Information Security Corporation

CertAgent/Dhuma v8.0 Patch Level 0.2

Assurance Activities Report

Version 1.1

August 23, 2024

Prepared by:

Leidos Inc.
https://www.leidos.com/CC-FIPS140
Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, MD 21046

Prepared for:

National Information Assurance Partnership
Common Criteria Evaluation and Validation Scheme

The Developer of the TOE:

Information Security Corporation
1011 W. Lake St., Ste. 425
Oak Park, IL 60301

The TOE Evaluation was Sponsored by:

Information Security Corporation
1011 W. Lake St., Ste. 248
Oak Park, IL 60301

Evaluation Personnel:

Greg Beaver
Armin Najafabadi
Pascal Patin
Srilekha Vangala

Contents

1 Introduction

This document presents results from performing evaluation activities associated with the CertAgent Version 8.0 patch level 0.2 for Windows and CertAgent Version 8.0 patch level 0.2 for Linux evaluation. This report contains sections documenting the performance of evaluation activities associated with each of the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) as specified in Evaluation Activities for the Protection Profile for Certification Authorities, Version 2.1, December 2017 and including the following optional, selection-based, and objective SFRs: FAU_SAR.1, FAU_SAR.3, FAU_SCR_EXT.1, FAU_SEL.1, FAU_STG_EXT.1, FCS_CKM.1, FCS_CKM.2, FCS_CKM_EXT.1(1), FCS_CKM_EXT.1(2), FCS_CKM_EXT.1(3), FCS_CKM_EXT.1(4), FCS_CKM_EXT.4, FCS_CKM_EXT.5, FCS_CKM_EXT.6, FCS_CKM_EXT.7, FCS_CKM_EXT.8, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5), FCS_HTTPS_EXT.1, FCS_RBG_EXT.1, FCS_TLSS_EXT.1, FDP_CRL_EXT.1, FDP_OCSPG_EXT.1, FDP_STG_EXT.1, FIA_ENR_EXT.1, FIA_ESTS_EXT.1, FIA_X509_EXT.3, FPT_TST_EXT.2, FTA_SSL.3, and FTP_ITC.1.

1.1 Applicable Technical Decisions

The NIAP Technical Decisions (TDs) referenced below apply to [PP CA]. Rationale is included for those TDs that do not apply to this evaluation.

TD0844- Addition of Assurance Package for Flaw Remediation V1.0 Conformance Claim

The TD is applicable to the TOE but does not affect any evidence or evaluation materials

TD0276 – X.509 Code Signing on TOE Updates

The TD is applicable to the TOE and the ST. The ST includes FIA_X509_EXT.2.1.

TD0278 – Clarification of Role for Managing Manual Certificate Requests

The TD is applicable to the TOE and the ST. The ST includes FMT_MOF.1(1) and FMT_MOF.1(3).

TD0286 – Audit Events for FPT_RCV.1

The TD is applicable to the TOE and the ST. The ST includes FPT_RCV.1.

TD0287 – FAU_STG.4 Testing

The TD is applicable to the TOE and the ST. The ST includes FAU_STG.4.

TD0294 – Correction of TLS SFRs in CA PP ver 2.1

The TD is applicable to the TOE and the ST. The ST includes FCS_TLSS_EXT.1 and FCS_TLSS_EXT.2.

TD0328 – Split Knowledge Procedures Distinction

The TD is not applicable to the TOE or the ST. FPT_SKY_EXT.1 and FPT_SKY_EXT.2 are optional SFRs that are not included in the ST. FPT_SKY_EXT.1/CA is performed entirely by the OE, this SFR is not included in the ST and OE.KEY_ARCHIVAL is included in the ST.

TD0348 – FCS_TLSS_EXT.2.4 for TLS 1.2 or Higher

The TD is applicable to the TOE and the ST. The ST includes FCS_TLSS_EXT.2.4.

TD0353 – Guidance for Certificate Profiles

The TD is applicable to the TOE and the ST. The ST includes FDP_CER_EXT.1.1.

TD0375 – FMT_MOF.1(4) Selection

The TD is applicable to the TOE and the ST. The ST includes FMT_MOF.1(4).

TD0415 – Trusted Update Test 4 Conditional

The TD is applicable to the TOE and the ST. The TOE supports use of X509 certificates for code signing.

TD0500 – Cryptographic Selections and Updates for CAPP

The TD is applicable to the TOE and the ST. The ST contains FCS_CKM.1 and FCS_CKM.2.

TD0522 – Updates to Certificate Revocation (FIA_X509_EXT.1)

The TD is applicable to the TOE and the ST. The ST contains FIA_X509_EXT.1.

TD0599 - Corrections to SAR Section in CAPP

This TD is applicable to the TOE. No changes are required for the ST or AGD.

TD0782 - Terminology Change in CAPP: Extended to Functional Package

The TD is not applicable to the TOE. SSH is not selected in FIA_X509_EXT.2.2 or FTP_ITC.1.3. The selection based FDP_ITT.1.1 and FPT_ITT.1.1 are not included in the ST.

TD0796 - Removal of SHA-1 from Various Selections

The TD is applicable to the TOE. The ST contains several of the affected SFRs.

1.2 Evidence

[PP CA] Protection Profile for Certification Authorities, Version 2.1, 01 December 2017

[ST] Information Security Corporation CertAgent/Dhuma v8.0 patch level 0.2 Security Target for Common Criteria Evaluation, Software Version: 8.0 patch level 0.2, Version: 5.0.12, Issue Date: August 23, 2024

[CCECG] Information Security Corporation CertAgent/Dhuma v8.0 patch level 0.2 Guidance for Common Criteria Evaluation, Version: 3.0.5, Issue Date: July 25, 2024.

[Test] Information Security Corporation CertAgent 8.0 Patch Level 0.2 Common Criteria Test Report and Procedures For Protection Profile for Certification Authorities Version 2.1, Version: 1.0, Dated: June 20, 2024.

[AVA] Information Security Corporation CertAgent/Dhuma 8.0 patch level 0.2 Vulnerability Analysis, Version: 1.1, Date: August 23, 2024

1.3 Conformance Claims

Common Criteria Versions

- Common Criteria for Information Technology Security Evaluation Part 1: Introduction, Version 3.1, Revision 5, dated: April 2017.

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Revision 5, dated: April 2017.
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Revision 5, dated: April 2017.

Common Evaluation Methodology Versions

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

Protection Profiles

- [PP CA] Protection Profile for Certification Authorities, Version 2.1, 01 December 2017

1.4 SAR Evaluation

The following Security Assurance Requirements (SARs) were evaluated during the evaluation of the TOE:

SAR	Verdict
ASE_CCL.1	Pass
ASE_ECD.1	Pass
ASE_INT.1	Pass
ASE_OBJ.1	Pass
ASE_REQ.1	Pass
ASE_TSS.1	Pass
ADV_FSP.1	Pass
AGD_OPE.1	Pass
AGD_PRE.1	Pass
ALC_CMC.1	Pass
ALC_CMS.1	Pass
ATE_IND.1	Pass
AVA_VAN.1	Pass

The evaluation work units are listed in the proprietary ETR. The evaluators note per the PP evaluation activities that many of the SARs were successfully evaluated through completion of the associated evaluation activities present in the claimed PP.

1.5 Certificate Table

The following tables summarize the relevant cryptographic validation certificates for the two components providing cryptographic functionality in the solution. The sections that follow provide additional information for each cryptographic requirement.

Thales TCT T-5000 Luna Network HSM- CMVP #3898				
Operation(s)	**SFR**	**Algorithm**	**Certificate**	**Standard**
Generate Random	FCS_RBG_EXT.1	DRBG (Hash) SHA-256	DRBG 349	SP800-90A
		SHA-256	SHS 2112	FIPS 180-4
ECDSA Key Gen	FCS_CKM.1(b)	ECDSA P-256, P-384, P-521	C1999	FIPS 186-4
		DRBG (Hash) SHA-256	DRBG 349	SP800-90A
ECDSA Sign	FCS_COP.1(2)(a)	ECDSA P-256, P-384, P-521	C1999	FIPS 186-4
		SHA2-256, SHA2-384, SHA2-512	C1999	FIPS 180-4
		DRBG (Hash) SHA-256	DRBG 349	SP800-90A
RSA Key Gen	FCS_CKM.1(a)	RSA 3072	C2010	FIPS 186-4
		SHA2-256, SHA2-384, SHA2-512	C1999	FIPS 180-4
		DRBG (Hash) SHA-256	DRBG 349	SP800-90A
RSA Sign	FCS_COP.1(2)(a)	RSA 3072	C2010	FIPS 186-4
		SHA2-256, SHA2-384, SHA2-512	C1999	FIPS 180-4
		DRBG (Hash) SHA-256	DRBG 349	SP800-90A

ISC CDK CAVP CERTIFICATES				
Operation(s)	**SFR(s)**	**Algorithm**	**Certificate**	**Standard(s)**
AES Encrypt/Decrypt	FCS_COP.1(1)	AES-CBC-256	A3042	FIPS 197; SP800-38A/D
AES Encrypt/Decrypt	FCS_COP.1(1)	AES-GCM-256	A3042	FIPS 197; SP800-38A/D
Generate Random	FCS_RBG_EXT.1	DRBG (HMAC-SHA-256)	A3042	SP800-90

		HMAC-SHA2-256	A3042	FIPS 198-1
ECDSA Key Gen	FCS_CKM.1(b)	ECDSA, ECDH P-256, P-384, P-512	A3042	FIPS 186-4
		DRBG (HMAC-SHA-256)	A3042	SP800-90
ECDSA Sign	FCS_COP.1(2)(b)	ECDSA P-256, P-384, P-521	A3042	FIPS 186-4
		SHA2-256, SHA2-384, SHA2-512	A3042	FIPS 180-4
		DRBG (HMAC-SHA-256)	A3042	SP800-90
ECDSA Verify	FCS_COP.1(2)(b)	ECDSA P-256, P-384, P-521	A3042	FIPS 186-4
		SHA2-256, SHA2-384, SHA2-512	A3042	FIPS 180-4
Password-based KDF	FCS_COP.1(5)	PBKDF (HMAC-SHA-256)	A3042	SP800-132
Key Establishment	FCS_CKM.2	KAS-ECC P-256, P-384, P-521	A3042	SP800-56A R3
		SHA2-256, SHA2-384, SHA2-512	A3042	FIPS 180-4
		HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512	A3042	FIPS 198-1
		TLS KDF (SHA2-256, SHA2-384, SHA2-512)	A3042	SP800-135
RSA Key Gen	FCS_CKM.1(a)	RSA 3072, 4096	A3042	FIPS 186-4
		DRBG (HMAC-SHA-256)	A3042	SP800-90
RSA Sign	FCS_COP.1(2)(b)	RSA 3072, 4096	A3042	FIPS 186-4
		SHA2-256, SHA2-384, SHA2-512	A3042	FIPS 180-4

		DRBG (HMAC-SHA-256)	A3042	SP800-90
RSA Verify	FCS_COP.1(2)(b)	RSA 3072, 4096	A3042	FIPS 186-4
		SHA2-256, SHA2-384, SHA2-512	A3042	FIPS 180-4
Hash	FCS_COP.1(3)	SHA2-256, SHA2-384, SHA2-512	A3042	FIPS 180-4
Keyed Hash	FCS_COP.1(4)	HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512	A3042	FIPS 198-1, FIPS 180-4

2 Security Functional Requirement Evaluation Activities

This section describes the evaluation activities associated with the SFRs defined in the ST and the results of those activities as performed by the evaluation team. The evaluation activities are derived from [PP CA] and modified by applicable NIAP TDs. Evaluation activities for SFRs not claimed by the TOE have been omitted.

2.1 Security Audit (FAU)

2.1.1 Audit Dependencies (FAU_ADP_EXT.1)

2.1.1.1 TSS Activities

> The evaluator shall examine the TSS and operational guidance in order to verify that they describe each of the relevant auditable events, how audit records of these events are formatted, and what component of the TOE or Operational Environment is responsible for handling these events.
>
> For those auditable events that are generated by the TOE and stored within the TOE boundary, the assurance activities are included for the relevant selection-based audit SFRs.

The TSS in section 9.1.1 describes that the TOE both implements audit functions and interfaces with the operational environment audit functions to generate audit events identified in table 14 of the ST. Table 14 identifies the events that are generated by the TOE and those that are generated by the underlying OS.

Details of how audit events are formatted are covered in [] Section 5.2.2.2.1 ***Audit Tables in Database***. The description of each available field, the format of the field, and data displayed in the Audit Trail page of the Admin Site are described. Samples of the audit records are identified in [CCECG] ***Table 43 Sample Database Audit Trail***.

All audit records are generated by the TOE, with the exception of FPT_STM.1. The audit record for FPT_STM.1 is generated by the operating system.

Requirement	Auditable Events	Additional Audit Record Contents	Retention	Event Type	Audit Location
FAU_GEN.1.1	Start-up of the TSF audit functions	None.			Admin table
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	None.	Normal	Audit	Admin table
FCS_CKM.1	All occurrences of non-ephemeral and [no other] key generation for TOE related functions.	Success: public key generated	Normal	System	Admin table
FCS_CKM.2	All occurrences of non-ephemeral and [no other] key establishment for TOE related functions.	Success: key established	Normal	System	Admin table
FCS_CKM_EXT.4	Failure of the key destruction process for TOE related keys.	Identity of object or entity being cleared.	N/A	N/A	N/A
FCS_CKM_EXT.5	Detection of integrity violation for stored TSF data.	None.	Normal	Login, NIAP, Request	Admin table, CA table
FCS_COP.1(2)	All occurrences of signature generation using a CA signing key.	Name/identifier of object being signed Identifier of key used for signing. None	Extended	System, Credential, Request, CRL, OCSP, RAMI, EST	Admin table, CA table

Requirement	Auditable Events	Additional Audit Record Contents	Retention	Event Type	Audit Location
	Failure in signature generation				
FCS_HTTPS_EXT.1	Failure to establish a HTTPS session. Establishment/Termination of a HTTPS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.	Normal	TLS Session	Admin table
FCS_TLSS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. None	Normal	TLS Session	Admin table
FCS_TLSS_EXT.2	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. None	Normal	TLS Session	Admin table
FDP_CER_EXT.1	Certificate generation.	Success: [certificate object identifier]	Extended	Credential, System, Request	Admin table (“System” credential), CA table (Issuer Key, user certificates)
FDP_CER_EXT.2	Linking of certificate to certificate request.	Success: [certificate object identifier], [link to certificate request object identifier]	Extended	Request	CA table

Requirement	Auditable Events	Additional Audit Record Contents	Retention	Event Type	Audit Location
		Failure: Reason for failure, [link to certificate request object identifier].			
FDP_CER_EXT.3	Failed certificate approvals.	Reason for failure, [link to certificate request object identifier].	Normal	Request	CA table
FDP_STG_EXT.1	Changes to the trusted public keys and certificates relevant to TOE functions, including additions and deletions	The public key and all context information associated with the key.	Normal	NIAP	Admin table
FDP_CRL_EXT.1	Failure to generate a CRL.	None.	Normal	CRL	CA table
FDP_OCSPG_EXT.1	Failure to generate certificate status information.	None.	Extended	OCSP	CA table
FIA_X509_EXT.1	Failed certificate validations.	None.	Normal	CA Account, Login, TLS Session	Admin table
FIA_X509_EXT.2	Failed authentications.	None.	Normal	CA Account, Login	Admin table
FIA_UAU_EXT.1	All uses of the authentication mechanism for access to TOE related functions.	Origin of the attempt (e.g., IP address).	Normal	Login	Admin table, CA table

Requirement	Auditable Events	Additional Audit Record Contents	Retention	Event Type	Audit Location
FIA_UIA_EXT.1	All use of the identification and authentication mechanism used for TOE related roles.	Provided user identity. Origin of the attempt (e.g., IP address).	Normal	Login	Admin table, CA table
FIA_ESTS_EXT.1	EST requests (generated or received) containing certificate request or revocation requests EST responses issued.	Identifiers for all entities authenticating the request, including the entity providing client authentication for the EST transport (if any). The submitted request. Any signed response.	Extended	EST	CA table
FMT_SMR.2	Modifications to the group of users that are part of a role.	Modifications to the group of users that are part of a role.	Extended	ACL, CA Account, Configuration	Admin table (ACL, CA Account), CA table (Configuration)
FPT_FLS.1	Invocation of failures under this requirement.	Indication that the TSF has failed with the type of failure that occurred.	Normal	System	Admin table, CA table, local log file for failure notices

Requirement	Auditable Events	Additional Audit Record Contents	Retention	Event Type	Audit Location
FPT_KST_EXT.2	All unauthorized attempts to use TOE secret and private keys.	ID of user or process that attempted access.	Normal	CA Account, Login, RAMI, EST	Admin table, CA table
FPT_RCV.1 (TD0286 applied)	The fact that a failure or service discontinuity occurred Resumption of the regular operation	The type of failure or service discontinuity.	Extended	System, Login, NIAP	Admin table, local log file for failure notices
FPT_STM.1 (Generated by the operating system)	Changes to the time.	The old and new values for the time.	Normal	System	Operating System
FPT_TUD_EXT.1	Initiation of update.	Version number	Extended	System	Admin table
FPT_TST_EXT.2	Execution of this set of TSF integrity tests. Detected integrity violations.	For integrity violations, the identity of the object that caused the integrity violation.	Normal	System	Admin table
FTA_SSL.4	The termination of an interactive session.	None.	Normal	Login	Admin table, CA table
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.	Normal	Login	Admin table, CA table
FTP_TRP.1	Initiation of the trusted channel.	Identification of the claimed user identity.	Normal	TLS Session	Admin table

Requirement	Auditable Events	Additional Audit Record Contents	Retention	Event Type	Audit Location
	Termination of the trusted channel. Failures of the trusted path functions.				
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.	Normal	TLS Session	Admin table

2.1.1.2 Guidance Activities

None defined.

2.1.1.3 Test Activities

For any auditable events that are handled by the TOE's Operational Environment, the evaluator shall demonstrate that these events are auditable.

All audit records are generated by the TOE, with the exception of FPT_STM.1. The audit record for FPT_STM.1 is generated by the operating system.

Testing that audit records associated with an SFR are generated is performed in conjunction with testing the SFR.

The evaluator provided the audit log records based on the specified auditable data in Table 14 of the [ST] in which the OE records the time change of the platform.

2.1.2 Generation of Certificate Repository (FAU_GCR_EXT.1)

2.1.2.1 TSS Activities

The evaluator shall examine the TSS to determine that it describes the certificate repository. If the certificate repository is provided by the OE, the evaluator shall check the TSS to ensure it describes the interfaces invoked by the TOE to store certificates (and CRLs).

The TSS in section 9.1.4 states that the TOE certificate repository is provided by the database in its operational environment. The TOE stores its certificates and CRLs in tables in the OE database. When a new issuer accounts is created, the TOE uses the JDBC API to create the database tables where it stores the certificates, CRLs, and audit information for the new issuer.

2.1.2.2 Guidance Activities

None defined.

2.1.2.3 Test Activities

The evaluator shall perform the following tests:

Test 1: The evaluator shall generate a certificate to be stored in the repository. The evaluator shall confirm that the certificate is stored in the certificate repository.

Evidence for this test can be seen in FDP_CER_EXT.2. Test 1 of the [Test] Report. That test requires the evaluator to generate a certificate from a CSR. The evidence for the test includes screenshots demonstrating that the certificate was stored in the TOE's repository prior to being downloaded.

Test 2 (conditional): If "CRLs" are selected in the SFR, the evaluator shall generate a CRL and verify that it is stored in the certificate repository.

Evidence for this test can be seen in FDP_CRL_EXT.1 Test 1 of the [Test] Report. That test required the evaluator to generate a CRL, and the test evidence included evidence of the CRL being stored in the certificate repository.

2.1.3 Audit Data Generation (FAU_GEN.1)

2.1.3.1 TSS Activities

The evaluator shall ensure that the TSS describes every audit event type mandated by the PP and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in Tables 4 through 6, depending on the characterization of the SFR associated with the particular event as mandatory, optional, or selection-based.

[ST] Section 9.1.2 identifies in Table 14 describes every audit event type mandated by the PP and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in the PP Tables 4 through 6.

The evaluator shall also ensure that the TSS describes all cases where the generation of ephemeral key pairs is not audited for FCS_CKM.1.

[ST] Section 9.1.2 states that "The TOE does not audit ephemeral key generation that occurs during TLS session establishment or when the "System Key" is configured as an ECC credential and used for key wrapping."

2.1.3.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure that it describes the audit mechanism, lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field.

Section 5.2.1 of [CCECG] describes the TOE audit function, it describes the audit mechanism, lists all of the required audit events, describes the format of audit events, and the fields in the audit records. During testing of the TOE, by configuring the various functions, the evaluator identified the administrative actions that are necessary to enforce the SFRs. The evaluator reviewed the guidance and ensured that all necessary administrative actions are described.

The TOE relies on the Operational Environment to provide an accurate time stamp for its use and expects the OS to generate an audit record for any changes to the time. The CC-Guide identifies the tools used to review the OS audit trail in section 5.2.2.2.3.

The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP. The evaluator shall examine the operational guidance 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 PP. The evaluator shall document the methodology or approach taken while determining which actions in the operational guidance are security relevant with respect to this PP. The evaluator may perform this activity as part of the activities associated with ensuring the operational guidance satisfies the requirements in accordance with AGD_OPE.

[CCECG] The evaluator identified the administrative actions that are necessary to enforce the SFRs. The evaluator reviewed the guidance and ensured that all necessary administrative actions are described.

The evaluator identified the following management actions:

Management Actions	Guidance reference
Starting and stopping the CertAgent and Audit services	4.1
Managing the administrative and CA Account sites	4.4 and 4.5
Using RAMI and DBACcess	4.7 and 4.8
Updating the TOE	4.10
Manage the audit mechanism/manage audit trail settings	4.4.4
Managing Certificate Profiles	4.5.7.3
Managing Certificate Issuance/ Managing Certificate Enrollment	4.5.7.1, 4.5.4.3 and 4.5.7.4
Managing Certificate Revocation Policy/ Managing CRL issuance/ Managing OCSP Responder settings	4.5.7.5, 4.5.7.6 and 4.5.7.8
Managing the Cryptographic functions	4.4.8, 4.5.3
Perform On-Demand Integrity Tests	4.4.2.1.2

The evaluator shall check that audit review tools are described in the operational guidance and conform to the requirements of FAU_SAR.1.

[CCECG] Section ***5.2.2.2.3 Operating System Logs*** identifies the tools used to review the OS audit trail.

When the Operational Environment is selected in FAU_GEN.1.1 or FAU_GEN.1.2, the evaluator shall examine the operational guidance to ensure the configuration of the Operational Environment necessary to generate the required elements, and instructions on how to examine the various audit records is provided.

[CCECG] Section ***5.2.2.1 FAU_GEN.1.1*** states that The TOE relies on the environmental Operating System's audit facility to generate audit entries for services that the Operating System provides. The Operating System supplies time services to the TOE. The Operating System's own audit capabilities audit changes to the system clock (FPT_STM.1). On the Windows platform, no specific configurations are needed for the OS generated audit event. Audit records can be viewed via the Windows Event Viewer.

[CCECG] Section ***5.1.8 Configuring Audit Setting in RHEL*** provides the instructions to set up auditing for time changes.

2.1.3.3 Test 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 Table 4, any events in Table 5 and Table 6 that correspond with the optional and selection-based SFRs claimed in the Security Target, startup of the audit functions (or startup of the TOE if audit functionality is not enabled or disabled independently of the TOE), and administrative actions. This should include all instances of an event. 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. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable.

The evaluator examined the listed auditable events and verified that the TOE produced appropriate audit records for all of them.

When verifying the test results, the evaluator shall use audit review tools in conformance of FAU_SAR.1 and the operational guidance. The evaluator shall ensure the audit records generated during testing match the format specified in the operational guidance, and that the fields in each audit record have the proper entries and that the audit records are provided in a manner suitable for interpretation. The evaluator shall also ensure the ability to apply searches of audit data based on the type of event, the user responsible for causing the event, and identity of the applicable certificate. When the Operational Environment is selected in FAU_GEN.1.1 or FAU_GEN.1.2, the evaluator shall follow the operational guidance to configure the Operational Environment as specified in the TSS and identify the audit records used and audit information assigned to each audit record. The evaluator shall then inspect the indicated audit records for audit information assigned to each audit record indicated.

Audit records were examined to verify that they contained the required information.

Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the operational guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.

2.1.4 User Identity Association (FAU_GEN.2)

2.1.4.1 TSS, Guidance, and Test Activities

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.

2.1.5 Audit Review (FAU_SAR.1)

2.1.5.1 TSS, Guidance, and Test Activities

This activity should be accomplished in conjunction with the testing of FAU_GEN.1. Review of each of each of the generated audit records demonstrates that these records are reviewable.

[ST] Section 9.1.7 ***FAU_SAR.1 Audit review*** describes how the TOE implements this SFR. The admin and CA Accounts sites have an 'Audit Trail' link visible only to the user with auditor role that is used to review and search the TOE audit trail. Testing of audit review is performed in conjunction with audit generation where the evaluator verified that the relevant audit records are generated and that the records include all required contents.

The evaluator reviewed [CCECG] ***Table 43 Sample Database Audit Trail*** and verified that the audit records contain all of the required content.

2.1.6 Selectable Audit Review (FAU_SAR.3)

2.1.6.1 TSS, Guidance, and Test Activities

This activity should be accomplished in conjunction with the testing of FAU_GEN.1.

[ST] Section 9.1.8 describes how the TOE implements this selectable audit review.

[CCECG] Section 4.5.2 ***Searching the Audit Trail*** Provides the instructions to search the audit trail of a CA account.

2.1.7 Certificate Repository Review (FAU_SCR_EXT.1)

2.1.7.1 TSS Activities

The following activities apply regardless of the selection made in the first selection in the SFR. The test activities can be conducted in conjunction with those for FDP_CER_EXT.1 and FAU_GCR_EXT.1.

[ST] Section 9.1.6 ***FAU_SCR_EXT.1 Certificate Repository Review*** describes how the TOE fulfills this SFR.

The TOE's web interface allows users in the auditor role to search for certificates by subject name or serial number using the certificate search link. The search results include the certificate request ID which can be used to search the audit trail for any events related to that certificate.

The TOE stores all certificate information in tables in the single environmental database. This database also stores most of the TOE's audit records (all but those created if the database is inaccessible) in separate tables. Access to the database is limited to the TOE.

2.1.7.2 Guidance Activities

> The evaluator shall examine the operational guidance to ensure it contains instructions for searching the specified information.

[CCECG] Section ***5.2.4 FAU_SCR_EXT.1 Certificate Repository Review*** describes how an auditor user can access the certificate repository for review on the CA Accounts site. The auditor can search for certificates using subject name and serial number.

2.1.7.3 Test Activities

> The evaluator shall generate a sufficient number and variety of certificates to populate the repository certificates having at least two values for each of the search fields selected in this SFR. The evaluator shall then, following the instructions within the operational guidance, search the repository or audit record for certificates containing specific values for each search field included in the ST, and confirm that all certificates matching the search criteria are returned; all returned certificates match the criteria; and the object identifier is returned. The object identifier will be used in testing for FAU_SAR.3.

The evaluator verified that the TOE's certificate repository could be searched using both certificate serial numbers and subjects.

2.1.8 Selective Audit (FAU_SEL.1)

2.1.8.1 TSS Activities

> There are no TSS assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

[ST] Section 9.1.9 describes that the TOE provides audit configuration functionality accessible to the admin user through the CertAgent/Dhuma Administration Site, where specific auditable events can be selected by event type.

2.1.8.2 Guidance Activities

> The evaluator shall examine the operational guidance to ensure that it itemizes all event types, as well as describes all attributes that are to be selectable in accordance with the requirement, to include those attributes listed in the assignment. The operational guidance shall also contain instructions on how to set the pre-selection as well as explain the syntax (if present) for multi-value pre-selection. The administrative guidance shall also identify those audit records that are always recorded, regardless of the selection criteria currently being enforced.

TABLE 40 ADMIN AUDIT TRAIL FIELD AND DESCRIPTION and ***TABLE 41 CA ACCOUNT AUDIT TRAIL FIELD AND DESCRIPTION*** in the [CCECG] identifies all the audit event types that are selectable for audit. The TOE provides the option to modify the set of events to be audited via the Manage Audit Trail link on the Administrative site. All audit event types are audited by default, but the administrator can modify the selectable audit events. There are no auditable events that are always recorded in the TOE, if the Audit Trail Configuration page is unchecked in the admin site, no audit events will be recorded.

2.1.8.3 Test Activities

The evaluator shall perform the following tests:

Test 1: For each attribute listed in the requirement, the evaluator shall devise a test to show that selecting the attribute causes only audit events with that attribute (or those that are always recorded, as identified in the administrative guidance) to be recorded.

The evaluator configured the TOE to record only TLS-related audit events. The evaluator then performed a number of actions on the TOE and verified that only the ones which opened or terminated a TLS session were audited.

Test 2: [conditional] If the TSF supports specification of more complex audit pre-selection criteria (e.g., multiple attributes, logical expressions using attributes) then the evaluator shall devise tests showing that this capability is correctly implemented. The evaluator shall also, in the test plan, provide a short narrative justifying the set of tests as representative and sufficient to exercise the capability.

The TOE does not support the specification of more complex audit pre-selection criteria.

2.1.9 Protected Audit Event Storage (FAU_STG.4)

2.1.9.1 TSS Activities

Modified by TD0287

The evaluator shall examine the TSS to ensure it describes the behavior of the TSF when the audit trail cannot be written to. The evaluator shall ensure the TSS describes where the audit trail is stored (locally, remotely, or both); how the TSF detects audit full conditions if the audit trail is stored locally; whether and how the TSF detects audit full conditions for remote audit repositories; and how the TSF detects loss of communication with external audit repositories (if using an external audit server). The evaluator shall also ensure the TSS describes what actions can be performed by the Auditor, if any, in each case where the audit trail cannot be written.

[ST] Section ***9.1.1 FAU_ADP_EXT.1 Audit Dependencies*** describes the TOE audit functions. The TOE stores its audit trail outside the TOE boundary, in an environmental database on the local host platform.

[ST] Section ***9.1.5 FAU_STG.4 Prevention of Audit Data Loss*** states that if the locally stored audit trail cannot be written to, the TOE stops all activity and shuts itself down. Once the issue is corrected, at the local console, by an administrator, the TOE can be restarted. The TOE detects issues with the audit trail when it attempts to write audit, or other information, to the environmental database and an exception is thrown. If the database fails with an error indicating that the local file system containing the database and other log files is full, it throws an exception indicating that and the TOE handles it as described.

To correct the issue, a user with administrator rights to the environmental Operating System must login to the local console, correct the storage issue, and restart the TOE.

The TOE does not allow an auditor (or anyone else) to perform any actions if the audit trail is full.

2.1.9.2 Guidance Activities

Modified by TD0287

The evaluator shall examine the operational guidance to ensure it describes what conditions result in the audit trail not being able to be written to, and how an Auditor recognizes that such a condition has occurred. The evaluator shall also examine the operational guidance to ensure it includes remedial steps for correcting such issues.

[CCECG] Section ***5.2.3 FAU_STG.4 Prevention of Audit Data Loss*** states that full audit trail means no audit records can be written to the audit trail table that happens when the hard disk is full.

If the database is inaccessible or the audit trail becomes full (or the disc storage is exhausted), the TOE will display a fatal error message on any attempt to access any of the TOE interfaces, including web interfaces, RAMI, DBAccess, etc. The TOE will record the error message to the local server log file indicating the database error and then shut itself down.

The TOE does not allow an auditor (or anyone else) to perform any actions after shutting down. Auditors trying to access the web interface will receive “Unable to connect” error message from the browser. This error message indicates the TOE is not running, and a fatal error may have occurred. An audit trail cannot be written to, and storage capacity has been reached could be the cause. Auditors should contact the Administrators of the OS platform to locate the fatal error from the local server log file. If the error is due to full disk space, Administrators of the OS platform should allocate more disk spaces to the existing one or migrate the database to a new hard disk with a larger capacity. Once the issue has been fixed, Administrators should follow the steps in [CCECG] Section ***4.1.1 Managing the TOE Service*** to start the TOE.

2.1.9.3 Test Activities

Modified by TD0287

The evaluator shall perform the following tests. The tests are conditional on where the audit data are being stored. Test 1 demonstrates the capability of the TOE to react to an indication that the repository is full; this is always applicable if the audit data are stored locally (either within the TOE boundary or on the TOE platform). If the TOE has a means to detect that a remote audit repository is full, then this test will be run for those types of TOEs as well. Test 2 is only executed in cases where an external repository is supported, and tests the ability of the TOE to detect when the connection to the repository becomes unavailable.

Modified by TD0287

Test 1 (conditional): The evaluator shall cause the audit trail to become full, verify that the TSF behaves as documented in the TSS, and verify that a privileged user can perform the documented remedial steps.

When the TOE’s audit storage space is exhausted the database (either PostgreSQL or HyperSQL) stops operating, which results in the TOE being unable to operate. The evaluator verified this and followed the provided guidance to restore the TOE to operation.

Modified by TD0287

Test 2 (conditional): If the TOE uses a remote repository in the Operational Environment to store audit data, the evaluator shall cause the audit trail to become unavailable, verify that the TSF behaves as documented in the TSS, and verify that a privileged user can perform the documented remedial steps.

N/A, the TOE does not use a remote repository for auditing functionality.

2.1.10 External Audit Trail Storage (FAU_STG_EXT.1)

2.1.10.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the audit storage mechanism from the perspective of the TOE. The TSS must also describe the means by which the audit data are stored locally, or transferred to the external IT entity (and how the trusted channel is provided).

[ST] Section ***9.1.10 FAU_STG_EXT.1 External Audit Trail Storage*** states that the TSF maintains audit data locally in the environmental database and file system which are both located on the same host system as the TOE. The TOE communicates with the database using the Java JDBC API.

2.1.10.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it describes the configuration of any local audit storage mechanism (first two items in the selection in the SFR), including its location and size.

The evaluator shall examine the operational guidance to determine that it describes the relationship between the local audit data (stored inside the TOE boundary and, if applicable, on the TOE platform) and the audit data that are sent to the external IT entity (if applicable). For example, when an audit event is generated, whether it is simultaneously sent to the external IT entity and the local store, or is the local store used as a buffer and “cleared” periodically by sending the data to the external IT entity.

If an external audit server is used, the evaluator shall also examine the operational guidance 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.

[CCECG] Section ***5.2.6 FAU_STG_EXT.1 External Audit Trail Storage*** states that the TOE audit trail is stored in tables in an operational environment database on the same platform as the TOE. Audit records are sent to the database as they are generated. The size of the audit database is restricted by the hard disk space as neither of the database products used in the TOE OE provide the capability to limit or configure the size of the database tables. The database configuration steps are described in the [CCECG] Section ***3.1 Download*** and [CCECG] Section ***3.2 Installation***. [CCECG] Section ***4.8 Using Database Access Service*** describes the DBAccess service including the methods and permissions required to access the TOE via this interface.

2.1.10.3 Test Activities

Testing of the trusted channel mechanism (if the last item is selected in the SFR) will be performed as specified in the associated assurance activities for the particular trusted channel mechanism.

The evaluator shall perform the following tests if the last selection in the SFR is made:

Test 1: [conditional] The evaluator shall establish a connection 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 verify that the connection has been successfully established, 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 TOE does not use an external audit server.

Test 2: [conditional] The evaluator shall examine the audit data transferred to the external audit server in Test 1 and compare it to the locally stored audit data. The evaluator shall verify that the audit records match. If there and any differences, the evaluator shall examine the operational guidance to verify that it explains any discrepancies between locally stored and transmitted audit data.

The TOE does not use an external audit server.

2.2 Communication (FCO)

2.2.1 Certificate-Based Proof of Origin (FCO_NRO_EXT.2)

2.2.1.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the mechanisms used for generating proof of origin and the security-relevant information to which the mechanism applies. The TSS shall describe how the TSF relates the identity and other specified attributes of the originator of the information to the security relevant portions of the information to which the evidence applies. The TSS shall also describe how verification of the proof of origin of information for all security-relevant information is performed and shall also specify the cases in which verification of proof of origin is performed.

For TOEs that only support EST, and do not support revocation requests under either CMC or EST, the TSS must describe the mechanism used to determine whether to revoke certificates.

For TOEs that select “support manual processes for revocation requests and responses,” the evaluator shall ensure the TSS describes those processes.

[ST] ***Section 9.2 Communication (FCO)*** describes how the TOE implements this SFR. The TOE uses the PKCS#11 Cryptographic module to digitally sign certificates, CRLs and OCSP responses it creates using RSA digital signature algorithm with a key size modulus of 3072 and ECDSA with NIST curves: P-256, P384 and P-521.

Except for requests submitted using RAMI, all requests submitted to the TOE must be digitally signed including requests submitted via EST. The TOE requires a valid signature covering the request by a private key matching the public key in the request. For each request submitted, the digital signature on the request is checked, and if not valid the request is rejected. In the case of an RA using RAMI, the RA is responsible for proving the origin of requests it submits, so the TOE supports unsigned certificate requests through RAMI only.

The TOE generates CRLs and provides a built-in OCSP responder for use by relying parties. CRLs and OCSP responses are digitally signed.

The TOE supports EST’s simple enrollment as defined by FIA_ESTS_EXT.1, which does not support revocation. The TOE’s EST responses contain the issued and signed certificate matching the request. The TOE allows a subscriber to request their certificate be revoked using a HTTPS/TLS client authenticated web page. Once the subscriber successfully authenticates to the TOE, using a certificate, the TOE displays a list of certificates issued by the same issuer DN and with the same subject DN as the certificate used to authenticate. The subscriber can then select one or more certificates from that list and revoke them.

2.2.1.2 Guidance Activities

If configurable, evaluator shall examine the operational guidance to ensure it defines how to configure the applicable algorithms used for providing and verifying proof of origin as defined in FCS_COP.1(2).

For TOEs that only support EST, and do not support revocation requests under either CMC or EST, the evaluator shall examine the guidance to ensure it describes support for privileged user functionality as part of this mechanism.

For TOEs that select “support manual processes for revocation requests and responses,” the evaluator shall ensure the operational guidance provides a description of the processes the administrators are to follow. The evaluator shall ensure these are consistent with the descriptions of these processes in the TSS.

[CCECG] Section ***4.5.7.4 Managing Certificate Issuance*** describes how to configure the applicable algorithms used for providing proof of origin. [CCECG] Section ***5.3.1 FCO_NRO_EXT.2 Certificate-based proof of origin*** describes that the TOE implements EST’s simple enrollment. Subscribers connect to the TOE using EST basic authentication or client authentication via HTTP/TLS. Request received via EST must be digitally signed. [CCECG] Section ***4.5.7.1.3 EST (Enrollment over Secure Transport)*** describes configuring EST and other related privileged user functionality.

2.2.1.3 Test Activities

The evaluator shall perform the following tests for each request format selected and for each request supported:

TOE is online (requires establishment of a client capable of generating certificate requests and has a valid HTTPS connection to the TOE):

Test 1: For each supported request, the evaluator shall generate and submit a properly authenticated request to the TOE and verify the responses are signed.

The RAMI portion of this test shows the TOE accepting a client authentication for the RA interface, which led to a successful certification signage.

The EST portion of the test was performed in conjunction with the tests for FIA_ESTS_EXT.1 Test 1(password-based authentication) and Test 3(certificate-based authentication).

Test 2: For each supported request, the evaluator shall generate requests that are unsigned, submit to the TOE, and verify that the TOE rejects the request.

The only supported request type for this SFR is EST. EST and certificate signing requests through the public site require a submission of a Certificate Signing Request (CSR) in order to function. It is not possible to pass an unsigned CSR to the TOE unless it is initiated from a RAMI interface.

Test 3: For each supported request, the evaluator shall generate requests that have an invalid signature based on the RFC, submit to the TOE, and verify that the TOE rejects the request.

For this test the evaluator generated a certificate signing request using a signature algorithm not supported by the TOE. This request was then submitted to the TOE EST/GUI/and RAMI interface. The TOE was shown to reject all three-request interface.

Test 4: For each supported request, the evaluator shall generate requests that are not signed by authorized entities, submit to the TOE, and verify that the TOE rejects the request.

For this test the evaluator submitted a CSR to the TOE’s RAMI and EST interface while providing credentials to accounts that are not known to be an authorized signer. The TOE rejected all of the attempts.

Test 5: For each supported request using password based authentication, the evaluator shall use invalid passwords and verify that the TSF rejects the requests.

The evaluator verified that a CSR sent over EST was rejected if an incorrect password was used for EST authentication.

Test 6: For each proof of possession mode supported, the evaluator shall generate an otherwise valid request but modify the proof of possession value. The evaluator shall submit the modified request and verify that the TSF rejects the request.

Digital Signature is the only proof of possession supported by the TOE. For this test the evaluator modified a byte in the signature block of a PKCS#10 CSR and then submitted it to the TOE over the public interface, EST and RAMI. Audit records indicate the TOE rejected this upload due to an invalid signature.

Transport Test:

Transport Test:

Test 7: For each supported request message, the evaluator shall send an otherwise valid request using HTTP rather than HTTPS and shall verify the TSF rejects the request.

The evaluator attempted to submit an EST and RAMI request over HTTP. The TOE was shown to reject the request.

TOE is offline:

Test 8: With the TOE in offline mode, the evaluator shall log into the TOE locally as the CA Operations Staff role and perform tests 1-4 above.

The TOE does not support an offline mode.

2.3 Cryptographic Support (FCS)

2.3.1 Cryptographic Dependencies (FCS_CDP_EXT.1(a)), (FCS_CDP_EXT.1(b))

2.3.1.1 TSS Activities

> If the TSF invokes interfaces to a cryptographic module in the Operational Environment to provide the necessary cryptographic functionality, the evaluator shall review the TSS to ensure that it specifies the interfaces that are invoked, and the cryptographic provider of the functionality. The evaluator shall review the TSS and verify that all cryptographic SFRs required by the ST—through inclusion of (other) mandatory and optional SFRs--are included.
>
> Other required TSS activities are associated with the cryptographic SFRs themselves.

[ST] Section ***9.3.1 FCS_CDP_EXT.1 Cryptographic Dependencies*** describes the TOE cryptographic dependencies. The TOE both implements cryptographic functionality using its ISC CDK cryptographic module and invokes cryptographic interfaces to a PKCS#11 cryptographic module in the OE. The description includes details of the cryptographic services for which the TOE invokes the HSM and the higher-level cryptographic functions that are invoked. All related mandatory and optional SFRs are included in the ST.

The TOE interfaces with the PKCS#11 Cryptographic Module using the module’s PKCS#11 API. This API is embodied as a shared library (a DLL on Windows and a .so on Linux) that exports a C API as defined by the PKCS#11 specification (also known as Cryptoki). The PKCS#11 API Functions are identified in Table 19.

The PKCS#11 Cryptographic Module performs all sensitive private key operations, and the TOE uses the ISC CDK for everything else.

The ISC CDK implements NIST CAVP validated cryptographic algorithms and NIST CAVP validated cryptographic algorithms and the PKCS#11 Cryptographic Module (Thales TCT T-5000 Luna Network HSM) is FIPS 140-2 certified (certificate # C1999).

2.3.1.2 Guidance Activities

> Required Guidance activities are associated with the cryptographic SFRs themselves.

2.3.1.3 Test Activities

> Required Test activities are associated with the cryptographic SFRs themselves.

2.3.2 Cryptographic Key Generation (FCS_CKM.1(a), FCS_CKM.1(b))

2.3.2.1 TSS Activities

> The evaluator shall ensure that the TSS identifies the key sizes supported. If the ST specifies more than one scheme, the evaluator shall examine the TSS to verify that it identifies the usage for each scheme.

[ST] Section ***9.3.6 FCS_CKM.1(a) Cryptographic Key Generation, FCS_CKM.1(b) Cryptographic Key Generation, FCS_CKM.2 Cryptographic Key Establishment*** identifies the key sizes and the usage for each scheme as shown in the tables below.

Key	Protocol	Information
TLS ECDHE Key	HTTPS/TLS	The TOE uses the ISC CDK to generate these keys as part of the HTTPS/TLS negotiation. The key type is negotiated during session setup and is one of the NIST curves P-256, P-384, or P-521.
TLS Server Credential	HTTPS/TLS	The TOE installer creates an RSA-3072/SHA-384 or an ECDSA P-384/SHA-384 credential for the HTTPS/TLS server using the PKCS#11 Cryptographic Module.
CMS ECDHE Key	CMS	The uses the ISC CDK to generate an ephemeral ECDH key pair when encrypting sensitive data using CMS when the “System” credential is an ECC key pair. The default “System” credential is RSA-3072/SHA-384, however this credential can be changed during or after installation. The allowed key sizes and types are RSA-3072, NIST curve P-256, NIST curve P-384, or NIST curve P-521.

Key	Information
HTTPS/TLS Server Credential	The TOE installer uses the PKCS#11 Cryptographic Module to create an RSA-3072 or ECDSA P-384 credential for HTTPS/TLS server authentication.
Initial Authentication Credentials	The TOE installer uses the ISC CDK to create three initial authentication credentials (certificates and private keys). All are either RSA-3072 or ECDSA P-384.
Issuer Credentials	The TOE uses the PKCS#11 Cryptographic Module to generate RSA or ECC keys for certificate issuance, CRL signing, and OCSP response signing. The allowed key sizes and types are RSA-3072, US-P-256, US-P-384, US-P-521.
“System” Credential	The TOE uses the PKCS#11 Cryptographic Module to generate RSA or ECC keys as the REK to be used when encrypting sensitive data before storing it in the environmental database. The allowed key sizes and types are RSA-3072, US-P-256, US-P-384, US-P-521.
OCSP Signer Credentials	The TOE uses the PKCS#11 Cryptographic Module to generate RSA or ECC keys for OCSP response signing. The allowed key sizes and types are RSA-3072, US-P-256, US-P-384, US-P-521.

2.3.2.2 Guidance Activities

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

[CCECG] Section ***5.4.1 FCS_CKM.1 Cryptographic Key Generation*** states that the TOE installer uses the PKCS#11 cryptographic module in the OE to create an RSA-3072 key (server key) used for TLS/HTTS server authentication, an RSA-3072 credentials for an issuer, and an RSA-3072 key for the initial set of authentication credential. The key type and size of these credentials is either RSA-3072 or US-P-384 and is configurable during the installation. The TOE invokes the PKCS#11 to generate cryptographic keys for certificate issuance, CRL signing, and OCSP response signing. The allowed key sizes are: RSA-3072, US-P-256, US-P-384, and US-P-521.

2.3.2.3 Test Activities

If this requirement is met by the TOE, the evaluator shall verify the implementation of the key generation routines of the supported schemes using the following tests:

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.

As per Section 9.3.3 of the [ST], the network HSM [FCS_CKM.1(a)] was awarded CAVP RSA certificates: C2010 (for RSA 3072), which demonstrate that the TSF performs this function as required. The TOE [FCS_CKM.1(b)] itself was awarded A3042 for ECDSA, ECDH P-256, P-384, P-512.

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.

As per section 9.3.3 of [ST], the TOE was awarded CAVP ECDSA certificate A3042 and the HSM device was awarded CAVP RSA C2010 certificate, which demonstrate that the TSF performs this function as required.

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.

As per section 9.3.3 of [ST], the TOE was awarded CAVP ECDSA certificate A3042 and the HSM device was awarded CAVP RSA C2010 certificate, which demonstrate that the TSF performs this function as required.

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.

Test not applicable.

Added by TD0500

Diffie-Hellman Group 14 and FFC Schemes using "safe-prime" groups

Testing for FFC Schemes using Diffie-Hellman group 14 and/or "safe-prime" groups is done as part of testing in FCS_CKM.2.1.

Test not applicable.

2.3.3 Cryptographic Key Establishment (FCS_CKM.2)

2.3.3.1 TSS 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 (including whether the TOE acts as a sender, a recipient, or both). If Diffie-Hellman group 14 is selected from FCS_CKM.2.1, the TSS shall describe how the implementation meets RFC 3526 Section 3.

[ST] Section ***9.3.6 9.3.6 FCS_CKM.1(a) Cryptographic Key Generation, FCS_CKM.1(b) Cryptographic Key Generation, FCS_CKM.2 Cryptographic Key Establishment*** Table 25 describes the elliptic curve-based key establishment scheme and identifies the NIST curves supported for TLS session establishment.

2.3.3.2 Guidance Activities

The evaluator shall examine the AGD guidance to ensure it contains instructions for configuring the TOE or Operational Environment to use the selected key establishment scheme(s).

[CCECG] Section ***5.4.2 FCS_CKM.2 Cryptographic Key Establishment*** states that the TOE uses the ISC CDK to generate an ephemeral asymmetric cryptographic key for key establishment as a recipient during TLS/HTTPS session establishment and as a sender and recipient when the System credential is an ECC key pair.

For TLS ECDHE Key, the TOE uses the ISC CDK to generate these keys as part of the HTTPS/TLS negotiation. The key type is negotiated during session setup and is one of the NIST curves P-256, P-384, or P-521. These curves have been configured upon installation automatically. These settings are specified in the CA_OPTS variable `(-Djdk.tls.namedGroups=secp256r1,secp384r1,secp521r1)` in the TOE's configuration file `(<ca home>/setenv.sh or setenv.bat)`. To configure the allowed curves, update the CA_OPTS variable, and restart the TOE.

The TOE installer creates an RSA-3072/SHA-384 or an ECDSA P-384/SHA-384 credential for the HTTPS/TLS server using the PKCS#11 Cryptographic Module. To configure a new TLS credential with a different key size, see [CCECG] Section ***4.11 Replacing TLS Credentials***.

The TOE uses the ISC CDK to generate an ephemeral ECDH key pair when encrypting sensitive data using CMS when the System credential is an ECC key pair. The TOE installer creates an RSA-3072/SHA-384 or an

ECDSA P-384/SHA-384 System credential. This system credential can be updated after installation. The allowed key sizes and types are: RSA-3072, US-P-256, US-P-384, and US-P-521. For details on updating the System credentials, see [CCECG] Section ***4.4.8 Managing System Credential***.

NOTE: The TOE supports RSA-4096 and larger key sizes but they were not tested for use in the evaluated configuration.

2.3.3.3 Test Activities

If this requirement is met by the TOE, the evaluator shall verify the implementation of the key generation routines of the supported schemes using the following tests:

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.

As per section 9.3.3 of [ST], the TOE was awarded CAVP certificate A3042 which covered KAS_ECC P-256, P-384, and P-521.

Modified by TD0500.

~~SP800-56B Key Establishment Schemes~~

~~If the TOE acts as a sender, the following assurance activity shall be performed to ensure the proper operation of every TOE supported combination of RSAbased key establishment scheme:~~

a) ~~To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each combination of supported key establishment scheme and its options (with or without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and for each supported mask generation function if KTSOAEP is supported), the tester shall generate 10 sets of test vectors. Each test vector shall include the RSA public key, the plaintext keying material, any additional input parameters if applicable, the MacKey and MacTag if key confirmation is incorporated, and the outputted ciphertext. For each test vector, the evaluator shall perform a key establishment encryption operation on the TOE with the same inputs (in cases where key confirmation is incorporated, the test shall use the MacKey from the test vector instead of the randomly generated MacKey used in normal operation) and ensure that the outputted ciphertext is equivalent to the ciphertext in the test vector.~~

~~If the TOE acts as a receiver, the following assurance activities shall be performed to ensure the proper operation of every TOE supported combination of RSA-based key establishment scheme:~~

~~a) To conduct this test the evaluator shall generate or obtain test vectors from a known good implementation of the TOE supported schemes. For each combination of supported key establishment scheme and its options (with our without key confirmation if supported, for each supported key confirmation MAC function if key confirmation is supported, and for each supported mask generation function if KTSOAEP is supported), the tester shall generate 10 sets of test vectors. Each test vector shall include the RSA private key, the plaintext keying material (KeyData), any additional input parameters if applicable, the MacTag in cases where key confirmation is incorporated, and the outputted ciphertext. For each test vector, the evaluator shall perform the key establishment decryption operation on the TOE and ensure that the outputted plaintext keying material (KeyData) is equivalent to the plaintext keying material in the test vector. In cases where key confirmation is~~

~~incorporated, the evaluator shall perform the key confirmation steps and ensure that the outputted MacTag is equivalent to the MacTag in the test vector.~~

~~b) The evaluator shall ensure that the TSS describes how the TOE handles decryption errors. In accordance with NIST Special Publication 800- 56B, the TOE must not reveal the particular error that occurred, either through the contents of any outputted or logged error message or through timing variations. If KTS-OAEP is supported, the evaluator shall create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800-56B section 7.2.2.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each. If KTS-KEMKWS is supported, the evaluator shall create separate contrived ciphertext values that trigger each of the three decryption error checks described in NIST Special Publication 800- 56B section 7.2.3.3, ensure that each decryption attempt results in an error, and ensure that any outputted or logged error message is identical for each.~~

~~Diffie-Hellman Group 14~~

~~The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_TRP.1/Admin, FTP_TRP.1/Join, FTP_ITC.1 and PT_ITT.1 that uses Diffie-Hellman group 14.~~

Modified by TD0500.

RSAES-PKCS1-v_5 Key Establishment Schemes

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_ITC, FTP_TRP, and FPT_ITT.

Test not applicable, the TOE claimed Elliptic curve-based Key establishments.

Modified by TD0500.

Diffie-Hellman Group 14

The evaluator shall verify the correctness of the TSF's implementation of Diffie-Hellman group 14 by using a known good implementation for each protocol selected in FTP_TRP.1, FTP_ITC.1, and FPT_ITT.1 that uses Diffie-Hellman group 14.

Test not applicable, the TOE claimed Elliptic curve-based Key establishments.

2.3.4 Symmetric Key Generation for DEKs (FCS_CKM_EXT.1(1))

2.3.4.1 TSS Activities

For DEKs generated using an RBG, the evaluator shall examine the TSS of the TOE to verify that it describes, for either the TOE or the Operational Environment, how the functionality described by FCS_RBG_EXT.1 is invoked. The evaluator shall review the TSS and other evidence to determine that the key size being requested from the RBG is identical to the key size used for the encryption/decryption of the data or key.

[ST] Section ***9.3.5 FCS_CKM_EXT.1(1) Symmetric Key Generation for DEKs*** indicates that the TOE generates DEKs of 256-bit using the ISC CDK to encrypt sensitive data prior to storage in the database. The information column in Table 24 describes how the functionality described by FCS_RBG_EXT.1 is invoked. The TOE request key sizes of 256-bit which is identical to the key size used for encryption/decryption of the data.

For each DEK that is formed from a combination by the TSF (that is, “perform” is selected in the first selection, and “combined from KEKs…” is selected in the second selection), the evaluator shall verify that the TSS describes the method of combination and contains a justification for preserving the effective entropy.

[ST] Section ***6.5.6 FCS_CKM_EXT.1(1) Symmetric Key Generation for DEK*** does not claim “combine from KEKs...”. This assurance requirement is not applicable.

2.3.4.2 Guidance Activities

There are no AGD assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.3.4.3 Test Activities

There are no ATE assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.3.5 Key Generation Key Encryption Keys (FCS_CKM_EXT.1(2))

2.3.5.1 TSS Activities

For KEKs generated using an RBG, the evaluator shall examine the TSS of the TOE to verify that it describes how the functionality described by FCS_RBG_EXT.1 is invoked. The evaluator shall review the TSS and other evidence to determine that the key size being requested from the RBG is identical to the key size used for the encryption/decryption of the data or key.

For KEKs generated according to an asymmetric key scheme, the evaluator shall review the TSS to determine that it describes how the functionality described by FCS_CKM.1 is invoked. The evaluator uses the description of the key generation functionality in FCS_CKM.1 or documentation available for the operational environment to determine that the key strength being requested is greater than or equal to 112 bits.

For each KEK that is formed from a combination, the evaluator shall verify that the TSS describes the method of combination and contains a justification for preserving the effective entropy.

[ST] Section ***9.3.15 FCS_CKM_EXT.1(2) Key Generation Key Encryption Keys*** states that the TOE uses the PKCS#11 crypto module in the operational environment to generate an asymmetric key pair designated as the system key. The system key public key is used to encrypt the DEK for the TOE. In the evaluated configuration, the system key is an RSA 3072 bit key, but can be changed using the TOE’s Admin Site to another key type as listed in [ST] Table 17 and described in [ST] Section 9.2.

[ST] Table 19 identifies the functions used to invoke the RBG, and [ST] section 9.3.16 describes how the key size requested from the RBG is identical to the key size used for encryption/decryption of the DEK.

2.3.5.2 Guidance Activities

None defined.

2.3.5.3 Test Activities

None defined.

2.3.6 Cryptographic Key Destruction (FCS_CKM_EXT.4)

2.3.6.1 TSS Activities

> The evaluator shall examine the TSS to ensure it describes each of the secret keys (keys used for symmetric encryption), private keys, and critical security parameters; when they are destroyed (for example, immediately after use, on system shutdown, etc.); and the type of destruction procedure that is performed (overwrite with zeros, overwrite three times with random pattern, etc.). If different types of memory are used to store the materials to be protected, the evaluator shall examine the TSS to ensure it describes the destruction procedure in terms of the memory in which the data are stored.

[ST] Table 27 and Table 28 in section **9.3.7 FCS_CKM_EXT.4(a) Cryptographic Key Destruction, FCS_CKM_EXT.4(b) Cryptographic Key Destruction** identify each of the secret keys, private keys and critical security parameters in the TSF and describes where they reside and how they are destroyed.

2.3.6.2 Guidance Activities

> The evaluator shall examine the AGD guidance to ensure it contains instructions for configuring the TOE to support the required key destruction functionality.

[CCSCG] Section ***5.4.3 FCS_CKM_EXT.4 Cryptographic Key Destruction*** states that the key destruction functionality is not configurable.

By default, the TSF destroys all cryptographic keys (CMS DEK and TLS ECDHE) and sensitive data (PKCS#11 cryptographic PIN, database passwords, and EST subscriber passwords) when no longer needed.

All other keys on which the TOE is dependent are managed and destroyed by the environmental PKCS#11 Cryptographic Module. The TOE’s own key destruction process does not fail as it simply clearing memory allocated by the environmental Operation System using APIs provided by the same.

2.3.6.3 Test Activities

If this requirement is met by volatile memory in the TOE boundary (the second item in the second selection of FCS_CKM_EXT.4.1), the evaluator shall attempt to perform the following tests:

> Test 1: The evaluator shall utilize appropriate combinations of specialized operational environment and development tools (debuggers, simulators, etc.) for the TOE and instrumented TOE builds to test that keys are cleared correctly, including all intermediate copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.
>
> Cryptographic TOE implementations in software shall be loaded and exercised under a debugger to perform such tests. The evaluator shall perform the following test for each key subject to clearing, including intermediate plaintext copies of keys that are subsequently encrypted for storage by the TOE:
>
> 1. Load the instrumented TOE build in a debugger.
> 2. Record the value of the key in the TOE subject to clearing.
> 3. Cause the TOE to perform a normal cryptographic processing with the key from #1.
> 4. Cause the TOE to clear the key.
> 5. Cause the TOE to stop the execution but not exit.
> 6. Cause the TOE to dump the entire memory footprint of the TOE into a binary file.
> 7. Search the content of the binary file created in #4 for instances of the known key value from #1.

The test succeeds if no copies of the key from #1 are found in step #7 above and fails otherwise.

The evaluator shall perform this test on all keys, including those subsequently encrypted for storage, to ensure plaintext intermediate copies are cleared.

The evaluator was provided with a debug version of the TOE. This version dumped the ECDH private key. Memory dumps of the TOE's running processes could then be searched for this file to verify the key was not present. The TOE only generates and destroys a key during an execution of a given function. The function can take less than one second to execute. Because of this, the process of logging the keys and getting a dump of the memory prior to the destruction is impossible to perform.

Test 2: (Conditional) In cases where the TOE is implemented in firmware and operates in a limited operating environment that does not allow the use of debuggers, the evaluator shall utilize a simulator for the TOE on a general purpose operating system. The evaluator shall confirm that keys can be tracked and that destruction occurs. The evaluator shall provide a rationale explaining the instrumentation of the simulated test environment and justifying the obtained test results.

The TOE is not implemented in firmware.

2.3.7 Public Key Integrity (FCS_CKM_EXT.5)

2.3.7.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes each applicable public key, where it is stored and protected, the purpose of the public key, the mechanism used to protect the public key from undetected modification, and the method (for each public key) by which the integrity of the key is checked in accordance with FCS_CKM_EXT.5.2.

[ST] Section ***9.3.8 FCS_CKM_EXT.5 Public Key Integrity***, Table 29 lists all applicable public keys, where they are stored and the mechanism used to protect the public key from undetected modification. All applicable public keys are protected with digital signatures which are verified upon each access to the key.

The following lists the public keys used by the TOE to meet CA requirements that are protected by the TOE or the Operational Environment:

- Issuer certificates
- OCSP signing certificates
- CertAgent/Dhuma trust anchor certificates
- Apache Tomcat trust anchor certificates
- ACL certificates
- HTTPS/TLS Server Key

2.3.7.2 Guidance Activities

There are no AGD assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.3.7.3 Test Activities

NOTE: It might not be possible to directly access certain public keys via the TOE interface in a way that is needed to perform the test below. If that is the case, then the evaluator must describe for each applicable key the interface and indicate why the interface does not allow access to the public keys.

For each public key identified in the TSS, the evaluator shall perform the following test:

Test 1: The evaluator shall attempt to violate the protection of a public key to verify that the action specified in FCS_CKM_EXT.5.2 occurs.

As described in the [CCECG] Section 5.4.11, the TOE does not provide any kind of interface to access or modify keys.

2.3.8 Key Hierarchy Entropy (FCS_CKM_EXT.8)

2.3.8.1 TSS Activities

The evaluator shall examine the TSS to ensure a key hierarchy is described showing the relationship of all KEKs and DEKs formed by combinations or by encrypting one key in another. The evaluator shall confirm that each independent hierarchy is terminated in a REK and that the each REK is generated, stored, and destroyed using hardware-based controls.

The evaluator shall examine the key hierarchy to ensure that the formation of all KEKs and DEKs is described, and that the key sizes match that described by the ST author.

For each KEK or DEK that is formed from a combination, the evaluator shall verify that the TSS describes the method of combination and contains a justification for preserving the effective entropy.

[ST] Section ***9.3.16 FCS_CKM_EXT.8 Key Hierarchy Entropy*** describes the relationship between the REK (System key) and the DEK that the system key is used to encrypt. The system key is an RSA 3072-bit key, that is generated, stored and destroyed by the HSM; the DEKs are 256-bit keys generated randomly using the TOE ISC CDK. The DEKs are used to encrypt the database password and HSM PIN stored in the database. They are created when the HSM password and the database password are encrypted (i.e. during installation or later if the HSM or database password are updated using the Web interface. Each independent hierarchy is terminated in a REK. The formation of the KEK (system key) and the DEKs is described and the key sizes match the SFRs.

2.3.8.2 Guidance Activities

There are no AGD assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.3.8.3 Test Activities

There are no ATE assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.3.9 Cryptographic Operation (AES Encryption/Decryption) (FCS_COP.1(1))

2.3.9.1 TSS Activities

Regardless of whether the requirement is met by the TSF or the TSF in conjunction with the TOE platform, the evaluator shall examine the TSS to ensure that all key encryption and decryption functions use the approved algorithms, modes, and key sizes.

[ST] Section ***9.3.9 FCS_COP.1(1) Cryptographic Operation (AES Encryption/Decryption)*** states that the TOE uses the ISC CDK to perform encryption/decryption during TLS/HTTPS negotiation using AES_CBC or AES-GCM with 256-bit keys. The TOE also uses AES-CBC with 256-bit keys when encrypting data prior to storage using the CMS format and asymmetric encryption of the AES DEK. The key encryption and decryption functions use the approved algorithms, modes and key sizes.

2.3.9.2 Guidance Activities

The evaluator shall examine the AGD guidance to ensure it contains instructions for configuring the TOE or the TOE in conjunction with the Operational Environment for the required encryption algorithms and associated modes and key sizes.

[CCECG] Section ***5.4.4 FCS_COP.1(1) Cryptographic Operation (AES Encryption/Decryption)*** states that the TOE uses the ISC CDK to perform AES-CBC or AES-GCM encryption/decryption with 256-bit keys during TLS/HTTPS negotiation and when storing/retrieving sensitive data in the database. The encryption algorithm, mode, and key size are automatically configured upon installation.

2.3.9.3 Test Activities

The following tests shall be performed for functionality implemented by the TSF.

AES-CBC Tests

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 5 plaintext values for each key size selected 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 an all-zeros key of length equal to the selected key size, for each key size selected.

To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using the 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 5 key values for each key size selected 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 of length equal to the selected key size, for each key size selected.

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 a set of key values described below for each key size selected 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 keys in each set shall have the same length as the selected key size, for each key size, N. 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 key 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. Each set of key/ciphertext pairs shall have N N-bit key/ciphertext pairs, and the second set of key/ciphertext pairs for selected key size, N. 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 ciphertext values that result from AES-CBC encryption of the given plaintext using a key value of all zeros of length equal to the selected key size with an IV of all zeros for each key size selected. 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 100 plaintext, IV, and key 3-tuples for each selected key size. 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.

As per section 9.3.3 of [ST], the TOE was awarded CAVP AES-CBC certificates: A3042 which demonstrates that the TSF performs this function as required.

AES-CCM Tests

The evaluator shall test the generation-encryption and decryption-verification functionality of AES-CCM for the following input parameter and tag lengths:

Each selected key length

Two payload lengths. One payload length shall be the shortest supported payload length, greater than or equal to zero bytes. The other payload length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits).

Two or three associated data lengths. One associated data length shall be 0, if supported. One associated data length shall be the shortest supported payload length, greater than or equal to zero bytes. One associated data length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits). If the implementation supports an associated data length of 216 bytes, an

Nonce lengths. All supported nonce lengths between 7 and 13 bytes, inclusive, shall be tested.

Tag lengths. All supported tag lengths of 4, 6, 8, 10, 12, 14 and 16 bytes shall be tested.

To test the generation-encryption functionality of AES-CCMP, the evaluator shall perform the following four tests:

Test 1. For EACH supported key and associated data length and ANY supported payload, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

Test 2. For EACH supported key and payload length and ANY supported associated data, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

Test 3. For EACH supported key and nonce length and ANY supported associated data, payload and tag length, the evaluator shall supply one key value and 10 associated data, payload and nonce value 3-tuples and obtain the resulting ciphertext.

Test 4. For EACH supported key and tag length and ANY supported associated data, payload and nonce length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

To determine correctness in each of the above tests, the evaluator shall compare the ciphertext with the result of generation-encryption of the same inputs with a known good implementation.

To test the decryption-verification functionality of AES-CCM, for EACH combination of supported associated data length, payload length, nonce length and tag length, the evaluator shall supply a key value and 15 nonce, associated data and ciphertext 3-tuples and obtain either a FAIL result or a PASS result with the decrypted payload. The evaluator shall supply 10 tuples that should FAIL and 5 that should PASS per set of 15.

Additionally, the evaluator shall use tests from the IEEE 802.11-02/362r6 document "Proposed Test vectors for IEEE 802.11 TG", dated September 10, 2002, Section 2.1 AES-CCMP Encapsulation Example and Section 2.2 Additional AES CCMP Test Vectors to further verify the IEEE 802.11-2007 implementation of AES-CCMP.

Test is not applicable to this evaluation.

AES-Galois\Counter Mode (GCM) Monte Carlo Test

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

Each selected key length

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.

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.

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.

As per section 9.3.3 of [ST], the TOE was awarded CAVP AES-GCM certificate A3042 which demonstrates that the TSF performs this function as required.

XTS-AES Monte Carlo Test

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

Each selected key length

Three data unit (i.e., plaintext) lengths. One of the data unit lengths shall be a non-zero integer multiple of 128 bits, if supported. One of the data unit lengths shall be an integer multiple of 128 bits, if supported. The third data unit length shall be either the longest supported data unit length or 216 bits, whichever is smaller.

Using a set of 100 key, plaintext and 128-bit random tweak value 3-tuples and obtain the ciphertext that results from XTS-AES encrypt.

The evaluator may supply a data unit sequence number instead of the tweak value if the implementation supports it. The data unit sequence number is a base-10 number ranging between 0 and 255 that implementations convert to a tweak value internally.

The evaluator shall test the decrypt functionality of XTS-AES using the same test as for encrypt, replacing plaintext values with ciphertext values and XTS-AES encrypt with XTS-AES decrypt.

Test not applicable.

AES Key Wrap (AES-KW) and Key Wrap with Padding (AES-KWP) Test

The evaluator shall test the authenticated encryption functionality of AES-KW for EACH combination of the following input parameter lengths:

Each selected key length

Three plaintext lengths. One of the plaintext lengths shall be two semi-blocks (128 bits). One of the plaintext lengths shall be three semi-blocks (192 bits). The third data unit length shall be the longest supported plaintext length less than or equal to 64 semi-blocks (4096 bits).

Using a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KW authenticated encryption. To determine correctness, the evaluator shall use the AES-KW authenticated-encryption function of a known good implementation.

The evaluator shall test the authenticated-decryption functionality of AES-KW using the same test as for authenticated-encryption, replacing plaintext values with ciphertext values and AES-KW authenticated-encryption with AES-KW authenticated-decryption.

The evaluator shall test the authenticated-encryption functionality of AES-KWP using the same test as for AES-KW authenticated-encryption with the following change in the three plaintext lengths:

One plaintext length shall be one octet. One plaintext length shall be 20 octets (160 bits).

One plaintext length shall be the longest supported plaintext length less than or equal to 512 octets (4096 bits).

The evaluator shall test the authenticated-decryption functionality of AES-KWP using the same test as for AES-KWP authenticated-encryption, replacing plaintext values with ciphertext values and AES-KWP authenticated-encryption with AES-KWP authenticated-decryption.

Test not applicable.

2.3.10 Cryptographic Operation (Cryptographic Signature) FCS_COP.1(2)(a), FCS_COP.1(2)(b)

2.3.10.1 TSS Activities

Regardless of whether the requirement is met by the TSF or TOE platform, the evaluator shall examine the TSS to ensure that all signature generation and verification functions use the approved algorithms and key sizes.

[ST] Section 9.3.10 includes Table 30 which identifies every instance where the TSF performs signature services. It identifies when the TOE itself performs these services and when the TOE in conjunction with the operational environment perform these services. The TSS describes how the cryptographic services are invoked. All signature generation and verification functions use the approved algorithms and key sizes.

2.3.10.2 Guidance Activities

The evaluator shall examine the AGD guidance to ensure it contains instructions for configuring the TOE or the TIE in conjunction with the Operational Environment for the required signature algorithms and associated modes and key sizes.

[CCECG] Section ***5.4.5 FCS_COP.1(2) Cryptographic Operation (Cryptographic Signature)*** states that by default, the TOE supports both RSA and ECDSA algorithms. The algorithm and key size for the specified operations are determined solely by the key type and size of the public/private key that is performing the operation. For RSA key pairs, the RSA algorithm will be used. For ECC key pairs, the ECDSA algorithm will be used. To change the key type/size used you must change the credential.

The System, Issuer, TLS, Authentication, and delegated OCSP signer credentials generated during the installation are either RSA-3072/SHA-384 or ECDSA P-384/SHA-384. However, these credentials can be changed post-installation. To change the Issuer credential that signs certificates, CRLs, and OCSP responses, see [CCECG] Section ***4.5.3 Managing CA Credentials***.

To change the System credential that encrypts sensitive data and signs Trust Anchor and CRL databases, see section [CCECG] Section ***4.4.8 Managing System Credential***.

To change the TLS credential, see [CCECG] section ***4.11 Replacing TLS Credentials***.

To change the delegated OCSP signer credential that signs OCSP responses, see [CCECG] Section ***4.5.7.6.1.2 Generating a New Delegated Signer Credential*** and section [CCECG] Section ***4.4.10.2.1 Generating a New Delegated OCSP Signer Credential***.

2.3.10.3 Test Activities

The following tests shall be performed for functionality implemented by the TSF.

Key Generation:

Key Generation for RSA Signature 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:

- Random Primes:
    - Provable primes
    - Probable primes
- 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.

As per section 9.3.3 of [ST], the TOE was awarded CAVP RSA SigGen and RSAS Sig Ver A3042 and the HSM device was awarded CAVP C2010, which demonstrates that the TSF performs this function as required. The HSM

ECDSA Key Generation Tests

FIPS 186-4 ECDSA 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.

As per section 9.3.3 of [ST], the TOE was awarded CAVP ECDSA SigGen and ECDSA Sig Ver certificates A3042 and the HSM device was awarded CAVP cert C1999 for both ECDSA SignGen and SignVer, which demonstrates that the TSF performs this function as required.

Certificate A3042 demonstrates the FIPS 186-4 Public Key Verification (PKV) Test.

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.

As per section 9.3.3 of [ST], the TOE was awarded CAVP ECDSA SigGen and ECDSA Sig Ver certificates A3042 and the HSM device was awarded CAVP cert C1999 for both ECDSA SignGen and SignVer, which demonstrates that the TSF performs this function as required.

Certificate A3042 demonstrates the FIPS 186-4 Public Key Verification (PKV) Test.

RSA Signature Algorithm Tests

Signature Generation Test

The evaluator shall verify the implementation of RSA Signature Generation by the TOE using the Signature Generation Test. To conduct this test the evaluator must generate or obtain 10 messages from a trusted reference implementation for each modulus size/SHA combination supported by the TSF. The evaluator shall have the TOE use their private key and modulus value to sign these messages.

The evaluator shall verify the correctness of the TSF's signature using a known good implementation and the associated public keys to verify the signatures.

Signature Verification Test

The evaluator shall perform the Signature Verification test to verify the ability of the TOE to recognize another party's valid and invalid signatures. The evaluator shall inject errors into the test vectors produced during the Signature Verification Test by introducing errors in some of the public keys e, messages, IR format, and/or signatures. The TOE attempts to verify the signatures and returns success or failure.

The evaluator shall use these test vectors to emulate the signature verification test using the corresponding parameters and verify that the TOE detects these errors.

As per section 9.3.3 of [ST], the TOE was awarded CAVP RSA SigGen and RSAS Sig Ver certificate A3042 which demonstrates that the TSF performs this function as required. The HSM device was also awarded C2010 for the RSA Sign.

2.3.11 Cryptographic Operation (Cryptographic Hashing) (FCS_COP.1(3))

2.3.11.1 TSS Activities

Regardless of whether the requirement is met by the TSF or TOE platform, the evaluator shall examine the TSS to ensure that all hash functions use the approved algorithms, modes and key sizes.

[ST] Section ***9.3.11 FCS_COP.1(3) Cryptographic Operation (Cryptographic Hashing)*** describes how the TOE provides cryptographic hashing services. The TOE uses its ISC CDK to perform cryptographic hashing when establishing TLS/HTTS connections, when creating certificates, certificate requests, CRLs and OCSP responses. SHA-384 and SHA-512 are supported when creating certificates, certificate request and CRLs. SHA-256, SHA-384, SHA-512 are supported when creating OCSP responses and when verifying certificates, CRLs and certificate requests. The application note in Section 6.5.16 indicates that all modes are byte oriented. All signature generation and verification functions use the approved algorithms and key sizes.

2.3.11.2 Guidance Activities

The evaluator shall examine the AGD guidance to ensure it documents how to configure the TOE or the TOE in conjunction with the Operational Environment for the required hash sizes. The AGD guidance shall also include instructions for disabling deprecated algorithms.

[CCECG] Section ***5.4.6 FCS_COP.1(3) Cryptographic Operation (Cryptographic Hashing)*** states that the TOE uses the ISC CDK to perform cryptographic hashing as follows:

- When establishing a TLS/HTTPS connection
- Supports the TLS 1.2 PRF with SHA-256, SHA-384, and corresponding message digest sizes 256-, 384-bits
  - Supports SHA-256, SHA-384, and SHA-512 for signature validation as required by TLS 1.2
  - Supports SHA-256, SHA-384, SHA-512 for signature generation as required by TLS 1.2
- When digitally signing the Trust Anchor and ACL database tables for integrity protection, SHA-384 is used (which is not configurable).

- When creating certificates, certificate requests, and CRLs, SHA-384, and SHA-512 and message digest size 384 and 512 are supported.
- For details on configuring the message digest for certificates, see sections 4.5.3.1 Generating Credential for a Root CA and 4.5.7.4.1 Properties.
- For details on configuring the message digest for certificate requests, see section 4.5.3.2 Generating Credential for Subordinate CA.
- For details on configuring the message digest for CRLs, see section 4.5.7.5 Managing CRL Issuance.
- When creating OCSP responses, SHA-256, SHA-384, SHA-512, and message digest sizes 256, 384, and 512 are supported. For details on configuring the message digest, see section 4.5.7.6 Managing OCSP Responder Settings for internal issuers and section 4.4.10.2.4 Configuring OCSP Response Settings for external issuers.
- When verifying certificates, CRLs, and certificate requests, SHA-256, SHA-384, SHA-512, and message digest sizes256, 384, and 512 are supported.

For details on configuring the message digest for certificates, see [CCECG] Sections ***4.5.3.1 Generating Credential for a Root CA*** and ***4.5.7.4.1 Properties***.

For details on configuring the message digest for certificate requests, see [CCECG] Section ***4.5.3.2 Generating Credential for Subordinate CA***.

For details on configuring the message digest for CRLs, see [CCECG] Section ***4.5.7.5 Managing CRL Issuance.***

When creating OCSP responses, SHA-256, SHA-384, SHA-512, and message digest sizes 160, 256, 384, and 512 are supported. For details on configuring the message digest, see [CCECG] Section ***4.5.7.6 Managing OCSP Responder Settings*** for internal issuers and section 4.4.10.2.4 Configuring OCSP Response Settings for external issuers.

When verifying certificates, CRLs, and certificate requests, SHA-256, SHA-384, SHA-512, and message digest sizes 160, 256, 384, and 512 are supported.

2.3.11.3 Test Activities

If this requirement is met by the TOE, 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 evaluator shall 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 evaluator shall 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 evaluator shall 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 evaluator shall 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 evaluator shall devise an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 99*i, where 1 ≤ i ≤ m. The message text shall be pseudorandomly generated. The evaluator shall 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 evaluator shall devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. The message text shall be pseudorandomly generated. The evaluator shall 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 evaluator shall 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 evaluator shall then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluator shall then ensure that the correct result is produced when the messages are provided to the TSF.

As per section 9.3.3 of the [ST], the TOE was awarded CAVP Cert A3042 for the use of SHA-256, SHA-384, and SHA-512 hash algorithms. The HSM device was also awarded CAVP cert C1999 for the use of SHA-256, SHA-384, and SHA-512 hashing algorithms.

2.3.12 Cryptographic Operation (Keyed-Hash Message Authentication) (FCS_COP.1(4))

2.3.12.1 TSS Activities

Regardless of whether the requirement is met by the TSF or TOE platform, the evaluator shall examine the TSS to ensure that all keyed hash functions use the approved algorithms and key sizes.

[ST] Section ***9.3.12 Cryptographic Operation (Keyed-Hash Message Authentication)*** indicates that the TOE uses the ISC CDK to perform keyed-hash message authentication when establishing TLS/HTTPS connections; when performing PBKDF2 for creating the EST password check values; and when generating random numbers. The TOE uses HMAC-SHA-384 for TLS/HTTPS connections and HMAC-256 when generating random numbers and when creating the EST password check values.

2.3.12.2 Guidance Activities

The evaluator shall examine the AGD guidance to ensure it contains instructions for configuring the TOE or the TOE in conjunction with the Operational Environment for the required hash sizes and message digest sizes.

[CCECG] Section ***5.4.7 FCS_COP.1(4) Cryptographic Operation (Keyed-Hash Message Authentication)*** states that the TOE does not provide an option to configure key-hashed message authentication.

2.3.12.3 Test Activities

If this requirement is met by the TOE, the evaluator shall perform the following test:

Test 1: 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 IV using a known good implementation.

As per section 9.3.3 of [ST], the TOE was awarded CAVP HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512 certificate A3042 which demonstrates that the TSF performs this function as required.

2.3.13 Cryptographic Operation (Password-Based Key Derivation Function) (FCS_COP.1(5))

2.3.13.1 TSS Activities

If this SFR is implemented by the TSF, then the evaluator shall perform the following activities.

The evaluator shall check that the TSS describes the method by which the password is first encoded and then fed to the SHA algorithm. The settings for the algorithm (padding, blocking, etc.) shall be described, and the evaluator shall verify that these are supported by the selections in this component as well as the selections concerning the hash function itself. The evaluator shall verify that the TSS contains a description of how the output of the hash function is used to form the submask that will be input into the function and is the same length as the DEK or KEK being protected.

For the NIST SP 800-132-based conditioning of the passphrase, the required assurance activities will be performed when doing the assurance activities for the appropriate requirements (FCS_COP.1.1(4)). If any manipulation of the key is performed in forming the submask that will be used to form the KEK, that process shall be described in the TSS.

[ST] Section ***9.3.4 FCS_COP.1(5) Cryptographic Operation (Password-Based Key Derivation Function)*** states that the TOE uses PBKDF2 to create the check values for EST user passwords. It uses HMAC_SHA-256, an 16-byte salt, and 20,000 iterations.

The TSS also notes that FCS_COP.1(5) requirement is included solely because the TOE uses PBKDF2 to store a check value of EST user passwords as described in FCS_COP.1(4). PBKDF2 is not used to enforce access by privileged users; and it is not used for encryption. EST users are not privileged users.

The TOE uses the input password directly as the HMAC key without any conditioning. Although the output is not used as a key, the output of the hash function is used within the HMAC construct and thus the resulting output is a 256-bit value that could be used as a 256-bit key.

2.3.13.2 Guidance Activities

There are no AGD assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.3.13.3 Test Activities

There are no ATE assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.3.14 HTTPS Protocol (FCS_HTTPS_EXT.1)

2.3.14.1 TSS Activities

> The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish protected communications with remote IT entities, focusing on when client authentication is required

[ST] Section ***9.3.14 FCS_HTTPS_EXT.1 HTTPS, FCS_TLSS_EXT.1 TLS Server Protocol, FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** describes how the TSF uses TLS/HTTPS to establish protected communication for access to the admin, CA and other sites. Client authentication is required when user access to the admin site and to the CA site.

[ST] Section ***9.9 Trusted Path/Channels (FTP)*** describes when client authentication is required.

2.3.14.2 Guidance Activities

> None defined.

2.3.14.3 Test Activities

> Testing for this activity is done as part of the TLS testing.

2.3.15 Cryptographic Random Bit Generation (FCS_RBG_EXT.1)

2.3.15.1 TSS Activities

> The evaluator shall examine the TSS to ensure it describes the deterministic random bit generation services provided by either the TSF or the Operational Environment, including a description of the entropy source.

[ST] Section ***9.3.13 FCS_RBG_EXT.1 Cryptographic Random Bit Generation*** identifies the deterministic random bit generation services provided by the TOE and the HSM in the operational environment. When the TOE uses the HMAC_DRBG (SHA-256) in its ISC CDK component, the entropy source is third-party software based noise source providing more than 256-bits of entropy. When the TOE uses the CTR_DRBG (AES-256) from the Network HSM required on its host platform, the entropy source is hardware-based noise source providing 256-bits of entropy.

The entropy document explains, in detail, the sources and amounts of entropy.

2.3.15.2 Guidance Activities

> If any part of the deterministic RBG services is configurable, the evaluator shall ensure that the operational guidance provides clear instructions for how to configure them, including those that pertain to the Operational Environment, if applicable.

[CCECG] Section ***5.4.8 FCS_RBG_EXT.1 Cryptographic Random Bit Generation*** states that the TOE uses the two cryptographic modules to generate random numbers in the following ways.

- The PKCS#11 Cryptographic Module

  - Hash_DRBG(SHA-256)
  - Provider: Thales TCT T-5000 Luna Network HSM
- The TOE using the ISC CDK
  - HMAC_DRBG(SHA-256)
  - Provider: ISC CDK

There is no option in the TOE to configure the RBG service.

2.3.15.3 Test Activities

> Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Annex D, Entropy Documentation and Assessment, regardless of whether the entropy source is implemented by the TOE or the Operational Environment. Note that this is only applicable if the TOE implements or directly invokes the DRBG. If this is not the case, then FCS_RBG_EXT.1 should not be included in the ST, as outlined in the application note for FCS_CDP_EXT.1.

The vendor produced the ISC CertAgent Entropy Dependency Statement which is considered proprietary.

> For RBG implementations in the TSF, the evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.

> **Implementations Conforming to NIST Special Publication 800-90A**
>
> The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RBG functionality.
>
> If the RBG 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 SP 800-90A).
>
> If the RBG 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 df 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.

As per section 9.3.3 of [ST], the TOE was awarded the following CAVP certificates which demonstrates that the TSF performs this function as required:

The PKCS#11 Cryptographic Module

- Hash_DRBG(SHA-256)– CAVP Certificate DRBG 349

The TOE using the ISC CDK

- HMAC_DRBG(SHA-256)– CAVP Certificate A3042

2.3.16 Cryptographic Key Storage (FCS_STG_EXT.1)

2.3.16.1 TSS Activities

Regardless of whether this requirement is met by the TOE or the Operational Environment, the evaluator will check the TSS to ensure that it lists each persistent secret and private key needed to meet the requirements in the ST. For each of these items, the evaluator will confirm that the TSS lists for what purpose it is used, and how it is stored.

[ST] Section ***9.3.2 FCS_STG_EXT.1 Cryptographic Key Storage*** Table 20 identifies the persistent private and secret keys that are stored in the HSM in the TOE operational environment as well as the encryption keys stored in the TOE database. Table 20 identifies each key, describes the purpose of each and specifies where key is stored either by the HSM or in the TOE database tables.

2.3.16.2 Guidance Activities

There are no AGD assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.3.16.3 Test Activities

There are no ATE assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.3.17 TLS Server Protocol (FCS_TLSS_EXT.1)

2.3.17.1 TSS Activities

Modified by TD0294

FCS_TLSS_EXT.1.1

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.

[ST] Section ***9.3.14 FCS_HTTPS_EXT.1 HTTPS, FCS_TLSS_EXT.1 TLS Server Protocol, FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** identifies the following supported ciphersuites:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

The ciphersuites specified in the TSS are identical to those listed in the SFR.

Modified by TD0294

FCS_TLSS_EXT.1.2

The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS versions.

[ST] Section ***9.3.14 FCS_HTTPS_EXT.1 HTTPS, FCS_TLSS_EXT.1 TLS Server Protocol, FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** states that connections requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, or TLS 1.1 are rejected.

Modified by TD0294

FCS_TLSS_EXT.1.3

The evaluator shall verify that the TSS describes the key agreement parameters of the server key exchange message.

[ST] Section ***9.3.14 FCS_HTTPS_EXT.1 HTTPS, FCS_TLSS_EXT.1 TLS Server Protocol, FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** states that the only supported elliptic curve key agreement parameters are NIST curves secp256r1, secp384r1, and secp521r1 also known as P-256, P-384, P-521, US-P-256, US-P-384, and US-P-521.

2.3.17.2 Guidance Activities

Modified by TD0294

FCS_TLSS_EXT.1.1

The evaluator shall also check the operational guidance 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).

FCS_TLSS_EXT.1.2

The evaluator shall verify that any configuration necessary to meet the requirement are contained in the AGD guidance.

FCS_TLSS_EXT.1.3

The evaluator shall verify that any configuration necessary to meet the requirement is contained in the AGD guidance.

[CCECG] Section ***5.4.9 FCS_TLSS_EXT.1 TLS Server Protocol*** states that the TOE supports the following ciphersuites:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

The default, initial, TLS/HTTPS server key is RSA-3072. The TLS/HTTPS server key can be replaced with an RSA-4096, or larger or an ECC key pair if desired, but this was not tested for use in the evaluated configuration. The only supported elliptic curve parameters are NIST curves secp256r1, secp384r1, and secp521r1, also known as P-256, P-384, P-521, US-P-256, US-P-384, and US-P-521.

Only TLS 1.2 is supported in the evaluated configuration. Connections requesting SSL 2.0, SSL 3.0, TLS 1.0 or TLS 1.1 are rejected.

TLS 1.2 and the ciphersuites have been configured in Tomcat upon installation automatically. These settings are specified in the `ciphers=" TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 "` and `sslEnabledProtocols="TLSv1.2"` attributes of the Connector elements in the Tomcat configuration file (`<ca home>/tomcat/conf/server.xml`). No configuration is required.

2.3.17.3 Test Activities

Modified by TD0294

FCS_TLSS_EXT.1.1

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 EAP 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 in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

The evaluator successfully opened a TLS connection to the TOE with a TLS test client using each of the ciphersuites listed in the ST.

Modified by TD0294

FCS_TLSS_EXT.1.1

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.

The evaluator used a TLS test client to attempt to open a connection to the TOE using an unsupported ciphersuite as well as with TLS_NULL_WITH_NULL_NULL. Both of these connection attempts resulted in an error from the TOE and a rejected connection.

Modified by TD0294

FCS_TLSS_EXT.1.1

Test 3: The evaluator shall use a client to send a key exchange message in the TLS connection that the does not match the server-selected ciphersuite (for example, send an ECDHE key exchange while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or send a RSA key exchange while using one of the ECDSA ciphersuites.) The evaluator shall verify that the TOE disconnects after the receiving the key exchange message.

The evaluator verified that after using a mismatched cipher and key exchange pair, the TOE terminated the unexpected session.

Modified by TD0294 and TQ1660

FCS_TLSS_EXT.1.1

Test 4: The evaluator shall perform the following modifications to the traffic:

~~a) Modify at a byte in the client's nonce in the Client Hello handshake message, and verify that the server rejects the client's Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message.~~

~~b) [conditional] If an ECDHE or DHE ciphersuite is selected, modify the signature block in the Client's Key Exchange handshake message, and verify that the server rejects the client's Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message.~~

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

d) After generating a fatal alert by sending a Finished message from the client before the client sends a ChangeCipherSpec message, send a Client Hello with the session identifier from the previous test, and verify that the server denies the connection.

e) Send a garbled message from the client after the client has issued the ChangeCipherSpec message and verify that the Server denies the connection.

c) A TLS test client was used to open a connection to the TOE. A byte was modified in the Client Finished message which resulted in the TOE terminating the connection.

d) The evaluator verified that the TOE rejects an attempt to resume a previous session that was terminated with a fatal alert.

e) A TLS test client was used to send a garbled message to the TOE after the ChangeCipherSpec message. The TOE was observed to reject the connection attempt.

Modified by TD0294

FCS_TLSS_EXT.1.2

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.

Connection attempts to the TOE were made with all versions of TLS and SSL older than TLS 1.2. The TOE rejected all of these connection attempts.

> **Modified by TD0294**
>
> **FCS_TLSS_EXT.1.3**
>
> If the second selection includes any choice other than “no other”, the evaluator shall attempt a connection using an ECDHE ciphersuite and a configured curve and, using a packet analyzer, verify that the key agreement parameters in the Key Exchange message are the ones configured. (Determining that the size matches the expected size for the configured curve is sufficient.) The evaluator shall repeat this test for each supported NIST Elliptic Curve and each supported Diffie-Hellman key size.

The evaluator attempted to open a connection with each of the elliptic curves claimed in the ST. All of these attempts were successful.

2.3.18 TLS Server Protocol with Mutual Authentication (FCS_TLSS_EXT.2)

2.3.18.1 TSS Activities

> **Modified by TD0294**
>
> **FCS_TLSS_EXT.2.1**
>
> 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.

[ST] Section ***9.3.14 FCS_HTTPS_EXT.1 HTTPS, FCS_TLSS_EXT.1 TLS Server Protocol, FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** identifies the following supported ciphersuites:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

The ciphersuites specified in the TSS are identical to those listed in the SFR.

> **Modified by TD0294**
>
> **FCS_TLSS_EXT.2.2**
>
> The evaluator shall verify that the TSS contains a description of the denial of old SSL and TLS versions.

[ST] Section ***9.3.14 FCS_HTTPS_EXT.1 HTTPS, FCS_TLSS_EXT.1 TLS Server Protocol, FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** states that connections requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, or TLS 1.1 are rejected.

> **Modified by TD0294**
>
> **FCS_TLSS_EXT.2.3**
>
> The evaluator shall verify that the TSS describes the key agreement parameters of the server key exchange message.

[ST] Section ***9.3.14 FCS_HTTPS_EXT.1 HTTPS, FCS_TLSS_EXT.1 TLS Server Protocol, FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** states that the only supported elliptic curve key agreement parameters are NIST curves secp256r1, secp384r1, and secp521r1 also known as P-256, P-384, P-521, US-P-256, US-P-384, and US-P-521.

Modified by TD0294

FCS_TLSS_EXT.2.4

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.

[ST] Section ***9.3.14 FCS_HTTPS_EXT.1 HTTPS, FCS_TLSS_EXT.1 TLS Server Protocol, FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** states that mutual authentication using valid X.509 certificates is required for access to the Admin Site, CA Account Site, RAMI, DBAccess, EST using certificate-based authentication, and the self-service portion of the Public Site while the remainder of the Public Site and EST without certificate-based authentication do not require mutual authentication. OCSP is available without mutual authentication over HTTPS/TLS or without any security over HTTP.

Modified by TD0294

FCS_TLSS_EXT.2.5

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.

[ST] Section ***9.3.14 FCS_HTTPS_EXT.1 HTTPS, FCS_TLSS_EXT.1 TLS Server Protocol, FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** states that mutual authentication using valid X.509 certificates is required for access to the Admin Site, CA Account Site, RAMI, DBAccess, EST using certificate-based authentication, and the self-service portion of the Public Site while the remainder of the Public Site and EST without certificate-based authentication do not require mutual authentication. OCSP is available without mutual authentication over HTTPS/TLS or without any security over HTTP.

[ST] Section ***9.5.2 FIA_X509_EXT.2 Certificate-Based*** Authentication states when the TOE cannot determine the validity of a certificate the TOE will not accept the certificate. If the certificate used for HTTPS authentication is not successfully validated, the connection will either be terminated (a TLS bad certificate error will be sent to the client), or the session will be established but the user will be shown an error page (an HTML error page will be sent to the client) and will be denied access.

Modified by TD0294

FCS_TLSS_EXT.2.6

The evaluator shall verify that the TSS describes how the DN or SAN in the certificate is compared to the expected identifier.

[ST] Section ***9.3.14 FCS_HTTPS_EXT.1 HTTPS, FCS_TLSS_EXT.1 TLS Server Protocol, FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** states that the TOE supports filtering client certificates by their distinguished name (DN) in order to allow an administrator to restrict access to only matching certificates using a wildcard specification. If a client certificate’s DN does not match the configured filter the TOE responds with a fatal TLS error.

2.3.18.2 Guidance Activities

Modified by TD0294

FCS_TLSS_EXT.2.1

The evaluator shall also check the operational guidance 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).

FCS_TLSS_EXT.2.2
The evaluator shall verify that any configuration necessary to meet the requirement is contained in the AGD guidance.

FCS_TLSS_EXT.2.3
The evaluator shall verify that any configuration necessary to meet the requirement is contained in the AGD guidance.

FCS_TLSS_EXT.2.4
The evaluator shall verify that any configuration necessary to meet the requirement is contained in the AGD guidance.

FCS_TLSS_EXT.2.5
The evaluator shall verify that any configuration necessary to meet the requirement is contained in the AGD guidance.

FCS_TLSS_EXT.2.6
For each service using mutual authentication TLS which does not automatically determine the expected identifier, the evaluator shall verify that the AGD guidance includes instructions on configuring the expected identifier.

[CCECG] Section ***5.4.10 FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication*** states that the TOE supports the following ciphersuites:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

Only TLS 1.2 is supported in the evaluated configuration. Connections requesting SSL 2.0, SSL 3.0, TLS 1.0 or TLS 1.1 are rejected.

TLS 1.2 and the ciphersuites have been configured in Tomcat upon installation automatically. These settings are specified in the `ciphers=" TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 " and sslEnabledProtocols="TLSv1.2"` attributes of the Connector elements in the Tomcat configuration file (`<ca home>/tomcat/conf/server.xml`). No configuration is required.

The only supported elliptic curve parameters are NIST curves secp256r1, secp384r1, and secp521r1 also known as P-256, P-384, P-521, US-P-256, US-P-384, and US-P-521.

Upon TOE installation, client and root credentials have been created with certificates fulfilling the above requirements. The TOE has been configured with the root certificate and CRL installed in the trust anchor database and CRL store respectively. Tomcat has been configured with the root certificate installed in the trust keystore and to open the TLS port with mutual authentication for Admin and CA sites. No configuration is required.

If the client certificate cannot fulfill any of the above requirements, the certificate is invalid, and the TOE will not establish a trusted channel. No configuration is required.

The TOE supports filtering client certificates by their distinguished name (DN) to allow an administrator to restrict access to only matching certificates. If a client certificate's DN does not match the configured filter, the TOE will respond with a fatal TLS error. This filter applies to all interfaces using mutual authentication. By default, the filter is set to "*" to allow any DNs. For details on configuring the DN filter, see [CCECG] Section ***4.4.2.2.3 Client Certificate DN Filter***.

2.3.18.3 Test Activities

Modified by TD0294

FCS_TLSS_EXT.2.1

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 EAP 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 in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).

This requirement was tested in conjunction with FCS_TLSS_EXT.1.1 Test 1. In that test, the evaluator was able to confirm that the TOE can establish a TLS connection with the supported cipher suites.

Modified by TD0294

FCS_TLSS_EXT.2.1

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.

This requirement was tested in conjunction with FCS_TLSS_EXT.1.1 Test 2. In that test, the evaluation confirmed that the TOE will reject any non-claimed and invalid cipher suites.

Modified by TD0294

FCS_TLSS_EXT.2.1

Test 3: The evaluator shall use a client to send a key exchange message in the TLS connection that the does not match the server-selected ciphersuite (for example, send an ECDHE key exchange while using the TLS_RSA_WITH_AES_128_CBC_SHA ciphersuite or send a RSA key exchange while using one of the ECDSA ciphersuites.) The evaluator shall verify that the TOE disconnects after the receiving the key exchange message.

This requirement was tested in conjunction with FCS_TLSS_EXT.1.1 Test 3.

Modified by TD0294 and TQ1660

FCS_TLSS_EXT.2.1

Test 4: The evaluator shall perform the following modifications to the traffic:

~~a) Modify at a byte in the client's nonce in the Client Hello handshake message, and verify that the server rejects the client's Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message.~~

~~b) [conditional] If an ECDHE or DHE ciphersuite is selected, modify the signature block in the Client's Key Exchange handshake message, and verify that the server rejects the client's~~

~~Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message.~~

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

d) After generating a fatal alert by sending a Finished message from the client before the client sends a ChangeCipherSpec message, send a Client Hello with the session identifier from the previous test, and verify that the server denies the connection.

e) Send a garbled message from the client after the client has issued the ChangeCipherSpec message and verify that the Server denies the connection.

This requirement was tested in conjunction with FCS_TLSS_EXT.1.1 Test 4.

c) A TLS test client was used to open a connection to the TOE. A byte was modified in the Client Finished message which resulted in the TOE terminating the connection.

d) The evaluator verified that the TOE rejects an attempt to resume a previous session that was terminated with a fatal alert.

e) A TLS test client was used to send a garbled message to the TOE after the ChangeCipherSpec message. The TOE was observed to reject the connection attempt.

Modified by TD0294

FCS_TLSS_EXT.2.2

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.

This requirement was tested in conjunction with FCS_TLSS_EXT.1.2 Test 1. In that test, the evaluator attempted to connect to the TOE will using a SSL/TLS protocol version that has been deprecated. The TOE rejected all connection attempts.

Modified by TD0294

FCS_TLSS_EXT.2.3

If the second selection includes any choice other than "no other", the evaluator shall attempt a connection using an ECDHE ciphersuite and a configured curve and, using a packet analyzer, verify that the key agreement parameters in the Key Exchange message are the ones configured. (Determining that the size matches the expected size for the configured curve is sufficient.) The evaluator shall repeat this test for each supported NIST Elliptic Curve and each supported Diffie-Hellman key size.

This test was performed in conjunction with FCS_TLSS_EXT.1.3 Test 1. In that test, the evaluator attempted to connect to the TOE while only using the supported curve size. The TOE completed the connection for the individual curve group.

Modified by TD0294

FCS_TLSS_EXT.2.4

Test 1: The evaluator shall configure the server to send a certificate request to the client and shall attempt a connection without sending a certificate from the client. The evaluator shall verify that the connection is denied.

The evaluator attempted to open a mutually authenticated TLS connection to the TOE using a client that was not configured to send a certificate. The TOE was shown to reject the connection attempt.

Modified by TD0348

FCS_TLSS_EXT.2.4

Test 2 [conditional]: If the TOE supports TLS 1.2 and higher, the evaluator shall configure the server to send a certificate request to the client without populating the supported_signature_algorithm field with the 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.

The evaluator configured the server to send a certificate request to the client without populating the supported_signature_algorithm field with the signature algorithm used by the client's certificate. PCAP captures verified the server sent a request to the client with the supported_signature_algorithm field identifying a MD5 hash. A PCAP capture also verified that the client supported_signature_algorithm field with the MD5 signature algorithm was missing. Identifying a mismatch between the client and server. The evaluator verified that the connection was denied.

Modified by TD0294

FCS_TLSS_EXT.2.4

Test 3: The evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the administrative guidance, the evaluator shall then load a certificate, or certificates, needed to validate the certificate to be used in the function and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates and show that the function fails.

The evaluator attempted to open a mutually authenticated TLS connection to the TOE while using a client certificate issued by an intermediate CA not in the TOE's trust store. The TOE rejected the connection attempt. The TOE was then presented with the intermediate CA certificate that was tied to a trusted root installed on the TOE, which lead to a successful connection.

Modified by TD0294

FCS_TLSS_EXT.2.4

Test 4: If the TOE supports sending a non-empty Certificate Authorities list in its Certificate Request message, the evaluator shall configure the client to send a certificate that does not chain to one of the Certificate Authorities (either a Root or Intermediate CA) in the server's Certificate Request message. The evaluator shall verify that the attempted connection is denied. If the TOE doesn't support sending a non-empty Certificate Authorities list in its Certificate Request message, this test shall be omitted.

The evaluator attempted to open a mutually authenticated TLS connection to the TOE using a client certificate that was issued by a CA that was in the TOE's Certificate Request message. The TOE rejected this connection attempt.

Modified by TD0294

FCS_TLSS_EXT.2.4

Test 5: 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.

The evaluator attempted to open a mutually authenticated TLS connection to the TOE using a client certificate without the Client Authentication purpose. The TOE rejected this connection attempt. Another connection attempt was made using a certificate that was identical except that the Client Authentication purpose was present. The TOE accepted this connection attempt.

Modified by TD0294

FCS_TLSS_EXT.2.4

Test 6: The evaluator shall perform the following modifications to the traffic:

a) Configure the server to require mutual authentication and then modify a byte in the client's certificate. The evaluator shall verify that the server rejects 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. The evaluator shall verify that the server rejects the connection.

b) The TOE rejected a mutually authenticated TLS connection attempt when the client's certificate had a modified byte.

c) The TOE rejected a mutually authenticated TLS connection attempt when the client's certificate had a modified signature block.

Modified by TD0294

FCS_TLSS_EXT.2.5

Testing for FCS_TLSS_EXT.2.5 is included in tests for FCS_TLSS_EXT.2.4

Modified by TD0294

FCS_TLSS_EXT.2.6

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 attempted to open a mutually authenticated TLS connection to the TOE using a client certificate with an identifier that didn't match what the TOE expected. The TOE rejected this connection attempt.

2.4 User Data Protection (FDP)

2.4.1 Certificate Profiles (FDP_CER_EXT.1)

2.4.1.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the certificate profile function in accordance with FDP_CER_EXT.1.1 The TSS shall describe how certificate profiles are configured and then selected to issue certificates in accordance with FDP_CER_EXT.1.2. The evaluator shall also ensure that the TSS describes how the TSF ensures that a certificate-requesting subject possesses the applicable private key. Finally, the evaluator shall ensure that the TSS describes how 20 bits of random are generated in accordance with FDP_CER_EXT.1.3 and which certificate fields are involved.

[ST] Section ***9.4.1 FDP_CER_EXT.1 Certificate Profiles*** describes the certificate profile function. The description covers all properties and fields required by FDP_CER_EXT.1. All certificates issued are consistent with the configured profile. Administrators configured the certificate profiles via the admin site. Subscribers select a certificate profile when submitting their certificate request via the public web site, EST or RAMI.

Except for certificate request submitted through Registration Authority Management Interface (RAMI), the TOE requires a valid digital signature covering the request by a private key matching the public key in the request. The digital signature on the certificate requests is checked when the requests are submitted and if not valid, the TOE will reject the requests. The TOE accepts unsigned certificate requests through RAMI only.

[ST] Section ***9.2 Communication (FCO)*** states that any certificate request submitted to the TOE must have a valid proof of origin regardless of how it is submitted (upload, EST, etc.). Except for requests submitted through RAMI, the TOE requires a valid digital signature covering the request by a private key matching the public key in the request. The digital signature on these requests is checked when they are submitted, and, if not valid, the request is rejected. In the case of an RA using RAMI, the RA can be responsible for proving the origin of requests it submits and such proof implied by the RA's submission of the request. Thus the TOE supports unsigned certificate requests through RAMI only.

The TSF uses the database sequence to keep track of the next sequential number. Each 20-byte serial number consists of 3 leading random bytes and 17 bytes representing the next sequential number, padded with leading zeros. The random bytes are obtained from the ISC CDK using getrand2() which meets the requirements of FCS_RGB_EXT.1

2.4.1.2 Guidance Activities

> The evaluator shall examine the operational guidance to ensure that instructions are available to configure certificate profiles used for certificate generation in accordance with this requirement. The operational guidance shall also specify how to configure proof of possession and, if applicable, how to configure unique serial number generation.

[CCECG] Section ***4.5.7.3 Managing Certificate Profiles*** describes management of certificate profiles and management of certificate issuance in section ***4.5.7.4 Managing Certificate Issuance***.

[CCECG] Section ***5.5.1 FDP_CER_EXT.1 Certificate Profiles*** states that any certificate request submitted to the TOE must have a valid proof of origin, unless it has been submitted through the RAMI interface. Otherwise, the request will be rejected. There is no option in the TOE to configure the proof of possession setting.

Unique serial number generation is not configurable. TSF uses the database sequence to keep track of the next sequential number. Each 20-byte serial number consists of 3 leading random bytes and 17 bytes representing the next sequential number, padded with leading zeros.

2.4.1.3 Test Activities

The evaluator shall perform the following tests for each supported certificate format:

> Test 1: The evaluator shall configure a certificate profile using the available guidance, request a certificate using the profile, and then examine the certificate contents to ensure it matches the configured certificate profile.

The evaluator configured a certificate profile on the TOE and then used it to issue a certificate. The certificate's attributes matched the profile.

Test 2: The evaluator shall specifically examine the certificate generated in Test 1 to ensure that it satisfies all field constraints in FDP_CER_EXT.1.2.

This test was performed as a part of FDP_CER_EXT.1 Test 1. The examination of the certificates generated in that test showed that they satisfied the constraints in FDP_CER_EXT.1.2.

Test 3: The evaluator shall test the fields "d", "e", "f", and "i" in FDP_CER_EXT.1.2 as follows:

Field "d": The evaluator shall send a request with a subjectPublicKeyInfo that is allowed by the profile, and observe the request succeeds. The evaluator shall then send a request with a subjectPublicKeyInfo that is not allowed by the profile, and observe that the request is rejected (or the value that is put into the certificate is what was in the profile).

Field "e": The evaluator shall send a request with a KeyUsage that is allowed by the profile, and observe the request succeeds. The evaluator shall then send a request with a KeyUsage that is not allowed by the profile, and observe that the request is rejected (or the value that is put into the certificate is what was in the profile).

Field "f": The evaluator shall send requests to show that the CA accepts requests that provide an identifier in either one or both of the subject and subjectAltName fields, but rejects requests that do not provide an identifier for either one of those fields.

Field "i": For each EKU listed in section 4.2.1.12 of RFC 5280, the evaluator performs the following tests. The evaluator shall send a request with a KeyUsage that is consistent (as documented in section 4.2.1.12 of RFC 5280) with the profile EKU, and observe the request succeeds. The evaluator shall then send a request with a KeyUsage that is not consistent (as documented in section 4.2.1.12 of RFC 5280) with the profile EKU, and observe that the request is rejected. The evaluator shall send the EKU to a profile with a consistent KeyUsage (but no specified EKU) and observe the request succeeds. The evaluator shall send the EKU to a profile with an inconsistent KeyUsage (but no specified EKU) and observe the request is rejected.

Field d: The evaluator verified that the TOE issued a certificate for a request whose subjectPublicKeyInfo matched what was in the certificate profile and rejected a request with a disallowed subjectPublicKeyInfo.

Field e: The evaluator verified that when a certificate request with a KeyUsage that wasn't allowed by a profile was submitted the TOE issued a certificate whose KeyUsage had been altered to what was allowed by the profile.

Field f: The evaluator verified that the TOE rejects certificate requests without an identifier in either of the SAN fields.

Field i: The evaluator verified that the TOE rejects certificate requests with inconsistent Key Usage and EKU values.

Modified by TD0353

Test 4: For each extendedKeyUsage value defined in section 4.2.1.12 of RFC 5280, the evaluator shall attempt to configure a certificate profile with each inconsistent keyUsage for that extendedKeyUsage field. If the CA rejects the attempt to create such a profile, then the test succeeds. If the creation of such a profile is allowed within the constraints of the AGD, the evaluator shall submit a certificate request using the profile, and show that the TSF does not issue the certificate.

The evaluator attempted to configure the serverAuth profile with each incompatible KeyUsage value. The TOE rejected these attempts with an invalid settings message. The test was then repeated for each set of EKU(Client Auth, Code signing, OCSP signing, Email protection, and time stamping) and confirmed that the TOE denied the setting from being applied.

Test 5: The evaluator shall configure a certificate profile and create a certificate request that violates the validity period setting in the configured profile (e.g., notBefore precedes the current time, the combination of notBefore and notAfter is beyond the validity period setting). The evaluator shall submit the certificate request using the profile and verify that the TSF rejects the request.

The evaluator configured a certificate request that violated a validity period that was set to be notBefore precedes the current time. The evaluator then verified that the TOE rejected this request. The evaluator then created a certificate request that violated the validity period by being beyond the default validity period. The evaluator then verified the TOE rejected this request.

2.4.2 Certificate Request Matching (FDP_CER_EXT.2)

2.4.2.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the linkage between submitted requests and issued certificates.

[ST] section ***9.4.2 FDP_CER_EXT.2 Certificate Request Matching*** states that each certificate request is identified by a unique request ID which is linked to the issued certificate. Each certificate is identified by a unique issuer DN and serial number.

2.4.2.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it contains instructions for how to trace a submitted request to an issued certificate and vice versa via the TOE's interface.

[CCECG] Section ***5.5.2.1 Tracing a Submitted Request to an Issued Certificate*** provides instructions for how to trace a request to an issued certificate and to trace an issued certificate to a request.

2.4.2.3 Test Activities

The evaluator shall perform the following test:

Test 1: The evaluator shall configure a certificate profile using the available guidance and request a certificate using the profile as a subscriber. The evaluator shall then assume the CA Operations role and verify that a linkage between submitted certificate requests and issued certificates is provided.

The evaluator submitted a certificate signing request to the TOE and issued a certificate based on it. The evaluator verified that the CSR's and assigned profile could be seen.

2.4.3 Certificate Issuance Approval (FDP_CER_EXT.3)

2.4.3.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the certificate issuance approval function, including the available interfaces that must be used.

[ST] Section ***9.4.3 FDP_CER_EXT.3 Certificate Issuance Approval*** indicates that TOE supports the approval of certificates issued according to a configured certificate profile through the web interface, RAMI, or EST. Only privileged users with 'CA Operations Staff' role and 'certify' permissions can approve certificates via the CA Account site (manual issuance approval is only performed for certificates request submitted via the public site). Certificate requests submitted via RAMI or EST and that specified the configured profile to use are automatically approved if the digital certificate covering the request is valid.

If the Registration Authority Management Interface (RAMI) is enabled, a privileged user with 'CA Operations Staff' role and 'RAMI' permission can submit a request to RAMI, specifying the profile to use, and the request will be approved automatically as long as they have 'RAMI' permission for the requested profile.

If EST is enabled, an authenticated subscriber can submit a request to the interface, on a per-profile basis, which will be approved automatically if it meets the requirements listed in Section 9.5.5.

2.4.3.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure that it contains instructions for any configuration aspects of the certificate issuance approval function and the steps needed to perform an approval.

[CCECG] Section ***5.1 Prerequisite*** contains the instructions to configure the RAMI and EST settings for certificate issuance approval. The steps for manual certificate issuance approval (requests submitted via the public site), are described in [CCECG] Section ***4.5.4.3 Issuing Certificates***.

2.4.3.3 Test Activities

The evaluator shall perform the following test:

Test 1: The evaluator shall configure the certificate issuance approval function in accordance with the operational guidance. The evaluator shall create a certificate request and submit it to the TOE. The evaluator shall access the TOE using the defined interface and verify that the submitted request is in the appropriate queue. The evaluator shall then assume either the CA Operations Staff role or the RA Staff role and approve the certificate request and issue the certificate. The evaluator shall verify that a certificate was issued.

This test was performed in conjunction with **FDP_CER_EXT.2 Test 1**. That test demonstrated the submitting of a Certificate Signing Request which is then queued and approved by the CA Operational staff. The new signed cert was then exported and examined.

If ‘rules’ is selected in FDP_CER_EXT.3.1 to allow automatic approval, the evaluator shall follow operational guidance to configure the certificate issuance approval function to follow a rule for automatic approval, and perform the following tests:

Test 2: The evaluator shall construct one or more certificate requests that meet the rules for automatic approval, and shall verify that each requested certificate was issued.

This test was performed in conjunction with the **FIA_ESTS_EXT.1 Test 1**. EST is configured to automatically approve certificate requests that meet specified criteria. **FIA_ESTS_EXT.1 Test 3** in particular shows the automatic issuance of a certificate when using certificate base authentication.

Test 3: The evaluator shall attempt to construct one or more certificate requests that violate the rules for automatic approval, and shall verify that the requested certificates are not issued.

This test was performed in conjunction with the **FIA_ESTS_EXT.1** tests. EST is configured to automatically approve certificate requests that meet specified criteria. **FIA_ESTS_EXT.1 Test 4** in particular shows the denial of a certificate that does not meet automatic approval criteria. In that test, the RA EKU was not set which resulted in the TOE rejecting the request.

2.4.4 Certificate Revocation List Validation (FDP_CRL_EXT.1)

2.4.4.1 TSS Activities

The evaluator shall examine the TSS to ensure it indicates whether the TOE supports CRL generation and, if so, describes the CRL generation function. Also, the evaluator shall ensure that the TSS identifies which of the values identified in FDP_CRL_EXT.1.1 can be included in CRLs.

[ST] Section ***9.4.7 FDP_CRL_EXT.1 Certificate Revocation List Validation*** indicates that the TOE supports CRL generation on demand, on schedule, or when certain revocation reasons are used depending on its configuration. CRLs issued by the TOE contain all fields required by the SFR and the TOE validates all field values required to be validated by the SFR.

2.4.4.2 Guidance Activities

If the TOE supports configuration of the CRL issuing function, the evaluator shall examine the operational guidance to ensure that instructions are available to configure issuance of CRL in accordance with FDP_CRL_EXT.1.1.

[CCECG] Section ***4.5.7.5 Managing CRL Issuance*** in the CC-Guide provides the instructions for configuring issuance of CRL.

2.4.4.3 Test Activities

The evaluator shall perform the following tests:

Test 1: If CRL can be issued, the evaluator shall configure the CRL function using available user guidance and request a CRL in order to ensure that the resulting CRL satisfies all field constraints in FDP_CRL_EXT.1.1.

The evaluator verified that the TOE could issue a valid CRL that satisfied all field constraints.

Test 2: For each field defined in FDP_CRL_EXT.1.1, the evaluator shall attempt to create a CRL that violates the required conditions of the field. The evaluator shall determine that all such attempts are rejected by the TSF.

The process of issuing and downloading a CRL is described in section 4.5.6 of the CCECG. Additionally, Test 1 for this SFR shows the process of issuing and downloading a CRL. As those show it is not possible to create a CRL that violates the conditions of FDP_CRL_EXT.1.1. There are no fields that are configurable by an administrator or user for such a purpose.

Test 3: The evaluator shall make a selection of fields from a configured CRL function and shall attempt to create a CRL that violates the required conditions of the field. The evaluator shall determine that all such attempts are rejected by the TSF.

The process of issuing and downloading a CRL is described in section 4.5.6 of the CCECG. Additionally, Test 1 for this SFR shows the process of issuing and downloading a CRL. As those show it is not possible to create a CRL that violates the conditions of FDP_CRL_EXT.1.1. There are no fields that are configurable by an administrator or user for such a purpose.

2.4.5 Certificate Status Information (FDP_CSI_EXT.1)

2.4.5.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the certificate status function and applicable formats, in accordance with this requirement, that can be used to issue certificate status. The TSS must reflect the selection made by the ST author as well as the selection-based requirements from Annex B.

[ST] Section ***9.4.4 FDP_CSI_EXT.1 Certificate Status Information*** describes the certificate status function; [ST] Section ***9.4.7 FDP_CRL_EXT.1 Certificate Revocation List Validation*** provides additional details about CRL issuance. The TOE issues CRLs in accordance with RFC 5280 and ITU-T Recommendation X.509 on demand or based on rules. The CRL format (validity, message digest, and whether or not reason codes are included) is configured by a CA Administrator. Privileged users with 'CA Operations Staff' role and 'revoke' or 'RAMI' permission, can issue a CRL on demand through the web interface or the RAMI interface respectively. Privileged users with 'CA Operations Staff' role can configure the rules for automatic CRL issuance using the web interface.

For TOEs that support OCSP, the TOE's ST shall specify the OCSP standard and the ST author shall ensure that a description of the format is available.

[ST] Section ***9.4.4 FDP_CSI_EXT.1 Certificate Status Information*** states that the TOE accepts OCSP requests that meet sections 2.1 and 4.1.1 of IETF RFC 6960. The TOE processes requests per Section 4.1.2 of the specification, which states that support for any specific extension is OPTIONAL. The only extension that the TOE supports is the Nonce extension as defined in section 4.4.1 of the RFC. Other extensions are ignored, unless they are critical in which case the TOE will reject the request. Section 4.1.2 of the RFC states that requests MAY be signed, but does not indicate if or how a responder should handle such requests. For these requests, the TOE simply ignores the signature and treats the request as unsigned.

OCSP responses created by the TOE comply with section 2.2, 2.3, 2.4, 4.2.1, and 4.2.2.3 of IETF RFC 6960. The TOE does not support sections 2.5, 2.6, or 2.7 of that specification. The only supported response type is id-pkix-ocsp-basic. Responses are signed by the CA who issued the certificate in question. The TOE returns "unknown" when it doesn't know about the certificate being requested and does not return "revoked" in this instance. Responses generated in response to requests containing the Nonce extension will include a Nonce extension. As allowed by section 4.2.2.3 of the RFC, responses do include certificates in the certs field to help the client verify the responder's signature.

> The evaluator shall also ensure that the TSS describes the process for approving changes to the status of a certificate, including the interfaces that must be used.

[ST] Section ***9.4.4 FDP_CSI_EXT.1 Certificate Status Information*** states that privileged users with 'CA Operations Staff' role and 'revoke' or 'RAMI' permission, or subscribers, can approve changes to the status of a certificate.

2.4.5.2 Guidance Activities

> If the TOE supports the configuration of certificate status information, the evaluator shall examine the operational guidance to ensure that instructions are available to configure the certificate status function to utilize the formats identified in FDP_CSI_EXT.1.1.
>
> The evaluator shall examine the operational guidance to ensure that it contains instructions for any configuration aspects of the certificate status change approval function and the steps needed to perform an approval.

[CCECG] Section ***4.5.7.8 Managing Revocation Policy*** provides the instructions for configuring the revocation policy. Section ***4.5.5.3 Revoking Certificates*** contains the instructions for revoking certificates via the CA Accounts site, Section ***4.7.3 Revoking a Certificate*** contains instructions for revoking a certificate via RAMI and Section ***4.6.5 Using Self-service Revocation*** contains the instructions for revoking a certificate using the subscriber self-service link on the Public site. The CC-Guide notes that on the Public site subscribers can only revoke, and are only presented with, certificates containing the same issuer DN and subject DN as the certificate which they used to authenticate to the revocation page. Once the revocation request has been submitted, it will be approved automatically.

2.4.5.3 Test Activities

Based on the selection, the evaluator shall perform the applicable tests associated with the requirements in Annex C:

> Test 1: For certificate status information, the evaluator shall configure the TSF to provide certificate status information according to each format identified in FDP_CSI_EXT.1.1 in turn and request certificate status for each format. Each certificate status response shall be examined to ensure that it conforms to the format as described in the TSS.

This test was performed in conjunction with FDP_CSI_EXT.1 Test 2. That test used OpenSSL to check a CRL and an OCSP response from the TOE. Both were shown to conform to the required formats.

> Test 2: For each selected certificate status format, the evaluator shall issue a valid certificate from the TOE. The evaluator shall then cause the TOE to issue certificate status information. The evaluator shall check the certificate status information to verify that it reflects that the certificate is valid.

For this test the evaluator checked the status of the certificate generated for FDP_CER_EXT.2. First the evaluator downloaded a CRL from the TOE and examined it in OpenSSL. Next the evaluator queried the TOE's OCSP responder with OpenSSL. Both of these showed that the certificate was valid.

Test 3: For each selected certificate status format, the evaluator shall revoke a valid certificate from the TOE. The evaluator shall then cause the TOE to issue certificate status information. The evaluator shall check the certificate status information to verify that it reflects that the certificate is revoked.

For this test the evaluator revoked and then checked the status of the certificate that was previously generated from the TOE. The evaluator then downloaded a CRL from the TOE and queried the OCSP responder for the certificate in question. Both of these showed that the certificate was revoked.

Test 4: The evaluator shall configure the certificate status change approval function in accordance with the operational guidance. The evaluator shall create a certificate status change request and submit it to the TOE. The evaluator shall access the TOE using the defined interface and verify that the submitted request is in the appropriate queue. The evaluator shall approve the certificate status change request. The evaluator shall then cause the TOE to issue certificate status information. The evaluator shall check the certificate status information to verify that it reflects the state of the certificate.

The evaluator used the TOE's self-service interface to request the revocation of a certificate. A CRL was downloaded and verified that the certificate was in fact revoked.

2.4.6 OCSP Basic Response Generation (FDP_OCSPG_EXT.1)

2.4.6.1 TSS Activities

The evaluator shall examine the TSS to ensure it indicates whether the TOE supports OCSP and, if so, describes the OCSP response function. Also, the evaluator shall ensure that the TSS identifies which of the values identified in FDP_OCSPG_EXT.1.1 can be included in OCSP responses.

[ST] Section ***9.4.8 FDP_OCSPG_EXT.1 OCSP Basic Response Generation*** describes how the TOE implements this SFR. The OCSP responses are signed by either the CA's issuer private key which resides in the PKCS#11 Cryptographic Module or a designated OCSP signing credential whose certificate includes the OCSP Signing extended key usage OID and whose private key resides in the PKCS#11 Cryptographic Module. SHA-1, SHA-256, SHA-384 and SHA-512 are supported and can be configured via either the TOE's CA Account web interface (for issuers the CA is hosting) or by the TOE's Admin Account web interface (for external issuers) by an administrator. OCSP responses are based on CRLs stored by the CA which can be those generated by issuers hosted by the CA itself or CRLs obtained from external issuers manually or via HTTP.

The following values are included in the OCSP responses:

a) The version field always contain a 0.

b) The signatureAlgorithm field contains the object identifier (OID) for a digital signature algorithm in accordance with FCS_COP.1(2). The supported OIDs are: ecdsa-with-SHA1 (1.2.840.10045.4.1), ecdsa-with-SHA256 (1.2.840.10045.4.3.2), ecdsa-with-SHA384 (1.2.840.10045.4.3.3), ecdsa-with-SHA512 (1.2.840.10045.4.3.4), sha1WithRSAEncryption (1.2.840.113549.1.1.5), sha256WithRSAEncryption (1.2.840.113549.1.1.11), sha384WithRSAEncryption (1.2.840.113549.1.1.12), sha512WithRSAEncryption (1.2.840.113549.1.1.13).

c) The thisUpdate field indicates the time at which the status being indicated is known to be correct.

d) The producedAt field indicates the time at which the OCSP responder signed the response.

e) The time specified in the nextUpdate field does not precede the time specified in the thisUpdate field.

2.4.6.2 Guidance Activities

If the TOE supports configuration of the OCSP function, the evaluator shall examine the operational guidance to ensure that instructions are available to configure the OCSP response function in accordance with FDP_OCSPG_EXT.1.1.

[CCECG] Section ***4.5.7.6 Managing OCSP Responder Settings*** contain the instructions for configuring the OCSP response function.

2.4.6.3 Test Activities

The evaluator shall perform the following tests:

Test 1: For each OCSP response format identified in FCO_NRO_EXT.2.2, the evaluator shall configure the OCSP response function, establish a client and submit, in turn, an OCSP request to the TSF for the status of a certificate issued by a CA implemented by the TOE and which is not revoked, a certificate issued by a CA implemented by the TOE which has been revoked, and a certificate not issued by a CA implemented by the TOE. The evaluator shall ensure that the responses satisfy all constraints in FDP_OCSPG_EXT.1.1 and reflects the correct status in accordance with the referenced standard.

The evaluator created and signed a pair of certificate signing requests for this this test. The first one was kept valid while the second one was revoked after being issued. The evaluator queried the TOE's OCSP responder for status information on both certificates, as well as for a certificate that was not issued by the TOE. Correct status information was returned for both of the first two certificates, while the third was identified as not having been issued by the TOE.

Test 2: For each OCSP response format defined in FDP_CSI_EXT.1.1, and for each item a-e of this SFR, the evaluator shall attempt to create an OCSP response that violates the required conditions. The evaluator shall determine that all such attempts are rejected by the TSF.

The only OCSP response format defined in FDP_CSI_EXT.1.1 is the OCSP standard as defined by RFC 6960.

As described in section 5.5.7 of the AGD [CCECG] it is not possible to create an OCSP response that violates any of the conditions defined in this SFR.

2.4.7 Subset Residual Information Protection (FDP_RIP.1)

2.4.7.1 TSS Activities

The evaluator shall examine the TSS to ensure that, at a minimum, it describes how the previous information content is made unavailable, and at what point in the buffer processing this occurs.

[ST] Section ***9.4.5 FDP_RIP.1 Subset Residual Information Protection*** states that the TOE handles EST passwords and the TLS session object. EST passwords are converted into their check value and the memory related to them is cleared and freed with garbage collection.

TLS session objects (session ID, roles, held by the session) are Java string objects. When a session object is created, possibly reusing memory, new Java strings are created and initialized to the empty string destroying any residual information they may contain. The buffer processing consists of the following:

1. A new connection is established requiring the allocation of a new TLS session object.
2. When allocated, the environmental Java Runtime Environment requests memory from the environmental Operating System.
3. The environmental Operating System clears each memory page before allocating it to the process.
4. The environmental Java Runtime Environment clears each object when it is created.
5. The TLS session object is made available to the TOE for use.

Memory holding TLS session information is cleared prior to allocation by the Operational Environment.

2.4.7.2 Guidance Activities

There are no AGD assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.4.7.3 Test Activities

There are no ATE assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

2.4.8 Public Key Protection (FDP_STG_EXT.1)

2.4.8.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the trusted public keys and certificates implemented, including trust stores that contains root CA certificates, used to meet the requirements of this PP. This description shall contain information pertaining to how certificates are loaded into the store, and (if the first selection in the requirement is chosen) how the store is protected from unauthorized access in accordance with the permissions established in FMT_SMF.1 and FMT_MOF.1(1) through FMT_MOF.1(5).

[ST] Section ***9.4.6 FDP_STG_EXT.1 Public Key Protection*** describes certificate data storage. It describes the trust store that contains the root CA certificate. The TOE stores its certificate data including the trust anchor list and the various ACLs in database tables on the OE. Certificates may be added to the trust anchor list, by an authorized individual, only if the certificate is valid, self-signed, and asserts the cA flag in a critical basicConstraints extension. Certificates may be added to the various ACLs, by an authorized individual, only if the certificate does not assert the cA flag in a critical basicConstraints extension, asserts the Client Authentication value in an extendedKeyUsage extension, and is not expired.

The integrity of the trust anchor table and the tables storing the ACLs, is maintained using a digital signature created using the CA system credentials. This signature is validated when the table is used. The signature is updated when administrator modifies the trust list.

The TOE's web interface and trusted path may be used to modify these tables through the Admin Site and the CA Account Site. Once the trusted path is established, the TOE checks the ACL for the site being accessed to determine if the certificate is present and which permissions have been assigned to the holder of the certificate. If they are authorized to access the site in some way, the TOE displays a web page providing them links for the actions to which they are entitled. If they are not authorized to access the site at all, they are shown a permission denied message. If they are not authorized to access a particular function, the TOE does not display a link for that action (the TOE also verifies the permissions after the link is clicked to prevent someone from directly accessing a page for which they lack access).

To add or remove a certificate using the TOE's command line interface, an administrator logs into the environmental Operating System at the local console with a username and password. Once the administrator is logged in, they use the TOE's command line interface program to add certificates to the list by specifying appropriate parameters on the command line which must include the filename of a file containing the X.509 v3 certificate to be added.

An administrator can add and/or remove a certificate from the certificate database, using links from the web interface. Only administrator can access the option to manage or modify the trust anchor lists or the ACL database tables via the admin site; and only a TOE administrator with admin permissions on the OS can access trust anchor and ACL database tables via the local console.

The TOE component, Apache Tomcat, has its own trust anchor key store. Access to the Apache Tomcat trust anchor key store is controlled by the environmental Operation System. An administrator logs into the environmental Operating System at the local console with a username and password. Once logged in, they use the Java keytool application to add or remove certificates by specifying appropriate parameters on the command line which must include the filename of the keystore being modified, a key alias, and, for additions, the filename of a file containing the X.509 v3 certificate to be added.

2.4.8.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it contains instructions for how to load certificates and public key into and remove certificates and public keys from the protected storage.

[CCECG] Section ***4.4.2.2.1 Certificate and Path Validations*** describes certificate and path validations which includes the instructions for adding and removing certificates from the certificate store via the admin site. Section ***5.5.5 FDP_STG_EXT.1 Public Key Protection*** provides the instruction to manage the trust anchor list from the local console using the Admin Site and command line program (CACLI). The Installation guide in Section ***4.4.2.2.1 Certificate and Path Validations*** describes the commands to add or remove a certificate to the Tomcat keystore.

2.4.8.3 Test Activities

This test is conditional on the first selection in the SFR being chosen. If the second selection is chosen, the evaluator does not perform this and instead performs the actions called for FCS_CKM_EXT.5.

The evaluator shall perform the following test:

Test 1 (conditional): The evaluator shall attempt to modify the contents of the Trust Anchor Database in a way that violates the documented permissions and verify that the attempt fails.

The ST chose the second selection for this SFR. Accordingly, the test is not applicable.

2.5 Identification and Authentication (FIA)

2.5.1 Certificate Enrollment (FIA_ENR_EXT.1)

2.5.1.1 TSS Activities

The evaluator shall examine the TSS to ensure that it describes the certificate enrollment function options

[ST] Section ***9.5.6 FIA_X509_EXT.3 Certificate Request, FIA_ENR_EXT.1 Certificate Enrollment*** describes the certificate enrollment function. The TOE supports the generation of a PKCS#10 certificate request when establishing an issuer or when cross-certification with another issuer is desired. When establishing an issuer, a user in the Administrator role selects PKCS#10 certificate request as the desired type, obtains the request, submits it to the other issuer, and then imports the response into the TOE. To cross-certify with another issuer, an administrator "exports" the current credential as a PKCS#10 certificate request for cross certification and provides it to the other issuer who issues the certificate. No further steps are required by the TOE.

2.5.1.2 Guidance Activities

The evaluator shall examine the operational guidance documentation and confirm that it contains instructions for obtaining a certificate for the embedded CA using the options claimed in FIA_ENR_EXT.1.1.

[CCECG] Section ***5.6.5 FIA_ENR_EXT.1.1 Certificate Enrollment*** states that an external certification authority can be used to issue the CA's certificate managed by the TOE. To generate a certificate request to an external certification authority, see the steps in section ***4.5.3.2.1 External CA***.

2.5.1.3 Test Activities

Testing is covered under the tests for the referenced SFR of the claimed options.

2.5.2 Enrollment over Secure Transport (EST) Server (FIA_ESTS_EXT.1)

2.5.2.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes the implementation of this protocol. If the description indicates the use or RA or AOR for initial issuance or authorization of certificates, the evaluator shall examine the TSS to ensure that these roles are supported.

[ST] Section ***9.5.5 FIA_ESTS_EXT.1 Enrollment over Secure Transport (EST) Server*** states that the TSF supports Enrollment over Secure Transport (EST) protocol as specified in RFC 7030 to receive and act upon certificate enrollment requests using the simple enrollment method described in RFC 7030 Section 4.2. Certificate enrollment requests are authenticated using an existing certificate and corresponding private key as specified by RFC 7030 Section 3.3.2, authenticated using a username and password as specified by RFC 7030 Section 3.2.3, or authenticated using a special RA certificate issued by the CA and asserting the id-kp-cmcRA OID in its extended key usage extension as specified by RFC 7030 Section 3.7.

In cases where the entity requiring a certificate does not have a valid certificate to use for authentication, EST basic authentication is used. In order for an entity to enroll via EST using basic authentication, a CA Operations Staff member of the CA account must add the common name of the subscriber to the EST list and create an EST password. They then have to pass the EST subscriber name and password information to the subscriber.

2.5.2.2 Guidance Activities

> The evaluator shall examine the operational guidance to ensure it contains instructions on configuring the TOE so that EST conforms to the description in the TSS.

[CCECG] Section ***5.1.6 Configuring EST*** describes the EST settings that conforms to the description in the TSS.

The TOE supports EST's simple enrollment, which is disabled by default.

To enable this service, follow the steps in section ***4.5.7.1.3 EST (Enrollment over Secure Transport)***. EST service allows subscribers with existing credentials and a special RA to enroll via the EST using client authentication interface.

For the subscribers without valid credentials for client authentication, a CA Operations Staff member must add the common name of the subscriber to the EST list and create an EST password. Follow the steps in section ***4.5.7.9.2 EST (Enrollment over Secure Transport)*** Users to add subscribers to the list.

2.5.2.3 Test Activities

The evaluator shall perform the following tests:

> Test 1: The evaluator shall use an EST client to request certificate enrollment of an authorized subject to obtain a new certificate from the TOE using the simple enrolment method described in RFC 7030 Section 4.2, authenticating the request using an existing certificate and corresponding private key as described by RFC 7030 Section 3.3.2. The evaluator shall confirm that the TOE issues a certificate and returns it to the client.

The evaluator verified that the TOE would issue a certificate in response to an EST simple enrollment request. An audit record was generated to verify the successful operation.

> Test 2: If username and password authentication is selected in FIA_ESTS_EXT.1.3, the evaluator shall use an EST client to request an initial certificate for a user from the TOE using the simple enrollment method described in RFC 7030 Section 4.2, authenticating the request using a username and password as described by RFC 7030 Section 3.2.3. The evaluator shall confirm that the TOE issues a certificate and returns it to the client.

This test was performed in conjunction with **FIA_ESTS_EXT.1.1 Test 1**. The evaluator verified that the TOE would issue a certificate in response to an EST request using a valid username/password authentication.

> Test 3: If "a certificate issued by the same issuer that asserts id-kp-cmcRA in its extended key usage extension" is selected in FIA_ESTS_EXT.1.4, the evaluator shall use an EST client to request certificate enrollment of a subject not known to the TOE to be authorized, to request an initial certificate from the TOE using the simple enrollment method described in RFC 7030 Section 4.2, authenticating the request

using an RA's certificate issued by the TOE's Certification Authority functionality that asserts id-kp-cmcRA in its extended key usage extension. The evaluator shall confirm that the TOE issues a certificate and returns it to the client.

The evaluator verified that the TOE would issue a certificate in response to an EST request using certificate authentication. In this case, the client certificate asserted id-kp-cmcRA in its extended key usage extension.

Test 4: If "a certificate issued by the same issuer that asserts id-kp-cmcRA in its extended key usage extension" is selected in FIA_ESTS_EXT.1.4, the evaluator shall use an EST client to request certificate enrollment of a subject not known to the TOE to be authorized, to request an initial certificate from the TOE using the simple enrollment method described in RFC 7030 Section 4.2, authenticating the request using a certificate issued by the TOE's Certification Authority functionality that does not assert id-kp-cmcRA in its extended key usage extension and which is not associated with RA or AOR privileges by the CA. The evaluator shall confirm that the TOE does not issue a certificate.

The evaluator verified that the TOE would not issue a certificate in response to an EST request using certificate authentication if the client certificate did not assert id-kp-cmcRA in its extended key usage extension and had a subject not know to the TOE.

Test 5: The evaluator shall modify the EST client or setup a man-in-the-middle tool between the EST client and TOE to perform the following modifications to a valid certificate request:

- Modify at least one byte in the certificationRequestInfo field of the certificate request message and verify that the TOE rejects the request.

The evaluator verified that the TOE rejected an EST request when the certificate request message had its certificationRequestInfo field modified.

2.5.3 Authentication Mechanism (FIA_UAU_EXT.1)

2.5.3.1 TSS, Guidance, Test Activities

Assurance 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.

2.5.4 User Identification and Authentication (FIA_UIA_EXT.1)

2.5.4.1 TSS Activities

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

[ST] Section ***9.5.4 FIA_UIA_EXT.1 User Identification and Authentication*** describes the logon process.

All privileged user actions require successfully authenticating to the TOE using TLS/HTTPS with client authentication using certificates, or use of the TOE's command line tools after logging in to the environmental operating system.

The TOE uses HTTPS/TLS with certificate-based client authentication as the only logon method to the Admin and CA Account Sites.

A “successful logon” to the TOE’s Admin Site, CA Account Site, RAMI, or DBAccess interface requires:

- Client credentials (certificate, private key, and certificate chain) must be installed in the browser’s key store or otherwise available to the process authenticating to the TOE.
- The client trust key store must contain a trust anchor for the server’s TLS/HTTPS certificate.
- The server’s trust key store must contain the trust anchor for the client certificate.
- The client certificate must appear on the proper ACL with an appropriate role and permission.

The client certificate must pass the TOE’s certificate path validation with CRL checking specified in FIA_X509_EXT.1.1.

A “successful logon” to use the TOE’s command line tools requires:

- Logging in to the environmental Operating System with administrator rights
- Executing one of the TOE’s command line tools

It is assumed that if the user can launch the TOE’s command line tool that they must be a TOE administrator.

Subscriber actions, other than those listed above, require successfully authenticating to the TOE using TLS/HTTPS with client authentication using certificates or via EST. Subscribers may submit certificate requests using either the TOE’s Public site or through EST. Using the TOE’s Public site, a subscriber may, without identification and authentication, submit a certificate request through one of two (2) web forms. A subscriber may upload an existing certificate request, or they may generate a new key pair and request in the browser itself. Certificate requests received in this manner are manually verified by having the subscriber confirm the request ID displayed post submission to a privileged user using an out-of-band communication method prior to issuance.

The evaluator shall examine the TSS to determine that it describes all actions that can be performed prior to I&A as well as all actions that require successful I&A, and by whom these actions can be performed. Any constraints on these services shall be documented in the TSS.

[ST] Section ***9.5.4 FIA_UIA_EXT.1 User Identification and Authentication*** describes the following actions prior to requiring a non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- Obtain certificate status information (retrieve CRL, submit OCSP request);
- Download certificate from repository;
- Respond to EST cacerts requests;
- Submit certificate requests;
- Obtain information about the TOE (version, current time, operating system type).

All other actions by privileged users or subscribers require successfully authenticating to the TOE.

2.5.4.2 Guidance Activities

The evaluator shall examine the operational guidance 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 login method, the evaluator shall ensure the operational guidance 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 operational guidance provides sufficient instruction on limiting all allowed services. The evaluator shall examine the operational guidance to verify that it describes how to configure the constraints on each type of subscriber self-service request.

[CCECG] Section ***5.1 Prerequisite*** and Section **5.6 Identification and Authentication (FIA)** Guide describe configuration of the TOE identification and authentication function, including login at the TOE Web interface, RAMI and DBAccess interfaces as well as EST client authentication. The description also covers configuring the subscriber self-service constraints for the public site, as well as the services that can be accessed before the user is authenticated.

2.5.4.3 Test Activities

The evaluator shall perform the following tests for each method by which privileged users access the TOE (local and remote), as well as for each type of credential supported by the access method in accordance with the authentication mechanisms listed in FIA_UAU_EXT.1:

Test 1: The evaluator shall use the operational guidance to configure the appropriate credential supported for the access method. For that credential/access 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.

Client certificates are the only type of credential used by the TOE for access control. The evaluator verified that a client certificate could be used to access the TOE. When a certificate was removed as an authorized credential it was no longer usable to access the TOE.

Test 2: The evaluator shall configure the non-authenticated services allowed according to the operational guidance, and then determine the services available to an external remote entity (including subscribers and relying parties). The evaluator shall determine that the list of services available is limited to those specified in the requirement. The evaluator shall also verify that non-authenticated remote entities cannot access the services listed in FIA_UIA_EXT.1.2 that require I&A.

All connections to the TOE over port 8443 are mutually authenticated which means that access is not possible without a client certificate.

Unauthenticated connections to the TOE on port 443 are directed to the TOE's public site.

Test 3: For local access, the evaluator shall exercise the services in accordance with FIA_UIA_EXT.1.1 available to a local privileged user prior to I&A, and make sure this list is consistent with the requirement.

N/A. There is no local access to the TOE without prior authentication, all access is over the remote HTTPS/TLS interface.

Test 4: The evaluator shall configure the constraints on subscriber self-service requests. The evaluator shall assume a CA Operations Staff or RA Staff role and issue a certificate to at least one unique subscriber. For each configured service, the evaluator shall request authorized activities using the issued certificates and verify that they can be performed.

The only type of self-service request that requires certificate authentication is submitting certificate revocation requests. The evaluator verified that this functionality could be accessed from the public site once the Revoked menu selection was initiated.

Test 5: The evaluator shall configure the constraints on subscriber self-service requests. The evaluator shall assume a CA Operations Staff or RA Staff role and issue a certificate to at least two unique subscribers. For each configured service, the evaluator shall request authorized activities using one issued certificate for the other subscriber's information and shall verify that the request is denied. The evaluator shall request unauthorized activities using one issued certificate and shall verify that the request is denied.

The only type of self-service request that requires certificate authentication is submitting certificate revocation requests. The evaluator verified that a certificate without correct permissions could not access this functionality.

2.5.5 Certificate Validation (FIA_X509_EXT.1)

2.5.5.1 TSS Activities

Modified per TD0522

The evaluator shall examine the TSS to ensure it describes where the check of validity of the certificates takes place. The evaluator shall ensure the TSS also provides a description of the certificate path validation algorithm for each certificate format supported by the TOE.

[ST] Section ***9.5.1 FIA_X509_EXT.1 Certificate Validation*** describes how the TOE performs certificate validation.

The TOE validates certificates as follows:

- IETF RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a certificate in the Trust Anchor Database managed by the TOE
- The TOE requires that all CA certificates in the path contain a basicConstraints extension asserting the cA flag.
- The TOE checks the revocation status using a Certificate Revocation List (CRL) as specified in FDP_CSI_EXT.1.
- The TOE validates that the certificate asserts the appropriate extended key usage values as follows:

- For certificates used for digitally signing trusted updates and executable code, the end entity certificate presented must have the Code Signing purpose (OID 1.3.6.1.5.5.7.3.3) set in the extendedKeyUsage field.
- For certificates used to authenticate to the TOE through its web interface the end entity certificate presented must have the Client Authentication purpose (OID 1.3.6.1.5.5.7.3.2) set in the extendedKeyUsage field.

Certificate validation occurs:

- When the TOE's command line update tool is executed to verify the update package's signature.
- When a TLS client connects to one of the TOE's web interfaces.
- When a certificate is added to the trust anchor list.

2.5.5.2 Guidance Activities

There are no AGD assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

[CCECG] Section ***5.6.1 FIA_X509_EXT.1 Certificate Validation, FIA_X509_EXT.2 Certificate-Based Authentication*** describes certificate validation.

2.5.5.3 Test Activities

Modified by TD0522.

The evaluator shall perform the following tests in conjunction with the other Certificate Services assurance activities, including the use cases in FIA_X509_EXT.2.1. The tests for the extendedKeyUsage rules are performed in conjunction with the uses that require those rules.

Modified by TD0522

Test 1: The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function failing, for each of the following reasons, in turn:

- by establishing a certificate path in which one of the issuing certificates is not a CA certificate,
- by omitting the basicConstraints field in one of the issuing certificates,
- by setting the basicConstraints field in an issuing certificate to have CA=False,
- by omitting the CA signing bit of the key usage field in an issuing certificate, and
- by setting the path length field of a valid CA field to a value strictly less than the certificate path.

The evaluator shall then establish a valid certificate path consisting of valid CA certificates, and demonstrate that the function succeeds. The evaluator shall then remove trust in one of the CA certificates, and show that the function fails.

This portion of the test was performed in conjunction with **FCS_TLSS_EXT.2.4 Test 3**. That test demonstrated that when the TOE received an X.509 certificate issued by a CA that the TOE didn't recognize, it automatically rejected the connection. It also demonstrated that when the issuing CA's certificate was present the TOE would accept the certificate.

The TOE also rejected certificates which lacked a valid certification path when the CA had a missing or FALSE BasicConstraints field. T

he evaluator also verified when a CA with no CA sign bit is provided to the TOE during a connection, the TOE rejects this connection.

For the last portion of the test, the evaluator set a path length of one even though there were more than two CA’s involved. The evaluator then verified that the TOE rejected this connection attempt.

Modified by TD0522

Test 2: The evaluator shall demonstrate that validating an expired certificate anywhere in a certificate path results in the function failing.

The evaluator created an end entity certificate with an expiration date of before the current day. The evaluator verified that the TOE rejects an expired certificate.

Modified by TD0522

Test 3: The evaluator shall test that the TOE can properly handle revoked certificates – conditional on whether CRL, OCSP, OCSP stapling, or OCSP multi-stapling is selected; if multiple methods are selected, and then a test is performed for each method. The evaluator has to only test one up in the trust chain (future revisions may require to ensure the validation is done up the entire chain). The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator shall then attempt the test with a certificate that will be revoked (for each method chosen in the selection) and verify that the validation function fails.

The evaluator verified that the TOE would accept a valid certificate that used CRL for revocation checking. When a similar certificate was revoked, the TOE rejected the client authentication attempt.

Modified by TD0522

Test 4: If any OCSP option is selected, the evaluator shall configure a delegated OCSP server to use a certificate that does not have the OCSP signing purpose to sign a valid response, 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.

The evaluator attempts to authenticate with an intermediate CA that lacked the CRL signing bit in the Key Usage field. This resulted in the TOE rejecting the communication attempt.

Modified by TD0522

Test 5: The evaluator shall modify a single byte in the certificate and verify that the certificate fails to validate.

The evaluator used a custom proprietary TLS test tool to modify a byte in the certificate used for a TLS connection. The evaluator then verified that the TOE would not accept a certificate that had had a byte modified.

Modified by TD0522

Test 6a: (Conditional on support for EC certificates as indicated in FCS_COP.1(3)). The evaluator shall establish a valid, trusted certificate chain consisting of an EC leaf certificate, an EC Intermediate CA certificate not designated as a trust anchor, and an EC certificate designated as a trusted anchor, where the elliptic curve parameters are specified as a named curve. The evaluator shall confirm that the TOE validates the certificate chain.

The evaluator created a trust anchor using an EC leaf certificate, an EC Intermediate CA not designated as a trust anchor, and an EC certificate designated as a trusted anchor on the TOE. The evaluator then connected to the TOE using an EC chain. The evaluator then verified that the TOE validates an EC certificate chain.

Modified by TD0522

Test 6b: (Conditional on support for EC certificates as indicated in FCS_COP.1(3)). The evaluator shall replace the intermediate certificate in the certificate chain for Test 6a with a modified certificate, where the modified intermediate CA has a public key information field where the EC parameters uses an explicit format version of the Elliptic Curve parameters in the public key information field of the intermediate CA certificate from Test 6a, and the modified Intermediate CA certificate is signed by the trusted EC root CA, but having no other changes. The evaluator shall confirm the TOE treats the certificate as invalid.

The evaluator provided an invalid EC intermediate cert with explicit parameters using the connection from the previous test. The evaluator then verified that the TOE rejects an EC certificate chain with a modified intermediate CA certificate and proved this with audit logs.

2.5.6 Certificate-Based Authentication (FIA_X509_EXT.2)

2.5.6.1 TSS Evaluation Activity

The evaluator shall examine the TSS to ensure it describes the certificate(s) used by the TOE, the different uses for each certificate, and how the TSF chooses which certificates to use. 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.

[ST] ***Section 9.5.2 FIA_X509_EXT.2 Certificate-Based Authentication*** describes the certificates used by the TOE, the different uses for each certificate, and how the TSF chooses which certificates to use.

The TOE relies on the following certificates:

- TLS/HTTPS server certificate – the certificate that authenticates the TOE to clients. Verification of this certificate is done by clients that are in the Operational Environment.
- Privileged user certificates – the certificates that are used by privileged users to authenticate to the TOE. These certificates are verified by the TOE as described in Section 9.5.1.
- RA or DBAccess certificates – the certificates that are used by client processes to authenticate to the TOE through the RAMI or DBAccess interfaces. These certificates are verified by the TOE as described in Section 9.5.1.
- Subscriber certificates – the certificates used by subscribers to authenticate to the TOE for self-service revocation or EST renewal. These certificates are verified by the TOE as described in Section 9.5.1.

- Code signing certificate – the certificate used by ISC to sign software updates. These certificates are verified by the TOE as described in Section 9.5.1.

Of the certificates listed above, the TOE directly uses only the TLS/HTTPS server certificate. The other certificates are used by others to prove their identity to the TOE, and which certificate to use is determined by those entities, not the TOE. The TLS/HTTPS server certificate is generated at installation time and its DN is stored in a configuration file. To change the TLS/HTTPS server certificate used by the TOE, the old credential must be removed from, and a new credential must be created on, the PKCS#11 Cryptographic Module. If the DN of the new certificate is different than the old certificate's value, the configuration file must also be updated.

2.5.6.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure clear instructions for configuring the operating environment so that the TOE can use the certificates which are provided. If the requirement is that the administrator is able to specify the default action if the peer certificate is deemed invalid, then the evaluator shall ensure that the operational guidance contains instructions on how this configuration action is performed.

[CCECG] Section ***5.6.1 FIA_X509_EXT.1 Certificate Validation, FIA_X509_EXT.2 Certificate-Based Authentication*** states that the TOE does not rely on the operating environment for certificate handling. When the TOE cannot determine the validity of a certificate, it will not accept the certificate and this action is not configurable.

2.5.6.3 Test Activities

The evaluator shall perform the following tests.

Test 1: For each function listed in FIA_X509_EXT.2.1 that requires the use of certificates the evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the operational guidance, the evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails.

This requirement was tested by FCS_TLSS_EXT.2.4 Test 3. That test requires the TOE to accept a client certificate with a valid certification path, but reject that same certificate when one or more of the CA certificates required to validate the client certificate are not present.

Test 2: 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 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 operational guidance to determine that all supported administrator-configurable options behave in their documented manner.

In this test, the evaluator removed the CRL information of the trusted entity certificate profile that had the permission to access the TOE's administrative webpage. However, since the CRL revocation data was unavailable for that certificate authority, the TOE rejected the communication.

2.5.7 X509 Certificate Request (FIA_X509_EXT.3)

2.5.7.1 TSS 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.

[ST] Section ***9.5.6 FIA_X509_EXT.3 Certificate Request, FIA_ENR_EXT.1 Certificate Enrollment*** describes the TOE implementation of this SFR. The ST author did not select 'device specific information".

2.5.7.2 Guidance Activities

> The evaluator shall check to ensure that the guidance documentation contains instructions on requesting certificates from a CA, including generation of a Certificate Request Message. 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 certificate request message.

[CCECG] Section ***4.5.3.2 Generating Credential for Subordinate CA*** provides the instructions for generating certificate request for a subordinate CA.

2.5.7.3 Test Activities

The evaluator shall perform the following tests:

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

The evaluator verified that the TOE generates a properly formatted CSR.

> Test 2: The evaluator shall demonstrate that validating a certificate response message 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 certificate response message, and demonstrate that the function succeeds.

The TOE requires signed CSR to be imported in a PKCS7 chain that includes the certificate of the issuing CA as well. It is not possible to import a signed CSR to the TOE without the required CA. The evaluator attempted to import a signed CSR without the signing CA and verified that this resulted in failure. The evaluator then created a signed CSR where the signing CA was present and attempted to import this onto the TOE. The evaluator verified that this attempt was successful. Audit records captured the events of the test.

2.6 Security Management (FMT)

2.6.1 Management of Security Functions Behavior (Administrator Functions) (FMT_MOF.1(1))

2.6.1.1 TSS, Guidance, and Test Activities

> Testing for this requirement is defined under FMT_MOF.1(4). The only difference between the iterations of FMT_MOF.1 is the specific set of management functions that are available to each administrative role. Testing for this SFR is conducted sufficiently thoroughly if the evaluator can demonstrate that the assigned role can perform only the functions specified in the SFR.

Assurance Activity for this SFR is defined under FMT_MOF.1(4).

2.6.2 Management of Security Functions Behavior (CA/RA Functions) (FMT_MOF.1(2))

2.6.2.1 TSS, Guidance, and Test Activities

> Testing for this requirement is defined under FMT_MOF.1(4). The only difference between the iterations of FMT_MOF.1 is the specific set of management functions that are available to each administrative role. Testing for this SFR is conducted sufficiently thoroughly if the evaluator can demonstrate that the assigned role can perform only the functions specified in the SFR.

Assurance Activity for this SFR is defined under FMT_MOF.1(4).

2.6.3 Management of Security Functions Behavior (CA Operations Functions) (FMT_MOF.1(3))

2.6.3.1 TSS, Guidance, and Test Activities

> Testing for this requirement is defined under FMT_MOF.1(4). The only difference between the iterations of FMT_MOF.1 is the specific set of management functions that are available to each administrative role. Testing for this SFR is conducted sufficiently thoroughly if the evaluator can demonstrate that the assigned role can perform only the functions specified in the SFR.

Assurance Activity for this SFR is defined under FMT_MOF.1(4).

2.6.4 Management of Security Functions Behavior (Admin/Officer Functions) (FMT_MOF.1(4))

2.6.4.1 TSS Activities

> The evaluator shall examine the TSS to ensure it identifies the restrictions consistent with this requirement. For every function specified across all iterations, the TSS must specify how the restriction is achieved and how (by role or some other specified mechanism). This applies whether the ST author selects “TSF” or “Operational Environment” in the first SFR selection.

[ST] Section ***9.6 Security Management (FMT)*** describes the TOE security management functions. The TOE provides two web interfaces for managing its functions and data: The administration webpages (admin site) and the CA Accounts webpages (CA site). Access to each site is protected with the corresponding ACLs (Access Control menu for the Administration site and CA Account menu for the CA Account webpage.

) and that control the user roles and permissions that can manage the provided admin functions and manipulate the TSF data.

The TOE depends on the underlying OS to provide a local console for managing the TOE locally and for managing the initial part of the TOE updates. In the evaluated configuration, the only users who are allowed to log in to the environmental Operating System are users in the administrator role. The environmental Operating System identifies its users by a username and authenticates them using a password and is capable of assigning roles and permissions to control its functions and protected data.

- Perform archival and recovery
- Perform destruction of sensitive data when no longer needed
- Perform private or secret key or critical data export

Privilege users with the admin and auditor roles can access the admin sites. Privilege users with admin, auditor, CA operations staff can access the CA sites.

The admin sites provide and restrict the capability to manage the TSF as follows:

- Administrator Role — manage the TOE locally; manage the audit mechanism; perform on-demand integrity tests; import and remove X509v3 certificates into/from the trust anchor database; manage the ACL of the Admin and CA sites; manage the CRL store for path validation; configure default TOE access banner; disable CA accounts
- Auditor Role — Review and search the audit data

The CA sites provides and restrict the capability to manage the TSF as follows:

- Administrator Role — configure and manage certificate profiles; modify revocation configuration; configure subscriber self-service constraints; configure certificate revocation list functions; configure OCSP function; configure automated certificate approval management; configure EST settings.
- Auditor —Review and search the audit data.
- CA Operations Staff — approve and execute the issuance of certificates; approve certificate revocation.

Only users in an admin role can access the TOE admin interfaces admin site and CA site; and no management capability are accessible prior to a user being identified and authenticated.

A user role is defined by the permissions assigned to the user on the ACL of the TOE protected resources. The admin role includes the admin permission; the auditor role includes the audit permission, the CA Operations Staff roles consists of one or more of the permissions: certify, revoke, DBAccess and RAMI. A user cannot assume two roles in the TOE, if a user has admin permission, the user cannot be assigned any of the other permissions; if a user has certify permission, they cannot be assigned ‘admin’ or ‘audit’ permissions; they can only be assigned the other 3 permissions that comprises the CA Operations Staff role.

Note: The TOE does not provide the capability to delete individual entries from the audit trail.

2.6.4.2 Guidance Activities

> If the role restriction mechanism is configurable, the evaluator shall examine the operational guidance to determine that the necessary instructions to meet each iteration of the FMT_MOF.1 requirement for the TOE in its evaluated configuration are provided. This applies only to management functions implemented by or accessible through the TSF.

[CCECG] Section ***4 Managing the TOE*** describes all management capabilities that the TSF is required to provide and restrict. [CCECG] Section ***1.4 Interfaces*** and ***1.5 Privileged User Roles*** describes the TOE user roles and what each role can access. User roles in the TOE correspond to the permissions assigned to the user. A user in an administrator role is assigned the admin permission; a user with auditor role is assigned the audit permission; a user in the CA Operations Staff role is assigned one or more of the following permissions: certify, revoke, RAMI, or DBAccess.

2.6.4.3 Test Activities

> Testing only applies to functions implemented by or accessible through the TSF.
>
> The evaluator shall, for each management function, assume the role defined for that function and demonstrate that the assigned role can perform the functions. The evaluator shall, for each management function, assume each role not assigned to that function, attempt to use the function, and verify that the TSF does not permit it.

The evaluator verified that users with Administrative, Operations and Audit permissions were appropriately restricted in what TOE functionality they could access.

2.6.5 Management of Security Functions Behavior (Auditor Functions) (FMT_MOF.1(5))

2.6.5.1 TSS, Guidance, and Test Activities

> Testing for this requirement is defined under FMT_MOF.1(4). The only difference between the iterations of FMT_MOF.1 is the specific set of management functions that are available to each administrative role. Testing for this SFR is conducted sufficiently thoroughly if the evaluator can demonstrate that the assigned role can perform only the functions specified in the SFR.

Assurance Activity for this SFR is defined under FMT_MOF.1(4).

2.6.6 Management of TSF Data (FMT_MTD.1)

2.6.6.1 TSS Activities

> The evaluator shall examine the TSS to determine that, for each administrative function identified in the operational guidance; 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.

[ST] Section ***9.6.6 FMT_MTD.1 Management of TSF Data*** states that none of the administrative functions listed in [ST] ***9.6 Security Management (FMT)*** are accessible through an interface prior to administrator log-in. When accessing the Admin Site or a CA Account Site from a browser, the browser will prompt for the user's certificate. The TOE will identify the certificate and validate it against the access control list and permission requirements of the requested URL. If the user is authorized, the Welcome page of the site will appear with a navigation panel to select the administrative tasks. If the user is not authorized, a page with an error message is displayed.

2.6.6.2 Guidance Activities

> The evaluator shall examine the operational guidance to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of this PP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

[CCECG] Section ***4 Managing the TOE*** describes all management capabilities that the TSF is required to provide and restrict. [CCECG] Section ***1.4 Interfaces*** and ***1.5 Privileged User Roles*** describes the TOE user roles and what each role can access. User roles in the TOE correspond to the permissions assigned to the user. A user in an administrator role is assigned the admin permission; a user with auditor role is assigned the audit permission; a user in the CA Operations Staff role is assigned one or more of the following permissions: certify, revoke, RAMI, or DBAccess.

2.6.6.3 Test Activities

> The evaluator shall ensure that all TSF data specified in the ST can be managed in the ways specified in the ST by Administrators, and that non-administrative roles are not authorized to manage TSF data. This activity may be performed in the course of performing other testing and does not necessarily need to be done as a separate test.

This test was performed in conjunction with other tests which required the evaluator to perform administrative actions on the TOE. All of those required the evaluator to login with a client certificate that had been assigned Admin privileges (see FMT_SMR.2 tests for examples of how this is done). Without such a certificate it was not possible to access the System Administration page.

2.6.7 Specification of Management Functions (FMT_SMF.1)

2.6.7.1 TSS Activities

> There are no TSS assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

[ST] Section ***9.6.7 FMT_SMF.1 Specification of Management Functions*** describes security management capabilities provided and restricted by the TOE.

2.6.7.2 Guidance Activities

> The evaluator shall check to make sure that every management function mandated by the PP is described in the operational guidance and that the description contains the information required to perform the management duties associated with the management function.

[CCECG] Section ***4 Managing the TOE*** describes all management capabilities that the TSF is required to provide and restrict. [CCECG] Section ***1.4 Interfaces*** and ***1.5 Privileged User Roles*** describes the TOE user roles and what each role can access. User roles in the TOE correspond to the permissions assigned to the user. A user in an administrator role is assigned the admin permission; a user with auditor role is assigned the audit permission; a user in the CA Operations Staff role is assigned one or more of the following permissions: certify, revoke, RAMI, or DBAccess.

The TOE's web interfaces are accessed using the environmental web browser. The TOE allows the following management functions to be performed.

1. Ability to manage the TOE locally and remotely;
    a. Local management of the TOE is performed using the local tools.
    b. Remote management is performed using one of the TOE's web interfaces accessed via the Operational Environment's web browser.
2. Ability to perform updates to the TOE;
    a. Updates to the TOE are performed using the update tool.
3. Ability to perform archival and recovery;
    a. The PKCS#11 Cryptographic Device (Thales Luna Network HSM) provides a cloning mechanism.
    b. To clone a key on the HSM to another HSM:
        i. Log into the Operational Environment's Operating System as a Local Administrator.
        ii. Attach a second Thales Luna Network HSM to the computer and configure it per HSM vendor guidance.
        iii. Use the Thales Luna tools installed in the Operational Environment's Operating System to clone the current HSM to the second HSM per HSM vendor guidance.
4. Ability to manage the audit mechanism;
    a. Management of the audit mechanism is performed using the TOE's Admin Site web interface. Click the Configure link under the Audit Trails heading in the navigation pane.
5. Ability to configure and manage certificate profiles;
    a. Certificate profiles can be created using the local tools.
    b. Certificate profiles can be created and managed using the TOE's CA Account Site web interface. Click the Certificate Profiles link under the Preferences heading in the navigation pane. When profiles exist, select the Active Profile from the list in the upper right corner of the web pages to control which profile is being managed.
6. Ability to approve and execute the issuance of certificates;
    a. Certificate requests are approved and issued using the TOE's CA Account Site web interface, EST, or RAMI.

   b. In the TOE's CA Account Site web interface, click the Pending link under Certificate Requests in the navigation pane, find the request to be issued, and click the Issue button next to it.

   c. EST is enabled using the TOE's CA Account Site web interface. Click the Enrollment link under Preferences in the navigation pane and then the EST tab.

7. Ability to approve certificate revocation;

   a. Certificate revocation is approved using the TOE's CA Account Site web interface, RAMI, or through the subscriber self-service portion of the TOE's Public Site web interface.

   b. In the TOE's CA Account Site web interface, click the Valid link under Certificates in the navigation pane, find the certificate to be revoked, and click the Revoke button next to the certificate to be revoked.

8. Ability to modify revocation configuration;

   a. Certificate revocation options are configured using the TOE's CA Account Site web interface. Click the Revocation Policy and CRL Processing links under Preferences in the navigation pane.

9. Ability to configure subscriber self-service request constraints;

   a. Subscriber self-service options are configured using the TOE's CA Account Site web interface. Click the Public Site link under Preferences in the navigation pane.

10. Ability to perform on-demand integrity tests;

    a. The on-demand integrity tests are run using the TOE's Admin Site web interface. Click the NIAP Conformance link under Servers in the navigation pane and then click one of the Run Integrity Test links.

11. Ability to destroy sensitive user data when no longer needed;

    a. Any sensitive data stored by the TOE can be destroyed by destroying the keys on the PKCS#11 Cryptographic Module that are used to encrypt that data.

    b. To destroy keys on the Thales Luna Network HSM:

       i. Log into the environmental Operating System as a Local Administrator.

       ii. Use the Thales Luna tools installed in the environmental Operating System to destroy the keys, or clear the device, per HSM vendor guidance.

12. Ability to import and remove X.509v3 certificates into/from the Trust Anchor Database;

    a. Certificates can be added to the Trust Anchor list using the local tools.

    b. Certificates can be added or removed from the Trust Anchor list using the TOE’s Admin Site web interface. Click the NIAP Conformance link under Servers in the navigation pane and then click the Manage Trust Anchors link.

13. Ability to configure automated process used to approve the revocation of a certificate or information about the revocation of a certificate;

    a. Automated processes for CRLs are configured using the TOE’s Admin Site web interface and the TOE’s CA Account Site web interface.

        i. In the Admin Site, click the Jobs link under Servers in the navigation pane to control the automatic job that removes expired certificates from the next CRL to be issued.

        ii. In the CA Account Site, click the CRL Processing link under Preferences in the navigation pane to control automated CRL issuance.

14. Ability to modify the CRL configuration;

    a. Certificate revocation options are configured using the TOE’s CA Account Site web interface. Click the Revocation Policy and CRL Processing links under Preferences in the navigation pane.

15. Ability to modify the OCSP configuration;

    a. OCSP options, for issuers hosted by the TOE, are configured using the TOE’s CA Account Site web interface. Click the OCSP Responder link in the navigation pane to control the OCSP responder.

16. Ability to configure the cryptographic functionality;

    a. Cryptographic functionality is configured using the TOE’s Admin Site web interface, CA Account Site web interface, and the PKCS#11 Cryptographic Module’s tools.

    b. The asymmetric algorithm used for the “System” credential can be configured through the TOE’s Admin Site web interface. Click the Credentials link under Local System in the navigation pane and then click the Update button a follow the pages to select the key type, size, and message digest to use.

    c. The asymmetric algorithm used for an Issuer or Root credential can be configured through the TOE’s CA Account Site web interface. The first time an Administrator logs in to the CA Account a “Click here to obtain a certificate” link is displayed. Clicking that link or clicking the Credentials link under the navigation pane followed by clicking the New Credential button, allows the selection of the key type, size, and message digest to use for the Root certificate or PKCS#10 certificate request.

d. The message digest used when issuing CRLs can be configured through the TOE's CA Account Site web interface. Click the CRL Processing link under Preferences in the navigation pane.

e. The message digest used when creating OCSP responses can be configured through the TOE's CA Account Site web interface. Click the OCSP Responder link under Preferences in the navigation pane.

f. The message digest used when issuing certificates can be configured through the TOE's CA Account Site web interface. Click the Certificate Issuance link under Preferences in the navigation pane.

g. The asymmetric algorithms that the TOE will accept in certificate requests can be configured through the TOE's CA Account Site web interface. Click the Enrollment link under Preferences in the navigation pane.

h. The only supported TLS version is 1.2 and is not configurable. The supported ciphersuites can be configured by a local Administrator using the environmental Operating System's text editor to modify a TOE configuration file. The available ciphersuites are:

    i. TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

    ii. TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

2.6.7.3 Test 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 PP 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 of the management functions required by this SFR are exercised in the course of performing other test activities. All management is done through the TOE's TLS/HTTPS interface and every test that requires the use of management functionality demonstrates the use of this interface.

2.6.8 Restrictions on Security Roles (FMT_SMR.2)

2.6.8.1 TSS Activities

The evaluator shall examine the TSS to ensure it identifies the roles, the privileges granted to and limitations of each role, and whether they are implemented by the TOE or by the TOE in conjunction with its environment. The evaluator shall also examine the TSS to ensure it describes the interfaces available to each role and how role separation is ensured.

[ST] Section ***9.6.8 FMT_SMR.2 Restrictions on Security Roles*** describes the user roles and the privileges and limitations of each role.

The TOE maintains the Administrator, Auditor, and CA Operations Staff roles available via the following interfaces:

Role	Permission	Interface
Administrator	admin	Admin Site Admin
		CA Account Site Admin
		Operating System
		DBAccess service access (Implicitly granted by the admin permission)
Auditor	audit	Admin Site
		CA Account Site
		DBAccess service access (Implicitly granted by the admin permission)
CA Operations Staff	certify	CA Account Site
	revoke	CA Account Site
	RAMI	RAMI Interface
	DBAccess	DBAccess service access

Unless the role restriction option is disabled:

- The TOE refuses to allow the same certificate to be granted Audit and Admin permission on the All Servers Access Control List which controls access to the Admin Site.
- The TOE refuses to allow the same certificate to be granted Audit permission on a CA Account's ACL and any other right on that same CA Account ACL.
- The TOE refuses to allow the same certificate to be granted CA Operations Staff permission on a CA Account's ACL and any other right on that same CA Account ACL.

The table below provides an example configurations and whether or not they would be allowed.

Site	Certificate	Permission(s)	Allowed
Admin Site	CN=Adam	admin	Yes
Admin Site	CN=Eve	audit	Yes
Admin Site	CN=Adam	admin & audit	No (the TOE will prevent this from occurring, or prevent access by Adam if it occurs)
CA Account A	CN=Adam	admin	Yes
CA Account A	CN=Eve	revoke	Yes (even though Eve has audit on the Admin Site she lacks it on CA Account A and it doesn't conflict)
CA Account A	CN=Jane	audit	Yes
CA Account A	CN=Bob	certify & revoke	Yes
CA Account A	CN=Adam	admin & revoke	No (the TOE will prevent this from occurring, or prevent access by Adam if it occurs)
CA Account B	CN=Adam	certify & revoke	Yes (even though Adam has admin permission for CA Account A and for the Admin Site he lacks those for CA Account B and it doesn't conflict)
CA Account B	CN=Jane	admin	Yes (even though Jane has revoke permission for CA Account A she only has admin permission for CA Account B and there is no conflict)

2.6.8.2 Guidance Activities

The evaluator shall examine the AGD documents to ensure they contain instructions for using either the TOE or the TOE in conjunction with its environment to assign roles to the corresponding users.

The evaluator shall review the operational guidance to ensure that it contains instructions for how the roles connect to and perform operations on the TOE and which interfaces are supported.

[CCECG] Section ***1.4 Interfaces*** and [CCECG] Section ***1.5 Privileged User Roles*** describe the TOE user roles and the actions that each role can perform. A user in an administrator role is assigned the admin permission; a user with auditor role is assigned the audit permission; a user in the CA Operations Staff role is assigned one or more of the certify, revoke, RAMI or DBAccess permissions. The CC-Guide in Section ***5.7.4 FMT_SMR.2 Restrictions on Security Roles*** describes restrictions on security roles. Section ***4.4.2.2.2 Restrictions on Security Roles*** describes how the 3 roles supported by the TOE are managed. Role assignments consistent of assigning permissions to a certificate when adding it to the ACL of a resource.

2.6.8.3 Test Activities

The evaluator shall perform the following tests:

> Test 1: For each supported role, the evaluator shall assume the role and connect to the TOE as specified in the AGD documentation. The evaluator shall verify that the role can perform the documented operations.

This requirement is covered by **FMT_MOF.1(4) Test 1** and all other tests. All tests which require the TOE to perform administrative actions on the TOE demonstrate the ability to assume the role of Administrator and connect to the TOE. All tests which require the approval, denial or revocation of certificates demonstrate the ability to assume the role of CA Operations Staff. The ability to access audit records demonstrates the ability to assume the role of Auditor.

> Test 2: The evaluator shall attempt to assume the Auditor role in conjunction with any other role as defined in FMT_SMR.2.1 and shall verify it is not possible.

The evaluator attempted to modify the permissions of a user who was given the Auditor role. It was not possible to select any other roles for that user.

> Test 3: The evaluator shall attempt to assume the CA Operations Staff role in conjunction with any other role as defined in FMT_SMR.2.1 and shall verify it is not possible.

The evaluator attempted to modify the permissions of a user who was given the Operations role. It was not possible to select any other roles for that user.

2.7 Protection of the TSF (FPT)

2.7.1 Failure with Preservation of Secure State (FPT_FLS.1)

2.7.1.1 TSS Activities

> The evaluator shall examine the TSS to determine that the TOE's implementation of the fail secure functionality is documented. The evaluator shall first examine the TSS section to ensure that all failure modes specified in the ST are described. The evaluator shall then ensure that the TOE will attain a secure state after inserting each specified failure mode type. The evaluator shall review the TSS to determine that the definition of secure state is defined and is suitable to ensure protection of key material and user data.

[ST] Section ***9.7.1 FPT_FLS.1 Failure with Preservation of Secure State*** describes the how the TOE preserves a secure state when failure occurs. For DRBG with the ISC CDK and integrity failure on the trust anchor database, or if the database is inaccessible, the TOE will record the error, destroy any sensitive data, and shuts down the CertAgent service. For a DRBG failure related to the PKCS#11 crypto module in the OE, the TOE will abort the actions, records the audit events, and return an error message.

The following table lists the possible faults and the action taken by the TOE when they occur.

Failure	Action

ISC CDK failure causing the hard error state including a failure of the DRBG	The TOE aborts the action, records the error in the audit trail, and local debug text file, destroys any sensitive data, and shuts down the CertAgent service in an orderly manner
Integrity failure on Trust Anchor database	The TOE records the error in the audit trail, and local debug text file, destroys any sensitive data, and shuts down the CertAgent service in an orderly manner
PKCS#11 failure including failure of the device's DRBG	The TOE aborts the action, records the error in audit trail, and returns an error message
Database inaccessible	The TOE aborts the action, records the error in a local debug text file, destroys any sensitive data, and shuts down the CertAgent service in an orderly manner

The following states are the TOE's secure states:

- The TOE is shutdown in an orderly manner.
- The TOE is running, but refusing to perform operations.

2.7.1.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it describes the actions that might occur and provides remedial instructions for the administrator.

[CCECG] Section ***5.8.1 FPT_FLS.1 Failure with Preservation of Secure State*** describes how the TOE maintains a secure state when failure occurs.

The TOE preserves a secure state when the following types of failures occur: RBG failure, ISC CDK failure, integrity failure on Trust Anchor list or ACLs, PKCS#11 Cryptographic Module failure, and database failure.

When the TOE detects a failure in itself that prevents operations from continuing, it shuts itself down in an orderly manner. This shutdown process is the same process that is used to shut down the TOE prior to a system restart. Plaintext keys and unencrypted user data are cleared from memory during this process leaving only encrypted keys and encrypted user data within the environmental storage.

Once the TOE has been shut down, no services will be available. A local administrator must login to the Operational Environment and identify the error from the local server log file.

2.7.1.3 Test Activities

The evaluator shall perform the following test:

Test 1: The evaluator shall attempt to cause each documented failure to occur and shall verify that the actions taken by the TSF are those specified in FPT_FLS.1.1. For those failures that the evaluator cannot cause, the evaluator shall provide a justification to explain why the failure could not be induced.

The evidence in the [Test] Report Section 8.7.1 includes failure states for each of the mentioned claims for both RHEL and windows:

- DRBG failure - The evaluator installed an invalid CDK file that caused the TOE to fail DRBG checks and verified the TOE was in a CDK error state due to DRBG failure. The evaluator restored the old files and the TOE returned to a normal state.

- ISC CDK Failure - The evaluator installed an invalid CDK file that caused the TOE to fail DRBG checks and verified the TOE was in a CDK error state due to DRBG failure. The evaluator restored the old files and the TOE returned to a normal state.

- PKCS#11 Cryptographic Module failure - The evaluator was able to issue a CRL due to the Module being available. When the module became unavailable, crypto functions were not able to be executed.

- Integrity failure on Trust Anchor database - The evaluator altered the database integrity by deleting a line from CA_TRUST. The log and the output from the browser explain that access wasn't allowed to the TOE due to an integrity failure. The evaluator reverted the settings back to the original settings and the TOE was able to pass the integrity checks.

- Database inaccessible – The evaluator verified the database was accessible and the user could login to the TOE. The evaluator turned the database service off. The TOE was unable to be accessed from the browser due to the database being shut down. The database was turned back on and the TOE was accessible again.

2.7.2 No Plaintext Key Export (FPT_KST_EXT.1)

2.7.2.1 TSS Activities

The evaluator shall examine the TSS to ensure it lists all keys that are not exported from the TOE for all platforms listed in the TOE's ST.

[ST] Section ***9.7.2 FPT_KST_EXT.1 No Plaintext Key Export*** states that all keys are listed in [ST] Section 9.3.7. As shown in that table, the TOE protects the symmetric keys used to encrypt sensitive data stored in the database. The TOE encrypts those symmetric keys with the public key from the "System" credential whose private key is managed by the PKCS#11 Cryptographic Module. The TOE interface provides no way to export those keys in any form.

In the evaluated configuration, the PKCS#11 Cryptographic Module manages the "System" credential's private key, the TLS server private key, and all issuer private keys. The module provides no way to export the keys in plaintext.

2.7.2.2 Guidance Activities

There are no AGD assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

[CCECG] Section ***5.8.2 FPT_KST_EXT.1 No Plaintext Key Export, FPT_KST_EXT.2 TSF Key Protection*** states that neither the TOE nor the PKCS#11 Cryptographic Module in the OE provides a mechanism for plaintext key export.

2.7.2.3 Test Activities

The evaluator shall perform the following test:

Test 1: The evaluator shall access the export interface of the TOE and shall verify that the interface prevents the export of all keys listed in the TSS.

As described in AGD [CCECG] section 5.8.2 there is no interface in the TOE for the export of keys. Throughout the course of testing this product the evaluator exercises the TOE's user interface and did not discover any undocumented key export functionality.

2.7.3 TSF Key Protection (FPT_KST_EXT.2)

2.7.3.1 TSS Activities

The evaluator shall examine the TSS to ensure it describes how unauthorized use of TSF private and secret keys is prevented for both users and processes.

[ST] Section ***9.7.3 FPT_KST_EXT.2 TSF Key Protection*** states that the TOE provides no interfaces where unauthorized users or unprivileged processes can access private and secret keys used by the TOE. All users accessing TSF data, or performing TSF provided and restricted functions, are identified and authenticated except when accessing the limited functions permitted by FIA_UIA_EXT.1.1 without prior authentication. The TOE uses ACLs to restrict privileged user actions based on roles. The TOE protects the symmetric keys it manages using a key hierarchy chaining to a single asymmetric REK, the "System" credential, which is generated, stored, and protected by the PKCS#11 Cryptographic Module which protects keys using hardware. Thus the TOE ensures that unauthorized users and unprivileged processes cannot access its private and secret keys.

2.7.3.2 Guidance Activities

The evaluator shall examine the AGD guidance to ensure it contains instructions for configuring the TOE or Operational Environment to prevent unauthorized access to TSF secret and private keys by users or processes.

[CCECG] Section ***5.8.2 FPT_KST_EXT.1 No Plaintext Key Export, FPT_KST_EXT.2 TSF Key Protection*** o states the TOE restricts access to operations that would use the issuer keys using client authenticated web pages. No user can use any key unless they've been successfully authenticated as a privileged user. Thus the TOE ensures that unauthorized users and unprivileged processes cannot access its private and secret keys.

The HSM provides its own protection mechanisms to prevent unauthorized users and unprivileged processes access to its protected functions and data. The TOE must authenticate to the PKCS#11 Cryptographic Module when the TOE starts using a password in order to access the cryptographic services of the PKCS#11 Cryptographic Module.

There is no configuration on the TOE or the PKCS#11 Cryptographic Module to change the above behavior.

2.7.3.3 Test Activities

The evaluator shall perform the following test:

> Test 1: The evaluator shall assume each of the non-Administrator roles supported by the TOE and shall attempt to use the available TOE interface to access the keys. The evaluator shall verify that these attempts fail.

As described in [Test] Report **Section 8.7.3 FPT_KST_EXT.2 Test 1**, there is no interface in the TOE for the export of keys. Throughout the course of testing this product the evaluator exercise the TOE's user interface and did not discover any undocumented key export functionality.

2.7.4 Manual Trusted Recovery (FPT_RCV.1)

2.7.4.1 TSS Activities

> The evaluator shall examine the TSS to determine that it describes how the TOE enters a maintenance mode after a failure and the possible actions that can take place while in that mode.

[ST] Section ***9.7.4 FPT_RCV.1 Manual Trusted Recovery*** states that when in maintenance mode, the TOE prevents normal operations and limits privileged user, subscriber, and relying party actions so that only an administrator may log on and correct the failure issue(s). All other functions (EST, OCSP, issuance, etc.) are disabled. When in maintenance mode the NIAP restrictions that are not enforced are:

- Requiring data integrity on the Trust Anchor list used for certificate path validation
- Requiring data integrity on the ACLs
- Checking integrity of the Trust Anchor and ACLs when the TOE starts
- Using strict certificate path validation
- Enforcing role separation

2.7.4.2 Guidance Activities

> The evaluator shall examine the AGD guidance to ensure it contains instructions for restoring the TOE to a secure state when it enters the maintenance mode, including the steps necessary to perform while in this state.

[CCECG] Section ***4.1.2 Starting the Service in Maintenance Mode*** and Section ***5.8.3 FPT_RCV.1 Manual Trusted Recovery*** provide the instructions for restoring the TOE to a secure state after an integrity failure has been detected.

When the TOE is in maintenance mode, the TOE prevents normal operations and limits privileged user, subscriber, and relying party actions so that only an administrator may log on and correct the integrity failure. All other functions (EST, OCSP, issuance, etc.) are disabled. When in maintenance mode, the NIAP restrictions that are not enforced are:

- Requiring data integrity on the Trust Anchor list used for certificate path validation
- Requiring data integrity on the ACLs
- Checking integrity of the Trust Anchor and ACLs when the TOE starts
- Using strict certificate path validation
- Enforcing role separation

To restore a secure state in the case of integrity failure, an administrator needs to remove all certificates from the corresponded list and reimport the certificates to the list via the Admit Site or CACLI.

2.7.4.3 Test Activities

The evaluator shall perform the following test:

Test 1: The evaluator shall attempt to cause each documented failure to occur and shall verify that the result of this failure is that the TSF enters a maintenance mode. The evaluator shall also verify that the maintenance mode can be exited and the TSF can be restored to a secure state. This testing may be performed in conjunction with FPT_FLS.1.

The following evidence was gathered in conjunction with in the [Test] Report **Section 8.7.1 FPT_FLS.1 Test 1** and **8.7.1 FPT_TST_EXT.2 Test 2** for both RHEL and windows.

That test shows the TOE behaving as described when it enters a failure state and then being restored as described in the Section ***5.8.3 FPT_RCV.1 Manual Trusted Recovery*** [CCECG] .

2.7.5 Protection of Keys (FPT_SKP_EXT.1)

2.7.5.1 TSS Activities

Regardless of whether this requirement is met by the TOE or the Operational Environment, the evaluator shall examine the TSS to determine that it details each persistent private and secret key needed to meet the requirements in the ST. For each of these items, the evaluator shall confirm that the TSS details how any secret or 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.

[ST] Section ***9.7.5 FPT_SKP_EXT.1 Protection of Keys*** states that the TOE provides no mechanisms allowing the reading of any pre-shared, private, or secret keys. The PKCS#11 Cryptographic Module maintains its own protections of keys it holds and in the evaluated configuration does not provide any mechanism for reading those keys.

[ST] Section ***9.3.2 FCS_STG_EXT.1 Cryptographic Key Storage*** identifies the secret and private keys and their purpose, protection and location.

Key	Purpose	Storage	Protection
CA Issuers (asymmetric)	Signing certificates, CRLS, and OCSP responses	PKCS#11 Cryptographic Module	Protected by the PKCS#11 Cryptographic Module.
TLS/HTTPS Server Key (asymmetric)	Server Authentication	PKCS#11 Cryptographic Module	Protected by the PKCS#11 Cryptographic Module.
“System” Credential (asymmetric)	Encryption of CA secrets (HSM PINs, database password, etc.), and signing the Trust Anchor and CRL tables for data integrity check	PKCS#11 Cryptographic Module	Protected by the PKCS#11 Cryptographic Module.
TLS/HTTPS Client Key (asymmetric)	Authentication	Web Browser key store	Protected by the web browser.
CA CMS DEKs	Data encryption	Stored with the encrypted data in the database	Encrypted using the CA’s “System” credential’s public key.
OCSP Signers (asymmetric)	Signing OCSP responses	PKCS#11 Cryptographic Module	Protected by the PKCS#11 Cryptographic Module.

2.7.5.2 Guidance Activities

The evaluator shall examine the AGD guidance to ensure it contains any necessary instructions for configuring the TOE or Operational Environment to support this requirement.

[CCECG] Section ***5.8.4 FPT_SKP_EXT.1 Protection of Keys of the CC-Guide*** states that the TSF provides no mechanisms allowing the reading of any pre-shared, private, or secret keys. The PKCS#11 Cryptographic Module maintains its own protections of keys it holds and in the evaluated configuration does not provide any mechanism for reading those keys.

2.7.5.3 Test Activities

The evaluator shall perform the following test:

Test 1: The evaluator shall assume each of the non-Administrator roles supported by the TOE and shall attempt to use the available TOE interface to read the keys specified by the TOE. The evaluator shall verify that these attempts fail.

As described in [CCECG] section 5.8.4 there is no interface in the TOE for the reading of keys. Throughout the course of testing this product the evaluator exercise the TOE's user interface and did not discover any undocumented key export functionality.

2.7.6 Reliable Time Stamps (FPT_STM.1)

2.7.6.1 TSS Activities

The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time. The TSS provides a description of how the time is maintained and considered reliable in the context of each of the time related functions.

[ST] Section ***9.7.6 FPT_STM.1 Reliable Time Stamps*** states that time stamps are based on the environmental Operating System's clock and managed by the environmental Operating System. The time is reliable for each of the TOE's purposes as the time is controlled by trusted administrators and maintained by the trusted platform on which the TOE operates.

The current system time is used when: generating audit records, issuing certificates, CRLs, and signing OCSP responses. The SFRs that use time are: FAU_GEN.1.2, FCO_NRO_EXT.2.2, FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1, FDP_CER_EXT.3, FDP_CSI_EXT.1, FDP_CRL_EXT.1, FDP_OCSPG_EXT.1.1, FIA_X509_EXT.1, FIA_X509_EXT.2, and FTA_SSL.3.

2.7.6.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it instructs the administrator how to set the time. If the TOE supports the use of a network time protocol (NTP) server, the operational guidance shall describe 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.

The TOE does not support the use of an NTP server. [CCECG] Section ***5.8.5 FPT_STM.1 Reliable Time Stamps*** of the CC-Guide includes instructions for changing the time on the OS platform.

2.7.6.3 Test Activities

The evaluator shall perform the following tests:

Test 1: The evaluator shall use the operational guidance to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.

The evaluator altered the time on the TOE's platform as described in the AGD [CCECG]. This time change was reflected in the TOE's interface.

Test 2: [conditional] If the TOE supports the use of an NTP server; the evaluator shall use the operational guidance 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 operational guidance.

The TOE does not support the use of an NTP server.

2.7.7 Integrity Test (FPT_TST_EXT.2)

2.7.7.1 TSS Activities

> The evaluator shall examine the TSS to ensure it describes the mechanisms that will be used to verify the integrity of the selected data and the action(s) taken if any of the integrity tests fails.

[ST] Section ***9.7.8 FPT_TST_EXT.2 Integrity Test*** states that the TOE verifies the integrity of the trust anchor table and the ACL table when the TOE starts, whenever any protected table is changed, and on-demand when requested through the NIAP section of the Admin Site.

[ST] Section ***9.4.6 FDP_STG_EXT.1 Public Key Protection*** describes the protection of the keys. The integrity of the CertAgent trust anchor table, and the tables storing the ACLs, is maintained using a digital signature created using the CA “System” credential. This signature is validated when the table is used. The signature is updated whenever an administrator modifies the trust list. By digitally signing the tables, and only using them if the signature is valid, the TOE prevents unauthorized users from changing the tables using methods other than those described herein. Access to these tables is controlled by the TOE’s web interface, through client authentication and ACL permissions, and by the environmental Operating System’s local console access control. Individuals who need to modify any of these lists are authenticated by either the TOE or the environmental Operating System.

If the integrity check fails, the TOE behaves as described in [ST] Section 9.7.4. After a failure of integrity is detected, the TOE shuts itself down.

2.7.7.2 Guidance Activities

> The evaluator shall examine the operational guidance to ensure that it includes instructions to verify the integrity of the selected data.

[CCECG] Section ***5.8.7 FPT_TST_EXT.2 Integrity Test*** of the CC-Guide includes instructions to verify the integrity of the trust anchor database. The TSF automatically does integrity checking at the TOE startup, the CC-guide in section ***4.4.2.1.2 Running Integrity Test on Demand*** provides the steps for performing manual integrity checks.

2.7.7.3 Test Activities

The evaluator shall perform the following tests:

> Test 1: The evaluator shall use the operational guidance instructions to verify the integrity of each protected element specified in the TSS.

The evaluator ran the two self-tests manually as described in 4.4.2.1.2 of the AGD [CCECG]. Each test passed and audit records were generated to verify the passing of the tests.

> Test 2: The evaluator shall modify an instance of each type of data selected in FPT_TST_EXT.2.1 to verify the integrity test fails and the action defined in FPT_TST_EXT.2.2 occurs. If this test cannot be performed, the evaluator shall provide a justification.

The evaluator tested the modification of the trust anchor database as part of section 8.7.1 FPT_FLS.1 Test 1 in the [Test] Report. Section 2.7.1.3 in the AAR also explains in more detail how the test was ran.

The evaluator then modified the ACL database on the TOE by deleting a line from the table. After the line was deleted, the evaluator then verified the TOE went into an error state where it displayed an “integrity error” on the GUI. The evaluator then restored the line in the table with the proper value and the TOE was available again.

2.7.8 Trusted Update (FPT_TUD_EXT.1)

2.7.8.1 TSS Activities

> The evaluator shall verify that the TSS section of the ST describes all TSF software update mechanisms for updating the system 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. The evaluator shall verify that all software and firmware involved in updating the TSF is described and, if multiple stages and software are indicated, that the software/firmware responsible for each stage is indicated and that the stage(s) which perform signature verification of the update are identified.
>
> The evaluator shall verify that the TSS describes the method by which the digital signature is verified.
>
> The evaluator shall verify that the TSS describes that the public key used to verify the signature is either hardware-protected or is validated to chain to a public key in the Trust Anchor Database. If hardware-protection is selected, the evaluator shall verify that the method of hardware-protection is described and that the ST author has justified why the public key may not be modified by unauthorized parties.
>
> [conditional] If the ST author indicates that the public key for software update digital signature verification, the evaluator shall verify that the update mechanism includes a certificate validation according to FIA_X509_EXT.1 and a check for the Code Signing purpose in the extendedKeyUsage.

[ST] Section ***9.7.7 FPT_TUD_EXT.1 Trusted Update*** describes how the TOE implements this SFR. The TOE provides local administrators the ability to check for updates on demand via the update tool. The update tool is a command line program included with the TOE that provides an interface for TOE software update and to verify the validity of the update’s digital signature. The update package is a file consisting of the update data, a digital signature computed over the hash of the update data, and the certificates needed to verify the digital signature. The update data is hashed, the hash is digitally signed and the data and signature are then combined with the certificate chain needed to verify the signature to create the output archive file.

The update tool verifies the validity of the update’s digital signature by hashing the content of the package, verifying the digital signature matches the computed hash value, and validating the certificate per FIA_X509_EXT.1. If the digital signature is valid, the administrator can initiate the installation of an update package. The update tool will stop the TOE, install the update, and restart the TOE. Once initiated the TOE verifies the digital signature on the package and will stop the update process if the signature or the certificate used is not valid.

A signature on the update package can be invalid for the following reasons:

- the hash value computed by the update tool does not match the hash value in the signature
- the certificate used to sign the update package
    - is invalid per FIA_X509_EXT.1

- does not contain an extendedKeyUsage extension with the Code Signing purpose
- does not chain to a certificate in the TOE's trust anchor list

2.7.8.2 Guidance Activities

The evaluator shall examine the operational user to ensure it contains the required information regarding TOE version verification and TOE updates as specified in AGD_OPE.1.

[CCECG] Section ***4.10 Updating the TOE*** describes managing TOE updates and it contains instructions for version verifications of the TOE.

The TOE provides OE Administrators the ability to check for updates on demand via the update tool. The update tool is a command line program included with the TOE that interfaces with the TOE to verify the validity of the update's digital signature, and if valid, stops the TOE, installs the update, and restarts the TOE. At the local console, a local administrator initiates the installation of an update package using the update tool. Once initiated, the TOE verifies the digital signature on the package and will stop the process if the signature or the certificate used is not valid.

The supplied Update tool program can be used to check if an update is required, validate, and install the update package.

```
sudo <ca home>/update/update-tool.sh        (RHEL)
<ca home>\update\update-tool.bat            (Windows)
```

2.7.8.3 Test Activities

The evaluator shall perform the following tests:

Test 1: The evaluator shall perform the version verification activity to determine the current version of the product. The evaluator shall obtain a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Then, the evaluator shall perform a subset of other assurance activity tests to demonstrate that the update functions as expected. After the update, the evaluator shall perform the version verification activity again to verify the version correctly corresponds to that of the update.

The evaluator verified the version of the TOE and then installed a legitimate update file. After completing the update, the evaluator accessed the TOE's public website and uploaded and signed a CSR to verify the TOE functioned correctly. Next another version verification activity was performed. This showed that the TOE was updated as expected.

Test 2: The evaluator shall obtain or produce an illegitimate update, and shall attempt to install it on the TOE. The evaluator shall verify that the TOE rejects the update.

The evaluator obtained in illegitimate update file to install onto the TOE. After the file was uploaded to the TOE, the evaluator then attempted to verify and install the illegitimate update file. The evaluator then verified that the TOE rejected an illegitimate update file due to it being illegitimate. The evaluator verified this with both logs and text output from the TOE.

Test 3: The evaluator shall obtain or produce an update with an invalid signature, and shall attempt to install it on the TOE. The evaluator shall verify that the TOE rejects the update and performs any other actions specified in the TSS.

The evaluator verified that the TOE rejects an update with an invalid signature. The verification tool rejected the updated file. An audit record was generated also generated which provide more context.

Modified by TD0415.

Test 4 [conditional]: If the TOE supports use of X509 certificates for code signing, the evaluator shall digitally sign the update with a certificate that does not have the Code Signing purpose and verify that application installation fails. The evaluator shall repeat the test using a valid certificate and a certificate that contains the Code Signing purpose and verify that the application installation succeeds.

The evaluator verified that the TOE will not accept an update signed by a certificate that does not have the Code Signing purpose. Test 1 demonstrated that the TOE would accept an updated that was signed with a good certificate.

Test 5: The tester shall attempt to install an update without the digital signature and shall verify that installation fails. The tester shall attempt to install an update with altered digital signature, and verify that installation fails.

The invalid signature update file was tested in FPT_TUD_EXT.1 Test 3 in which the TOE rejected an update file that contained an altered signature . Then the evaluator attempted to update the TOE with a update file that lacked the digital signature which lead to the TOE denying the update attempt. Audit records were generated that verified the failure.

2.8 TOE Access (FTA)

2.8.1 TSF-Initiated Termination (FTA_SSL.3)

2.8.1.1 TSS Activities

There are no TSS assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

[ST] Section ***9.8 TOE Access (FTA)*** describes session timeout at the TOE web interface.

2.8.1.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it includes instructions for configuring the inactivity time period for remote interactive sessions.

[CCECG] Section ***5.9.3 FTA_SSL.3 TSF-initiated Termination*** of the CC-Guide provides the instructions for modify the inactive TLS/HTTPS session timeout value. The default value is 30 minutes

To change the TOE's session time-out:

1. Append the following option to the CATALINA_OPTS variable to the Tomcat's startup script (`<ca home>/tomcat.sh or <ca home>\tomcat.bat`):

```
-Disc.ca.web.session.timeout=<time-out value in minutes>
```

1. Restart the TOE. For details, see Section ***4.1.1 Managing the TOE Service***.

2.8.1.3 Test Activities

The evaluator shall perform the following test:

Test 1: The evaluator shall follow the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a remote interactive session with the TOE. The evaluator shall then observe that the session is terminated after the configured time period.

The evaluator configured the TOE for 3 and 5 minute timeout periods. The evaluator opened an inactive session that resulted in the timeout and termination of the traffic that was based on the time period configured.

Examination of TOE audit records showed that sessions were terminated approximately 3 and 5 minutes after the last user activity.

2.8.2 User-Initiated Termination (FTA_SSL.4)

2.8.2.1 TSS Activities

There are no TSS assurance activities for this requirement beyond what is necessary to satisfy the requirements in [CEM].

[ST] Section ***9.8 TOE Access (FTA)*** describes the user logout options available on the TOE Web interface using a logout link and local console using the environmental Operating System.

2.8.2.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it describes how to terminate interactive sessions.

[CCECG] Section ***5.9.1 FTA_SSL.4 User-initiated Termination*** of the CC-guide describes the **Log Out** button available on the Admin site and the CA Accounts site.

2.8.2.3 Test Activities

The evaluator shall perform the following tests:

Test 1: The evaluator shall initiate an interactive local session with the TOE. The evaluator shall then follow the operational guidance to terminate the session and observe that the session has been terminated.

This evaluator logged into the TOE’s local platform. The evaluator verified they were able to successfully login. The evaluator then proceeded to logout of the platform and verified that the user was logged out. Screen shots verify the local logging onto the TOE and termination of the session.

Test 2: The evaluator shall initiate an interactive remote session with the TOE. The evaluator shall then follow the operational guidance to terminate the session and observe that the session has been terminated.

The evaluator logged in to the TOE over an HTTPS connection and then used the logout button to terminate the remote session.

2.8.3 Default TOE Access Banners (FTA_TAB.1)

2.8.3.1 TSS Activities

The evaluator shall examine the TSS to ensure that it details each method of access (local and remote) available to the administrator (e.g., serial port, SSH, HTTPS).

[ST] Section ***9.8 TOE Access (FTA)*** describes the access banner that can be configured on the TOE management interfaces (admin site and CA site). The TSS indicates that the TOE provides web interfaces for remote user access via TLS/HTTPS. Local access to the TOE is provided by the OS platform.

2.8.3.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it includes instructions for how to configure notices and consent warning messages.

[CCECG] Section ***4.4.2.8 Access Banner*** of the CC-guide provides the instructions for configuring a TOE access banner.

2.8.3.3 Test Activities

The evaluator shall perform the following test:

Test 1: The evaluator shall follow the operational guidance 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 used the TOE's administrative interface to configure a notice and consent message and then verified that that message was displayed upon login for each site(CA Site and Administrative Site).

2.9 Trusted Path/Channels (FTP)

2.9.1 Inter-TSF Trusted Channel (FTP_ITC.1)

2.9.1.1 TSS Activities

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, 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 are specified and included in the requirements in the ST.

If an external cryptographic module is selected in FTP_ITC.1.1, the evaluator shall examine the TSS to ensure it describes how the external module is used for cryptographic operations versus how any locally provided cryptographic functionality is used.

[ST] Section ***9.9 Trusted Path/Channels (FTP)*** describes the TOE trusted channel function. The TOE provides a trusted channel to protect communications between itself and an RA, and an optional audit server in the operational environment. The TOE uses TLS/HTTPS with client authenticated HTTPS for this trusted channel. The TOE does not initiate communication with any external entity via the trusted channels. The RA initiates communications over the trusted channel when accessing the TOE resources, the audit server polls the TOE for audit information and retrieve audit records from the TOE audit trail.

For communication between the TOE and environmental components (notably the database and the HSM), the Operational Environment provides a non-encrypted, trusted channel. Secure communication is enforced between the TOE and IT entities in the Operational Environment using the environmental JRE, JNDI, JDBC, and PKCS #11 Cryptographic Module components installed on the local system. These trusted channels transfer TOE data to and from IT entities within the Operational Environment. Trust is established between the TOE and the PKCS#11 Cryptographic Module using the PKCS#11 Cryptographic Module's password (or other authentication mechanism). Trust is established between the TOE and the database using a password.

2.9.1.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be interrupted.

[CCECG] Section ***4.8 Using Database Access Service*** in the CC-guide describes how the privileged user with auditor roles uses DBAccess to retrieve the audit data from the TOE audit trail. [CCECG] Section ***4.7 Using RAMI*** provides the instructions for using the RAMI interface. The TOE establishes a new HTTPS connection for each request submitted to the DBAccess and RAMI interfaces. The [CCECG] Section ***5.10.2 FTP_ITC.1 Inter-TSF trusted channel*** indicates a new HTTPS connection is established for each request submitted to the DBAccess and RAMI API. If the HTTPS connection is interrupted, it cannot be recovered. Simply resubmit the request.

2.9.1.3 Test Activities

The evaluator shall perform the following tests:

Test 1: The evaluator 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 operational guidance and ensuring that communication is successful.

The testing for this requirement is covered by FIA_ESTS_EXT.1 Test 1 and FTP_ITC_EXT.1 test 3. Those tests demonstrate the ability of the TOE to communicate with an RA and DBAccess interface over a secure channel.

Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE.

This test is not applicable. The only communications covered by this requirement are EST communications. In those tests the TOE functions as a responder and does not initiate communications.

Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.

The testing for this requirement is covered by the FIA_ESTS_EXT test 1. That test demonstrates the ability of the TOE to communicate with an RA over a secure channel. All communications are encrypted by TLS.

Then the evaluator provides evidence of the DBAccess communication channel which show the network traffic is indeed in encrypted text.

Those tests demonstrate the ability of the TOE to communicate with an RA/DBAccess interface over an encrypted tunnel. All communications are encrypted by TLS.

Test 4: The evaluator shall, for each protocol associated with each authorized IT entity tested during test 1, cause an interruption to the connection. The evaluator shall ensure that when connectivity is restored, communications are appropriately protected.

The evaluator connected to the TOE using each of the interfaces claimed for secure connections. For each interface used, the evaluator disconnected (interrupting the connection) the TOE from the network while a secure connection was in effect. When the evaluator reconnected the TOE to the network, the evaluator verified that the connection was appropriately restored and protected.

2.9.2 Trusted Path (FTP_ TRP.1)

2.9.2.1 TSS Activities

The evaluator shall examine the TSS to determine that the methods of remote TOE communication 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 communication are consistent with those specified in the requirement, and are included in the requirements in the ST.

[ST] Section ***9.9 Trusted Path/Channels (FTP)*** describes the methods of remote TOE communication. The TOE uses TLS/HTTPS to provide a trusted path for communication with its remote privilege users. The communication is client-authenticated and is initiated by remote users for initial subscriber and privilege user authentication and for all administration actions.

Privileged users, and subscribers that chose to do so, initiate trusted communication using the environmental web browser. The web browser, ensures that the server's HTTPS/TLS server certificate is valid and belongs to the server (by comparing the CN and SAN entries to the server name in the URL specified). If trusted communication cannot be established, both the TOE and the web browser will terminate the connection.

2.9.2.2 Guidance Activities

The evaluator shall examine the operational guidance to ensure it contains instructions for establishing the remote sessions for each supported method.

[CCECG] Section ***5.10.1 FTP_TRP.1 Trusted Path*** of the CC-Guide describes how to establish remote communications with the TOE. The description identifies all interfaces that privileged users can use to access the TOE services and to manage the TOE functions and data.

2.9.2.3 Test Activities

The evaluator shall perform the following tests:

Test 1: The evaluator shall ensure that communications using each specified (in the operational guidance) remote method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.

All administrative and operational actions on the TOE are performed over HTTPS/TLS. Every test that required the use of the TOE's management interfaces demonstrated the use of this trusted path.

Test 2: For each method of remote communication supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote session without invoking the trusted path.

The only way to access the TOE's management interface is by directing a web browser or an API call that requires a trusted path.

Test 3: The evaluator shall ensure, for each method of remote communication, the channel data is not sent in plaintext.

All communications with the TOE are done over HTTPS/TLS. This can be seen in every screenshot of the use of the TOE's management interface where the web browser used for management (Mozilla Firefox) indicates in the address bar that the connection is secured. Additionally, a wire capture was made of the evaluator connecting to and logging into the TOE. It shows that no channel data is sent in plaintext.

Further assurance activities are associated with the specific protocols.

3 Security Assurance Requirements

3.1 Class ADV: Development

3.1.1 ADV_FSP.1 Basic Functional Specification

3.1.1.1 ADV_FSP.1 Evaluation Activity

There are no specific assurance activities associated with these SARs, except ensuring the information is provided. The functional specification documentation is provided to support the evaluation activities described in Section 5.1, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because there is insufficient interface information, then an adequate functional specification has not been provided.

The developer note for this SFR includes this "The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary." The evaluator completed all assurance activity for the SFRs, so the information is provided.

3.2 Class AGD: Guidance Documents

3.2.1 AGD_OPE.1 Operational User Guidance

3.2.1.1 AGD_OPE.1 Evaluation Activity

> Some of the contents of the operational guidance will be verified by the assurance activities in Section 5.1 and evaluation of the TOE according to the CEM. The following additional information is also required.

The [CCECG] Section 1.2 lists all the processes that comprises the TOE, in its evaluated configuration. Sections 3 and 4 provides the instructions for installation and initial configuration of the TOE. Section 2 provides the steps for configuring the OE components required by the TOE including the HSM and the database. Section 4.10 provides instructions for obtaining a TOE update and for initiating the update process. All CertAgent functions included in the CertAgent installation package are included in the evaluated configuration.

3.2.1.2 AGD_OPE.1 Evaluation Activity

> The operational guidance shall at a minimum list the processes that comprise the TOE in its evaluated configuration.
>
> The operational guidance shall contain instructions for configuring the Operational Environment to support the functions of the TOE. These instructions shall include configuration of the cryptographic engine associated with the evaluated configuration of the TOE as well as configuration of the underlying platform. It shall provide a warning to the administrator that use of other cryptographic engines or platforms was not evaluated nor tested during the CC evaluation of the TOE. The documentation must describe the process for installing updates to the TOE. The evaluator shall verify that this process includes the following steps:
>
> - Instructions for obtaining the update. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).
> - Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful.
>
> The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.

In [CCECG] Section 1.2 lists all the processes that comprises the TOE, in its evaluated configuration. Sections 3 and 4 provides the instructions for installation and initial configuration of the TOE. Section 2 provides the steps for configuring the OE components required by the TOE including the HSM and the database. Section 4.10 provides instructions for obtaining a TOE update and for initiating the update process. All CertAgent functions included in the CertAgent installation package are included I the evaluated configuration.

3.2.2 AGD_PRE.1 Preparative Procedures

> As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.

The evaluator shall check to ensure that the following guidance is provided:

- As indicated in the introductory material, administration of the TOE is performed by one or more administrators that are a subset of the group of all users of the TOE. While it must be the case that the overall system (TOE plus Operational Environment [Operational Environment]) provide this capability, the responsibility for the implementation of the functionality can vary from totally the Operational Environment's responsibility to totally the TOE's responsibility. At a high level, the guidance must contain the appropriate instructions so that the Operational Environment is configured so that it provides the portion of the capability for which it is responsible.
- Many of the cryptographic requirements in the PP can be met by the TOE, the Operational Environment, or a combination of the two. The Operational Environment may provide the necessary functionality via use of an external cryptographic module such as a HSM. The guidance must contain the appropriate instructions so that the TOE or Operational Environment is configured to provide the portion of the capability for which it is responsible.

The [CCECG] addresses configuration for both the Windows and the RHEL platforms supporting the evaluated configuration. For initial installation and configuration, the CC-guide clearly marks the steps for Windows and the steps for RHEL. For the management operations, where the steps are different, the CC-guide make the distinction clear. The CC-guide also describes the requirements met by operational environment components including the OS, the HSM and the database. In Section 2, the CC-guide includes instructions for configuring the OE components that supports the TOE including:

Creating users and user roles in the underlying OS that the TOE will use ([CCECG] Section 2.1)

- Changing the time on the OS platform ([CCECG] Section 5.8.5)
- Configuring the PKCS#11 cryptographic module, including all the HSM settings relevant to the TOE. ([CCECG] Section 2.4)
- Installing and configuring the database in the OE that the TOE will use to store its audit trail and to provide its certificate repository. ([CCECG] Section 2.5)
- Firewall settings to ensure that there are no interfaces that a remote user could use to bypass the trusted channel. ([CCECG] Section 2.9).

3.3 Class ALC: Life-Cycle Support

3.3.1 ALC_CMC.1 Labeling of the TOE

The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

The ST and the guidance documentation identifies the TOE as ISC CertAgent version 8.0. The TOE provided for testing was identified as CertAgent version 8.0 patch level 0.2. The TOE label is consistent in all evaluation evidence and in the vendor's site.

3.3.2 ALC_CMS.1 TOE CM Coverage

The “evaluation evidence required by the SARs” in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.

The evaluator checked that the TOE identification is consistent between the ST and the AGD. Both the ST and the AGD identify the TOE as version 8.0 with patch level 0.2.

3.4 Class ASE: Security Target Evaluation

As per activities defined in [CEM].

3.5 Class ATE: Tests

3.5.1 ATE_IND.1 Independent Testing – Conformance

3.5.1.1 ATE_IND.1 Evaluation Activity

The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of this PP’s Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered.

The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a “fail” and “pass” result (and the supporting details), and not just the “pass” result.

The ATE assurance activity are all addressed in the [Test] Report. [TEST] Report describes the testing environment, the tested configuration of the TOE including the configuration of each platform, all setup steps, the test tools used during testing, test procedures descriptions, expected results, and actual test results, as well as a mapping of the SFRs testing to the test cases described in the [Test] Report.

3.6 Class AVA: Vulnerability Assessment

3.6.1 AVA_VAN.1 Vulnerability Survey

3.6.1.1 AVA_VAN.1 Evaluation Activity

> As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in certification authority products, the communications and enrollment protocols used, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, a test would be suitable at the assurance level of this PP. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

The vulnerability report is provided in [AVA]. It includes the result of searches performed on public vulnerability database and for any vulnerability identify for a component of the TOE, the vulnerability survey includes a rationale for why the TOE is not vulnerable.

Searches were performed using the NIST National Vulnerability Database.
(https://nvd.nist.gov/vuln/search).

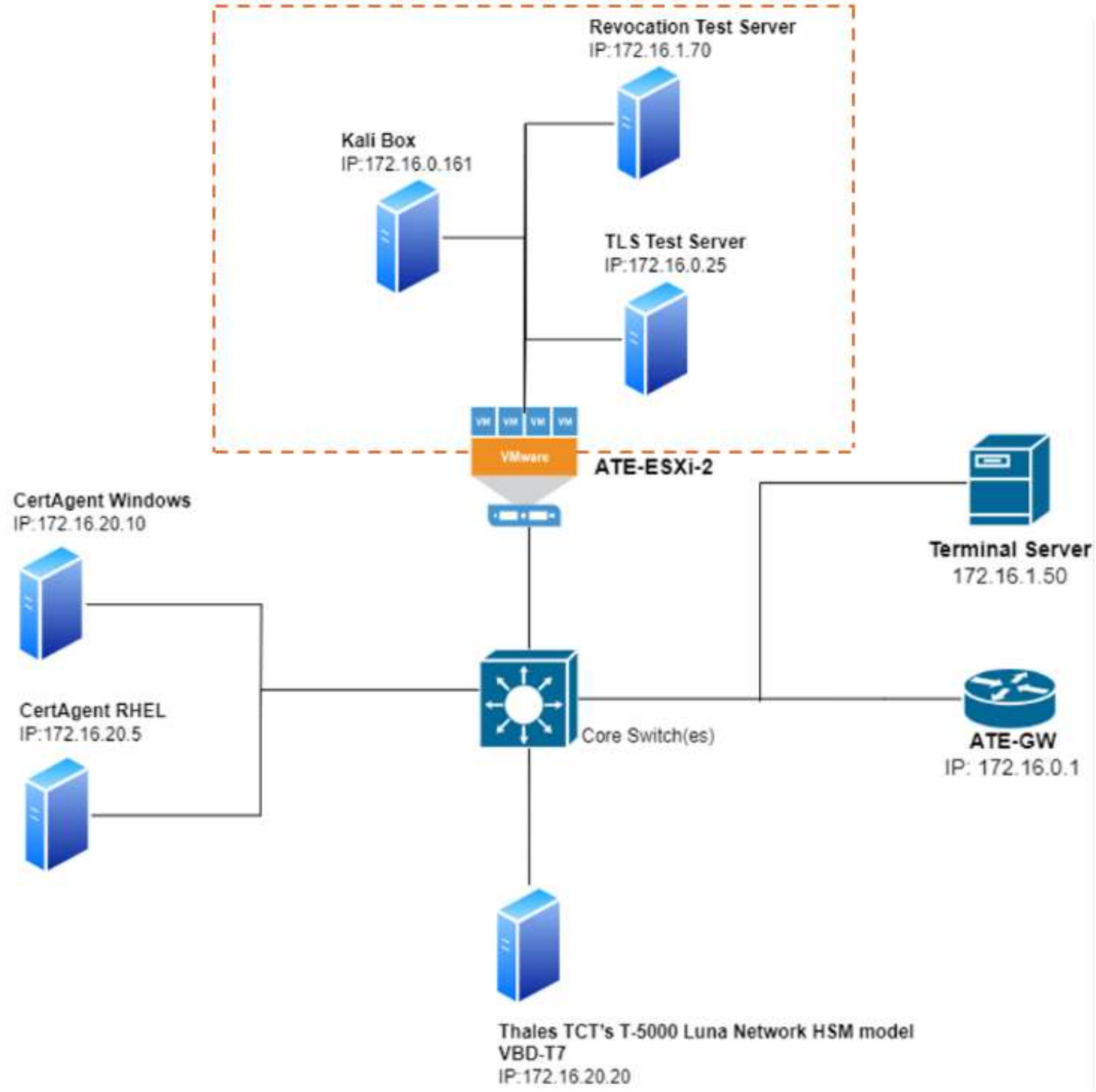
The vulnerability searches were performed on August 23rd, 2024, and are based on the following terms that are known components, modules, and features of the TOE.

- information security corporation
- CertAgent
- Information Security Corporation CertAgent
- ISC's Cryptographic Development Kit
- ISC CDK DRBG
- ISC CDK
- JDK 17.0.12
- Oracle Java 17.0.12
- Thales TCT T-5000 Luna Network HSM model VBD-T7 firmware version 7.11.1
- Thales TCT T-5000 Luna Network HSM model VBD-T7
- Apache Tomcat 9.0.84
- HyperSQL
- PostgreSQL 15.7
- PostgreSQL postgresql_jdbc_driver
- RedHat Enterprise 9.2

- Dhuma
- Windows Server 2019

4 TOE Test Configuration

This section identifies the devices used for testing the TOE and describes the test configuration.



The following components were used to create the test configurations:

- CertAgent RHEL(TOE)
  - IP address: 172.16.20.5 | MAC: 60:7d:09:4c:56:f8
  - Red Hat Enterprise Linux 9.2 (Plow)
  - 13th Gen Intel(R) Core(TM) i7-1370P
  - CertAgent/Dhuma v8.0 patch level 0.2
  - PostgreSQL Version 15.7
  - Oracle Java 17.0.12
  - Postgresql-42.7.3 Driver
- CertAgent Windows(TOE)
  - IP address: 172.16.20.10 | MAC: 60:7d:09:4c:56:f8
  - Windows Server 2019 Standard
  - 13th Gen Intel(R) Core(TM) i7-1370P
  - CertAgent/Dhuma v8.0 patch level 0.2

    - HyperSQL Version 2.7.3
    - Oracle Java 17.0.12
- Thales TCT’s T-5000 Luna Network HSM model VBD-T7

    - IP address: 172.16.20.20
    - firmware version 7.11.1
- Terminal Server(Jump server)
    - 172.16.1.50
    - Microsoft Windows Server 2016 Datacenter
    - RDP
    - XCA Version: 2.1.2
    - FireFox 125.0.3 (64-bit)
    - Wireshark Version 4.2.4
- ATE-ESXi-2:
    - IP:172.16.1.63/DNS:esxi2.leidos.ate
    - Hypervisor: VMware ESXi, 6.7.0, 13006603
    - Hypervisor used to host the following systems:
        - TLS Test Server
            - IP: 172.16.0.25/DNS:tlss.leidos.ate
            - Ubuntu 18.04.5 LTS (Bionic Beaver)
            - OpenSSL 1.1.1
            - proprietary Lab TLS Server and Client test tools
            - Wireshark 2.6.10
            - curl 7.58.0
        - Revocation1.leidos.ate
            - IP: 172.16.1.70
            - 18.04.4 LTS (Bionic Beaver)
            - OpenSSL 1.1.1
            - Wireshark 2.6.10
        - Kali Box
            - IP: 172.16.0.161
            - Kali Linux Release 2019.3
            - Curl 7.65.3
            - Wireshark 3.0.3
            - OpenSSL 1.1.1
            - SSLyze 2.0.6 SSL/TLS Testing Tool