex and may be difficult to detect, but a number of software testing tools have been created to assist in that process. The iTC strongly recommends that at least one of the tools mentioned in Bock et al's ROBOT attacks of 2017 webpage or paper, as effective to detect padding oracle attacks, be used to test TOE communications channels using RSA based Key Establishment (related sources: http://archiv.infsec.ethz.ch/education/fs08/secsem/bleichenbacher98.pdf, https://eprint.iacr.org/2003/052, https://robotattack.org/). Network Device Equivalency Considerations

Since the TOE acts as a TLS server and supports ciphersuites that use RSA transport, the evaluator utilized the robot-check detection tool to check for implementation flaws allowing Bleichenbacher and Klima et al. style attacks. Both the SMC and Virtual SMC were found to be not vulnerable.