CertAgent

Security Target for Common Criteria Evaluation

Software Version: 7.0

Document Version: 4.1.1

Issue Date: July 11, 2018

Authors: Michael J. Markowitz, Jonathan C. Schulze-Hewett, Pauline Tang

Abstract: This document is the Security Target (ST) for Information Security Corporation's CertAgent, Version 7.0. It provides the basis for the evaluation of CertAgent, a certificate authority, as the Target of Evaluation (TOE). This ST defines assumptions about the environment, threats that the product intends to counter, security objectives, security requirements and the Information Technology (IT) security functions provided by the TOE which meet the requirements.

Legal Notice

Use of CertAgent is subject to the terms of your license agreement with Information Security Corp.

CertAgent is a registered trademark of Information Security Corp. Other product and company names mentioned in this document may be the trademarks of their respective owners.

Document History

Version	Date	Change	Author
1.0.0	2016-01-11	Initial document	Pauling Tang
1.0.1	2016-03-03	Added cryptographic related items in Cryptographic Support (FCS)	Jonathan Schulze-Hewett
1.0.2	2016-04-06	Draft Security Target	Jonathan Schulze-Hewett
1.0.3	2016-06-14	Revised per lab comments	Jonathan Schulze-Hewett
1.0.4	2016-10-24	Revised per NIAP validators comments	Jonathan Schulze-Hewett
2.0.0	2017-03-30	Updated to comply with version 2.0 of the protection profile	Jonathan Schulze-Hewett
2.0.1	2017-07-19	Updated based on lab and NIAP validator comments	Jonathan Schulze-Hewett
2.0.2	2017-09-27	Updated based on lab and NIAP validator comments	Pauline Tang, Jonathan Schulze-Hewett
3.0.0	2017-11-30	Updated to comply with final draft version 2.1 of the protection profile	Jonathan Schulze-Hewett
3.0.1	2017-12-07	Updated to comply with the final 2.1 version of the protection profile	Jonathan Schulze-Hewett
3.0.2	2017-12-12	Added FIA_ENR_EXT.1 and FIA_X509_EXT.3	Jonathan Schulze-Hewett
3.0.3	2018-03-05	Minor tweaks per lab and post testing	Jonathan Schulze-Hewett
3.0.4	2018-03-14	Updated TSS based on internal review	Jonathan Schulze-Hewett, Pauline Tang
3.0.5	2018-03-23	Updated based on lab feedback	Jonathan Schulze-Hewett
3.0.6	2018-03-26	Updated based on lab feedback	Jonathan Schulze-Hewett
3.0.7	2018-03-29	Added section about included TOE documentation	Jonathan Schulze-Hewett
3.0.8	2018-05-04	Updated to address ECR comments	Jonathan Schulze-Hewett
3.0.9	2018-05-22	Updated to address ECR comments	Jonathan Schulze-Hewett
4.0.0	2018-05-29	Updated to address ECR comments	Jonathan Schulze-Hewett
4.1.0	2018-07-05	Updated to address ECR comments	Jonathan Schulze-Hewett
4.1.1	2018-07011	Updated to address ECR comments	Jonathan Schulze-Hewett

Table of Contents

1. Introduction

This section identifies the Security Target (ST) and the target of evaluation (TOE), presents an overview of the TOE and describes the organization of the ST. The TOE is Information Security Corporation's CertAgent certificate authority, a Web-based, X.509-compliant certificate authority (CA) that is intended to be used as the core component of an enterprise public key infrastructure.

This ST is divided into the following Sections:

- Introduction
- TOE Description
- Conformation Claims
- Security Problem Definition
- Extended Components Definition
- Security Functional Requirements
- Security Assurance Requirements
- Security Requirements Rationale
- TOE Summary Specification

1.1 Security Target Reference

Title:	CertAgent Security Target for Common Criteria Evaluation
ST Version:	4.1.1
Status:	Final
Date:	07/11/2018

This ST targets conformance with the following Protection Profile (PP):

- Protection Profile for Certification Authorities, Version 2.1

1.2 TOE Reference

TOE Identification: CertAgent Version 7.0

TOE Developer: Information Security Corporation

Evaluation Sponsor: Information Security Corporation

1.3 TOE Overview

CertAgent, the TOE, is an X.509-compliant certificate authority (CA). It is an easily managed, web-based certificate authority (CA) intended to be used as the core component of an enterprise public key infrastructure (PKI). Designed to meet the needs of a wide variety of organizations, the current release offers enhanced enrollment services (EST), remote administration, integrated certificate and CRL database, and an OCSP responder. It supports an unlimited number of root and intermediate CAs, providing support for as complex a certificate hierarchy as the size of the enterprise warrants. The following diagrams shows the TOE boundary and major components.

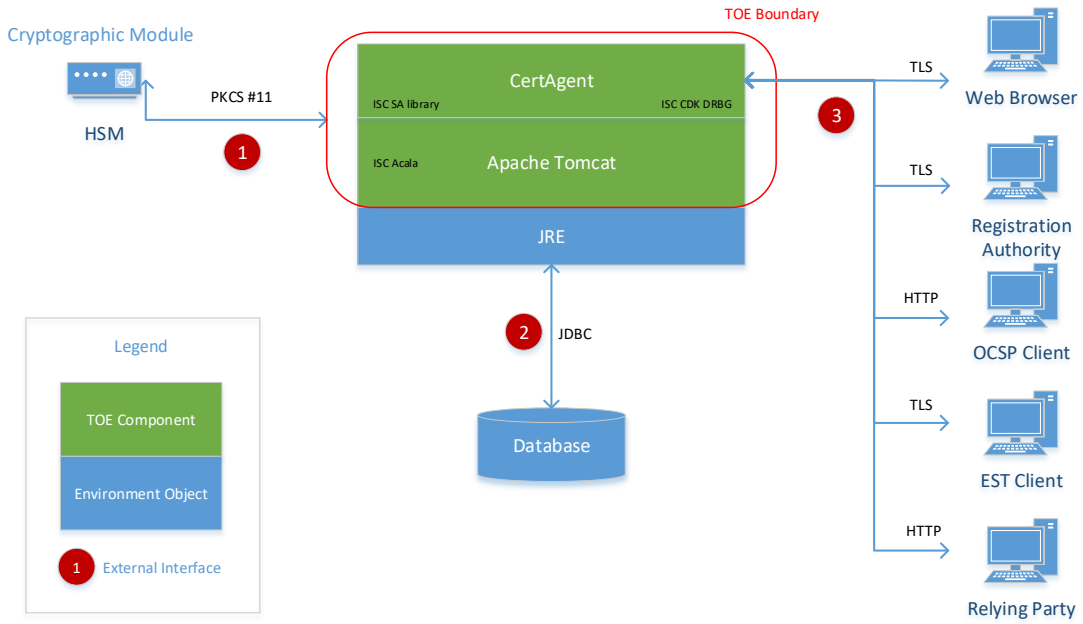

FIGURE 1 TOE BOUNDARY

As Figure 1 shows, CertAgent combined with Apache Tomcat form the Target of Evaluation (TOE). There are 4 high level interfaces that are external to the TOE. The data that traverses these interfaces is protected as shown in the following table.

Interface	Protection
Web	HTTP, HTTPS/TLS
Database	Operating system
OCSP	HTTP, HTTPS/TLS
PKCS#11	Operating system

TABLE 1 TOE INTERFACES

Most CA activities are completed by using a web browser or other tool that connects to the CertAgent web interface. The CA supports seven web-based interfaces using different ports or URLs (Admin Site, CA Account Site, Public Site, RAMI (Registration Authority Management Interface), DBAccess, EST, and OCSP).

- The Admin Site, CA Account Site, DBAccess, and Registration Authority (RAMI) channels require valid identification and authentication credentials in the form of certificates. This channel is secured using client authenticated HTTPS/TLS.
- The Public Site channel is secured using HTTPS/TLS and HTTP. All pages except the CA Information page are HTTPS/TLS protected. The CA information page, used by relying parties to obtain CRLs, issuer certificates, and CA version information, is available without security over HTTP. All pages except the self-service revocation page are unauthenticated. The self-service revocation page requires valid identification and authentication credentials in the form of certificates.
- The EST channel is secured using HTTPS/TLS. Connections are authenticated with either certificates or a subscriber name and password.
- The OCSP interface is available without security over HTTP or secured using HTTPS/TLS. All access is unauthenticated.

Configuration data (including ACLs), most audit logs, certificates, and CRLs are stored in tables in a single database. In the evaluated configurations, the database is either HyperSQL or PostgreSQL and is hosted on the same physical system as the TOE. The connection to the database is not secured, but is authenticated. Sensitive data stored in the database is encrypted before it is sent to the database for storage. The environmental JRE's JDBC API is used to communicate with the database using a database vendor supplied JDBC driver.

CertAgent has an option to connect to LDAP servers to push certificate and CRLs as they are issued. Since certificates and CRLs are public information this connection may be unsecure or secure and may or may not be authenticated. There is no LDAP server in the evaluated configuration, LDAP publishing was not evaluated, and LDAP publishing is disabled when CertAgent is configured with strict NIAP compliance settings.

Private keys used for issuing certificates, issuing CRLs, authenticating the TLS server, and signing OCSP responses reside in the environmental PKCS#11 Cryptographic Module. In the evaluated configuration, the PKCS#11 Cryptographic Module, is Gemalto's SafeNet USB HSM, but any PKCS#11 Cryptographic Module that is at least FIPS 140-2 Level 2 validated, provides hardware security of keys, includes a PKCS#11 library, supports the required algorithms (in particular a 256-bit DRBG with 256-bits of entropy input), and provides a backup capability, is considered equivalent. PKCS#11 is a C API exported from a shared library (a DLL or .so depending on platform that is provided by the HSM vendor). The TOE loads this library on startup and calls functions in it as it would any other local library. Data traversing this interface is protected by the environmental operating system in which processes are segregated in to their own process space and are logically separated from all other processes by the operating system and underlying hardware.

The following figure, taken from the CA PP, shows the reference architecture for a certification authority product:

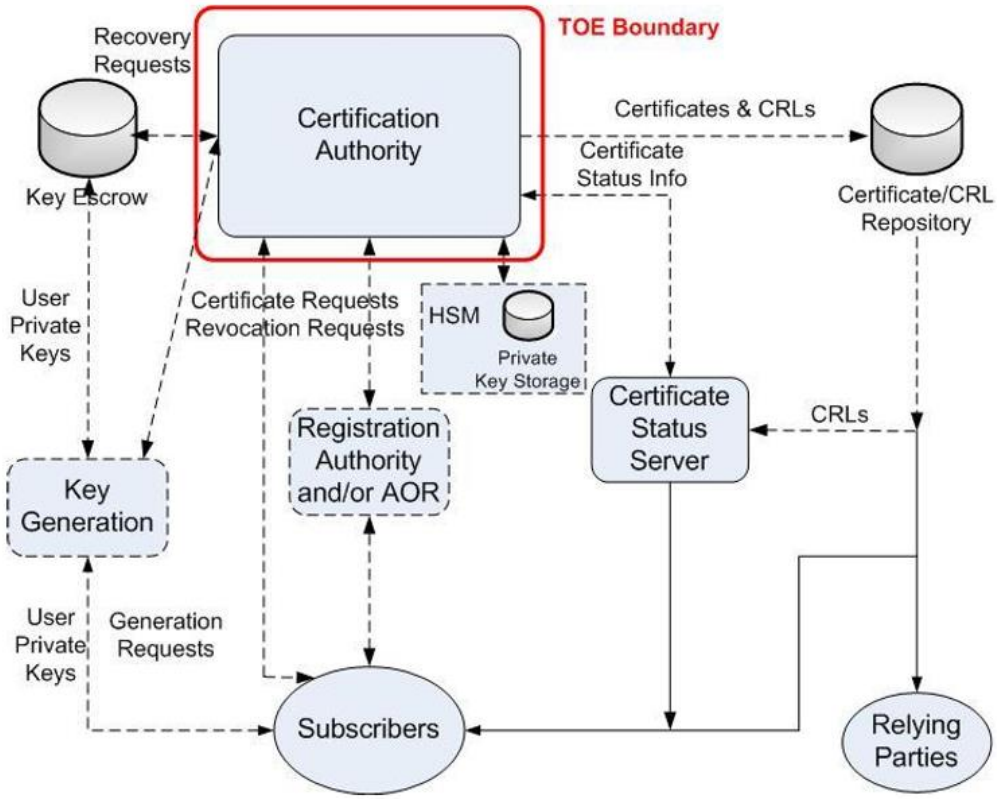

FIGURE 2 CA PP CONTEXT FOR THE TOE

In general, the following correspondence can be seen between Figure 2 above and the TOE diagram shown in Figure 1:

- Certification Authority – the TOE
- Certificate Status Server – the TOE's built-in OCSP responder
- Certificate/CRL Repository – the Database
- HSM – the PKCS#11 Cryptographic Module
- Subscribers – EST client, Web browser
- Relying Parties – the Relying Party, Web browser
- Registration Authority – Registration Authority

There are some minor differences between the two diagrams that do not impact the ability of the TOE to claim conformance with the CA PP. They are as follows:

- The TOE does not interface with a key escrow system
- The TOE does not interface with a user key generation system
- The TOE contains a certificate status server

1.4 TOE Type

The TOE type for CertAgent is Certification Authority (CA). The TOE is a software package installed on a general computing platform that is used to issue and manage public-key certificates and provide certificate status information.

2. TOE Description

This section contains a description of the TOE in its evaluated configurations and includes the physical and logical boundaries of the TOE.

2.1 Evaluated Components of the TOE

The following table describes the major TOE components in the evaluated configuration:

Component	Definition
CertAgent	The certification authority software web application.
Application Server (web container, Java Servlet	Apache Tomcat application server that hosts the CertAgent web app and the web interface. In the evaluated configuration Apache Tomcat is configured to use the ISC CDK and the PKCS#11 Cryptographic Module for cryptographic operations. Apache Tomcat is

Container)	part of the TOE's installation package and is installed when CertAgent is installed.
ISC CDK	The cryptographic module in the TOE.

TABLE 2 EVALUATED COMPONENTS OF THE TOE

The TOE uses ISC's Cryptographic Development Kit version 8.0 (ISC CDK) for many of the cryptographic operations described later in this document. The ISC CDK is a shared library to which the CertAgent web application and Apache Tomcat are linked dynamically. The ISC CDK is similar to OpenSSL®, Crypto++, Cryptlib®, and other cryptographic libraries implemented as software libraries.

2.2 Components and Applications Required in the Operational Environment

The following table lists components and applications in the environment that the TOE relies upon in order to function properly:

Component	Definition
Database	Stores configuration data (including ACLs), audit logs, certificates, and CRLs for the TOE. In the evaluated configurations, the TOE uses PostgreSQL or HyperSQL for its database. HyperSQL is included in the installation package and, optionally, installed when the TOE is installed. PostgreSQL must be obtained and installed before installing the TOE. PostgreSQL may be downloaded from https://jdbc.postgresql.org/download.html. Either database must be installed on the same host as the TOE.
Java JRE	Runs the CertAgent and application servers. It is included in the installation package and installed by TOE installer.
JCE Unlimited Strength Jurisdiction Policy	Enables unlimited strength cryptography in Java which is required to support the AES-256 ciphersuite. It is included in the installation package and installed by the TOE installer.
PKCS#11 Cryptographic Module	Stores the private keys used to sign certificates, issue CRLs, create OCSP responses, and authenticate the server to clients via TLS/HTTPS. In the evaluated configuration, the PKCS#11 Cryptographic Module is Gemalto's USB HSM and it must be directly connected to the same host as the TOE via USB (it is not a network HSM).
Server	Physical system on which the CertAgent software is installed. The physical system is either: A 64-bit Microsoft Windows Server 2012 R2 OS and JRE. A CentOS 6.7 x86_64 OS, rng-tools, and JRE.
Web Browser	The interface that is used to access the CertAgent web user interface. In the evaluated configuration the web interface is accessed via Firefox. Firefox can be downloaded from https://www.mozilla.org.

TABLE 3 COMPONENTS OF THE OPERATIONAL ENVIRONMENT

The TOE does not include the operating systems or hardware of the systems on which it is installed. It also does not include the third-party software required for the TOE to run. Table 4 and Table 5 list the software components required by the TOE in the evaluated configurations. The Operational

Environment components should be maintained such that the latest security fixes for each component are installed in a timely manner.

Component	Requirement
Server OS	Windows Server 2012 R2
OS Type	64-bit
Database	HyperSQL Version 2.4
Java JRE	Oracle Java JRE 8 1.8
PKCS#11 Cryptographic Module	Gemalto SafeNet USB HSM

TABLE 4 OPERATIONAL ENVIRONMENT SOFTWARE REQUIREMENTS (WINDOWS)

Component	Requirement
Server OS	CentOS 6.7 w/rng-tools package
OS Type	x86_64 (64-bit)
Database	PostgreSQL Version 9.4
Java JRE	Oracle Java JRE 8 1.8
PKCS#11 Cryptographic Module	Gemalto SafeNet USB HSM

TABLE 5 OPERATIONAL ENVIRONMENT SOFTWARE REQUIREMENTS (LINUX)

The TOE requires that the environmental Operating System maintains the Operational Environment (OE) Administrator role and the Operational Environment (OE) Auditor role. Members of the OE Administrator role shall be granted root/administrator permissions in the Operating System. On Linux, members of the OE Auditor role shall be placed in the ca_audit group created per guidance and must not be granted root/sudo permissions. On Windows, members of the OE Auditor role are normal users and must not be granted administrator permissions. Only members of OE Administrators or OE Auditors may access the environmental Operating System.

In addition to the server requirements, a web browser is required for any system used to remotely access the TOE (or to access certain functionality when logged into the Operating System in which the TOE is running). In the evaluated configuration, the TOE was tested using Firefox ESR version 52 and the compatibility of other browsers was not assessed.

2.2.1 Excluded from the Evaluated Configuration

The following list contains Operational Environment software and hardware supported by the TOE but not included or tested as part of this evaluation:

Operating Systems

- Oracle Enterprise Linux 6 and 7
- Red Hat Enterprise Linux 6 and 7
- SuSE Linux Enterprise Server 10 and 11
- Ubuntu 14 and 16
- Windows 7
- Windows 10
- Windows Server 2008
- Windows Server 2008 R2
- Windows Server 2012
- Windows Server 2016

Databases

- Oracle 11g, 12c

PKCS#11 Cryptographic Modules

- Thales nShield Connect Series
- Thales nShield Solo Series
- Thales nShield Edge Series
- Gemalto Safenet Network HSM
- Gemalto Safenet PCIe HSM
- ISC Acala (HSM emulator)

Application Servers

- Apache Tomcat 7.0.x, 9.0.x
- Oracle WebLogic 11g, 12c

Java Platform

- Oracle JDK 7, 8, 9
- Oracle JRE 7, 9

Note that while both of the TOE supported OS platforms support remote access to the local console (*e.g.*, SSH), this capability is excluded from the evaluated configuration and is removed or disabled during initial installation and configuration of the TOE.

2.3 TOE Physical Boundary

The physical boundary of the TOE includes the CertAgent software, Apache Tomcat, and the ISC CDK installed on a general purpose computer running a supported operating system. The TOE does not include the hardware, database, HSM, or the operating systems of the computers on which the TOE is installed. It also does not include third-party software required for the TOE to run as detailed in Section 2.2.

The following external IT entities can connect to the TOE using REST APIs:

- DBAccess clients such as an external audit server, or external user management server
- RAMI clients such as a registration authority

2.3.1 TOE Installation Package and Configuration Management

TOE installation packages and TOE update packages are delivered in a zipped archive via a download link. A valid serial number is required to download the package. Licensed customers receive a digitally signed email with a download and serial number. Both Windows and CentOS installation packages contain the TOE components (CertAgent, Apache Tomcat, and ISC CDK), required software (HyperSQL database, Oracle JRE, and JCE Unlimited Strength Jurisdiction Policy), and documentation. An installation executable and script is used to install the TOE. An update package is verified by the TOE, unpacked, and executed to update the TOE from one version to another. The files changed and/or replaced by the update vary based on the complexity of the update.

The TOE's source code is maintained under version control (Subversion and Microsoft Visual SourceSafe) and each release or patch is checked in and labeled prior to release. External releases are labeled by version in dot notation of major.minor.patch (e.g., 7.0.6 represents major version 7, minor version 0, patch level 6). Internal releases are labeled major.minor.patch.qa (e.g., 7.0.6.QA1 represents major version 7, minor version 0, patch level 6, quality assurance cycle 1). Both version control systems maintain the history and integrity of the TOE's source code. Each developer has an individual account in the version control system so that changes are attributable and tracked to the user. A defect tracking system, Bugzilla, is used to manage defects and feature requests in releases of the TOE. In combination, the source code control systems and the defect tracking system, allow the developers to determine exactly what changed in any given version or update of the software.

2.3.2 TOE Documentation

The TOE includes the following guidance documents:

- CertAgent Administrator Guide, version 7.0, July 5, 2018

- CertAgent Installation, Configuration and Management Guide, version 7.0, July 5, 2018
- CertAgent Certificate Authority Guide, version 7.0, July 5, 2018
- CertAgent Public Site Guide, version 7.0, July 15, 2018
- CertAgent Guidance for Common Criteria Evaluation, version 2.3.0, July 7, 2018
- CertAgent 7.0.6 Release Notes, June 26, 2018

2.4 TOE Logical Boundary

The TSF is comprised of several security features. Each security feature identified belongs to one of several general categories:

1. Security Audit
2. Communication
3. Cryptographic Support
4. User Data Protection
5. Identification and Authentication
6. Security Management
7. Protection of the TSF
8. TOE Access
9. Trusted Path/Channels

2.4.1 Security Audit

The TOE generates audit records of administrator, user, and its own activities. Audit data includes date, time, event type, subject identity, and other data as required. Most audit data are written to the database. Audit records indicating a database failure are stored in a local text file as the database is inaccessible. The TOE allows an external IT entity to access TOE audit records in the database by polling the TOE using the DBAccess REST API.

2.4.2 Communications

The TOE relies on the TLS/HyperText Transfer Protocol Secure (HTTPS) when transmitting sensitive data to and from applicable endpoints.

Certificate requests, certificates, CRLs, and OCSP responses are formed and verified according to RFC 5280, RFC 6960, and RFC 7030. Certificate validation is performed by the TOE.

Sensitive data that needs to be recovered (PKCS#11 Cryptographic Module PINs and other authentication passwords) are encrypted using CMS, in conformance with RFC 5652, and then stored in the database. Sensitive data that does not need to be recovered, EST passwords, are not stored directly, but a check value is created, using PBKDF2/SHA-256, and stored.

2.4.3 Cryptographic Support

Cryptographic support is provided by two components.

- ISC's Cryptographic Development Kit (ISC CDK)
- PKCS#11 Cryptographic Module

ISC CDK

The ISC CDK is within the TOE's boundary and is used by the TOE:

- to generate the initial set of authentication credentials (certificates and associated private keys) during installation,
- to generate symmetric keys, wrap them with public keys, and use them to encrypt sensitive data using the CMS format,
- to hash the "to be signed" message bodies of certificates, CRLs, and OCSP responses
- to validate signatures on certificates, CRLs, and requests, and
- to provide communication protection when clients establish TLS/HTTPS connections to the Administrative, CA, Public, EST, OCSP, DBAccess, or RAMI interfaces. Note: Cryptographic functions involving the TLS server private key are provided by the environmental PKCS#11 Cryptographic Module.

The ISC CDK is FIPS 140-2 Level 1 validated on 64-bit Windows 10 and 64-bit CentOS 6.7. Since the FIPS validation does not cover Windows Server 2012R2, the necessary algorithm tests have been performed for that platform through the NIST Cryptographic Algorithm Validation Program (CAVP) and algorithm certificates obtained.

PKCS#11 Cryptographic Module

The PKCS#11 Cryptographic Module is used by the TOE:

- to generate, store, and provide cryptographic operations (unwrapping DEKs) involving the private key for the "System" credential (certificate and private key),
- to generate, store, and provide cryptographic operations (digital signatures) involving the private key for all issuer credentials (certificates and private keys), and

- to generate, store, and provide cryptographic operations (digital signature) involving the private key for the TLS/HTTPS server credential (certificate and private key).

The PKCS#11 Cryptographic Module securely stores the high value certificate authority keys and uses them to perform the signature operations that define a certificate authority. The PKCS#11 Cryptographic Module also securely stores the TLS/HTTPS server key and provides cryptographic services involving that key. PKCS#11 Cryptographic Modules often require a PIN or other authentication when the application using them starts and the TOE provides mechanisms for injecting this information during its startup procedures.

In the evaluated configuration, the PKCS#11Cryptographic Module is Gemalto's SafeNet USB HSM.

2.4.4 User Data Protection

The TOE supports the creation of multiple certificate profiles by CA Administrators. Each profile is customizable by a CA Administrator, and includes a certificate-based ACL of CA Operations Staff members allowed to issue or revoke certificates using the profile. Certificate requests are assigned a unique identifier upon submission that is used to link the request to the issued certificate.

The TOE provides relying parties two methods to check the status of a certificate:

- X.509v2 CRLs
- OCSP

CRLs can be issued manually, on a schedule, or when a certificate is revoked for a set of configurable reason codes.

2.4.5 Identification and Authentication

The TOE uses two different identification and authentication methods, described in Sections 2.4.5.1 and 2.4.5.2, depending on the role and action being performed. Most TOE activities, and all activities involving the issuance or revocation of certificates, require certificate-based authentication.

PKCS#11 Cryptographic Modules support a variety of authentication options including: passwords, smart cards, PED devices, and fingerprints. In all cases, someone must enable the PKCS#11 Cryptographic Module as part of the initialization of the TOE. This step is performed locally on the system during startup of the TOE.

Access to the TOE's local console is controlled by the underlying environmental operating system which performs the required identification and authentication when an administrator logs on.

2.4.5.1 EST Authentication

EST authentication supports either certificate-based authentication or subscriber name/password authentication (over HTTPS) in cases where the requester does not have a valid certificate.

For subscriber name/password authentication (over HTTPS), privileged users in the CA Operation Staff role create and manage the subscriber name/password associations.

2.4.5.2 Certificate-based Identification and Authentication

Access to the Admin Site, CA Account Site, DBAccess API, or RAMI API requires certificate-based client authentication using HTTPS. The functions available depend on the ACL and permissions that are assigned to the certificate used to authenticate.

The portion of the Public Site allowing self-service revocation by subscribers also requires certificate-based client authentication using HTTPS.

2.4.6 Security Management

The TOE is managed by authorized administrators using a web user interface and the local console as needed. All certificate issuance related administrative actions are performed via the web interface. The TOE supports three (3) roles (Administrator, Auditor, and CA Operations Staff) each of which consists of an access control list (ACL) of one or more X.509 certificates and one or more permissions (issue, revoke, RAMI, etc.).

Only users who hold an administrator role in the TOE are allowed to have administrator privileges on the physical system on which the TOE is installed. They can:

- inject the PKCS#11 Cryptographic Module PIN to unlock the “System” credential’s private key,
- start/stop the TOE and the Database,
- run the CACLI program (allows the scripting of the creation of a root or issuer, trust anchor management, ACL management),
- run the Report Generator Program, or
- run the update tool (to check for updates or apply updates to the system).

2.4.7 Protection of the TSF

The TOE encrypts any sensitive information, before it is sent to the environmental database, using the asymmetric “System” credential’s public key and the CMS format. These encrypted symmetric keys are the only symmetric keys that are persisted by the TOE. When the information is needed later, the encrypted data is retrieved from the database, and the TOE uses the “System” credential’s private key, via the PKCS#11 Cryptographic Module’s PKCS#11 API, to unwrap the symmetric key.

The TOE maintains the password of PKCS#11 Cryptographic Module storing the “System” credential in memory until it exits. The TOE does not store, or directly use, any private keys (they are stored and protected by the PKCS#11 Cryptographic Module which performs operations with those keys at the TOE’s request). When the TOE shuts down all sensitive data in memory is cleared.

2.4.8 TOE Access

The TOE's Admin Site and CA Account Site display a warning banner prior to allowing any administrative actions to be performed. The TOE's web interface will terminate sessions when they time out or when an authenticated user clicks the logout link in the navigation pane.

2.4.9 Trusted Path/Channels

The TOE provides a trusted path/channel for secure communication between itself and external IT entities such as a registration authority (RA), audit server, or similar entities which are permitted to connect to the TOE, over client authenticated HTTPS/TLS. Privileged users accessing the TOE's web interfaces also use a trusted path established and secured with client authenticated HTTPS/TLS. Subscribers with existing, valid certificates, also use a trusted path, established and secured with client authenticated HTTPS/TLS, to perform certificate renewal, via EST, or self-management, via the TOE's Public Site web interface. Subscribers, and other non-privileged users, are permitted to connect to the TOE's Public Site with unauthenticated HTTPS/TLS. Relying parties are permitted to connect to parts of the TOE's Public Site, with either unauthenticated HTTPS/TLS or unprotected HTTP, to obtain certificate status or other information required to validate certificates issued by the TOE.

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 Operational Environment's 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.

3. Conformance Claims

3.1 CC Version

This ST is compliant with Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 5 April 2017.

3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) are Part 2 extended to include all applicable NIAP and International interpretations through 11 July 2018.

3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) are Part 3 conformant to include all applicable NIAP and International interpretations through 11 July 2018.

Note that this evaluation also includes evaluation assurance activities that are defined in the claimed Protection Profile that has augmented the CEM and are not considered to be alterations to Part 3.

3.4 PP Claims

This ST claims compliance to the following Protection Profile:

- Protection Profile for Certification Authorities, version 2.1 [PP_CA_V2.1]

3.5 Applicable Technical Decisions

The following technical decisions are applicable to the PP_CA_V2.1:

- TD0294
- TD0287
- TD0286
- TD0278
- TD0276

3.6 Package Claims

This ST claims compliance to a Protection Profile. The are no package claims in the PP or ST.

3.7 Conformance Claim Rationale

The PP states “A CA system is an entity that issues and manages public-key certificates.” The TOE is a software application that issues and manages public-key certificates. As such, it is consistent with the definition of a certification authority as stated in the PP. Therefore, the conformance claim is appropriate.

4. Security Problem Definition

4.1 Threats

This section identifies the threats against the TOE. These threats have been taken from the PP.

Threat Name	Threat Definition
T.PRIVILEGED_USER_ERROR	A privileged user or non-person entity (NPE) improperly exercises or adversely affects the TOE, resulting in unauthorized services, ineffective security mechanisms, or unintended circumvention of security mechanisms.
T.TSF_FAILURE	Security mechanisms of the TOE may fail, leading to a compromise of the TSF.
T.UNAUTHENTICATED_TRANSACTIONS	Relying parties within an information system depend on the TOE to accurately bind subjects to their credentials for use in authenticating and providing privacy for transactions. Without the proper binding provided by the TOE, relying parties cannot ensure adequate access controls on sensitive information, ensure transactional integrity, ensure proper accountability, and/or enforce non-repudiation.
T.UNAUTHORIZED_ACCESS	A malicious user, process, or external IT entity intentionally circumvents TOE security mechanisms.
T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.UNDETECTED_ACTIONS	Remote users or external IT entities may take actions that adversely affect the security of the TOE.
T.USER_DATA_REUSE	A malicious user, process, or external IT entity may gain access to user data that is not cleared when resources are reallocated.
T.WEAK_CRYPTO	A weak hash or signature scheme may be compromised by an attacker and used to apply integrity checks to malicious content so that it appears legitimate.

TABLE 6 TOE THREATS

4.2 Assumptions

The specific conditions listed are assumed to exist in the TOE's Operational Environment and these assumptions have been taken from the PP.

Assumption Name	Assumption Definition
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

TABLE 7 TOE ASSUMPTIONS

4.3 Organizational Security Policies

This section identifies the organizational security policies to be implemented by an organization that deploys the TOE. These policies have been taken from the CA PP.

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

TABLE 8 TOE ORGANIZATIONAL SECURITY POLICIES

4.4 Security Objectives

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

4.4.1 Security Objectives for the TOE

This section identifies the security objects of the TOE. These objectives have been taken from the CA PP. A subset of the optional security objects has been included based on the set of optional SFRs that are claimed by the TSF.

Objective	Objective Definition
O.AUDIT_LOSS_RESPONSE	The TOE will respond to possible loss of audit records when audit trail storage is full or nearly full by restricting auditable events.
O.AUDIT_PROTECTION	The TOE will protect audit records against unauthorized access, modification, or deletion to ensure accountability of user actions.
O.CERTIFICATES	The TSF must ensure that certificates, certificate revocation lists, and certificate status information are valid.
O.CONFIGURATION_MANAGEMENT	The TOE will conduct configuration management to assure identification of system connectivity (software, hardware, and firmware), and components (software, hardware, and firmware), auditing of configuration data, and controlling changes to configuration items.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.INTEGRITY_PROTECTION	The TOE will provide appropriate integrity protection for user data and software.
O.NON_REPUDIATION	The TOE will prevent a subscriber from avoiding accountability for sending a message by providing evidence that the subscriber sent the message; and control communications from unknown source.
O.PROTECTED_COMMUNICATIONS	The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities. The TOE will protect data assets when they are being transmitted to and from the TOE, including through intervening untrusted components.
O.RECOVERY	The TOE will have the capability to store and recover to a previous state at the direction of the administrator (e.g., provide support for archival and recovery capabilities).
O.RESIDUAL_INFORMATION_CLEARING	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.SESSION_LOCK	The TOE with provide mechanisms that mitigate the risk of unattended

	sessions being hijacked.
O.SYSTEM_MONITORING	The TOE will provide the capability to generate audit data and send those data to an external IT entity. The TOE will record in audit records: date and time of action and the entity responsible for the action.
O.TOE_ADMINISTRATION	The TOE will provide mechanisms to ensure that only privileged users are able to log in and configure the TOE, and provide protections for logged-in users. The TOE will ensure that administrative responsibilities are separated across different roles in order to mitigate the impact of improper administrative activities or unauthorized administrative access.
O.TSF_SELF_TEST	The TOE will provide integrity protection to detect modifications to firmware, software, and archived data.
O.VERIFIABLE_UPDATES	The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source

TABLE 9 TOE OBJECTIVES

4.4.2 Security Objectives for the Operational Environment

This section identifies the security objectives of the environment into which the TOE is expected to be deployed. These objectives have been taken from the CA PP. A subset of the optional environmental objectives has been included based on the set of optional SFRs that are not claimed by the TSF.

Objective	Objective Definition
OE.AUDIT_GENERATION	The Operational Environment provides a mechanism for the generation of portions of the audit data.
OE.CERT_REPOSITORY	The Operational Environment provides a certificate repository for storage of certificates (and optionally CRLs) issued by the TSF.
OE.AUDIT_RETENTION	The Operational Environment provides mechanisms for retention of audit records for both normal and extended retention periods.
OE.AUDIT_REVIEW	The Operational Environment provides a mechanism for the review of specified audit data.
OE.AUDIT_STORAGE	The Operational Environment provides a mechanism for the storage of specified audit data.
OE.CRYPTOGRAPHY	The Operational Environment provides cryptographic services that can be invoked by the TSF in order to perform security functionality.
OE.KEY_ARCHIVAL	The Operational Environment provides the ability to use split knowledge procedures to enforce two-party control to export keys necessary to resume CA functionality if the TSF should fail.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.PUBLIC_KEY_PROTECTION	The Operational Environment provides protection for specified public keys associated with CA functions.
OE.SESSION_PROTECTION_LOCAL	The Operational Environment provides the ability to lock or terminate local administrative sessions.

OE.TOE_ADMINISTRATION	The Operational Environment provides specified management capabilities required for the overall operation of a Certificate Authority, and the ability to restrict access to a subset of the capabilities as specified in the ST.
OE.TRUSTED_ADMIN	The administrator of the TOE is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.
OE.TRUSTED_PLATFORM	The operating system no which the TOE has been installed is securely configure, regularly patched, and not subject to unauthorized access.

TABLE 10 TOE OPERATIONAL ENVIRONMENT OBJECTIVES

4.5 Security Problem Definition Rationale

The assumptions, threats, Organizational Security Policies (OSPs), and objectives that are defined in this ST represent the assumptions, threats, OSPs, and objectives that are specified in the Protection Profile to which the TOE claims conformance. The associated mappings of assumptions to environmental objectives, SFRs to TOE objectives, and OSPs and objectives to threats are specified in Table 11.

SPD Element	Objective	Requirements
A.NO_GENERAL_PURPOSE **It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.**	**OE.NO_GENERAL_PURPOSE** There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.	N/A
A.PHYSICAL **Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.**	**OE.PHYSICAL** Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.	N/A
A.TRUSTED_ADMIN **TOE Administrators are assumed to follow and apply all administrator guidance in a trusted manner.**	**OE.TRUSTED_ADMIN** The administrator of the TOE is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.	N/A
T.PRIVILEGED_USER_ERROR **A privileged user or non-person entity (NPE) improperly exercises or adversely affects the TOE, resulting in unauthorized**	**O.AUDIT_LOSS_RESPONSE** The TOE will respond to possible loss of audit records when audit trail storage is full or nearly full by restricting auditable	FAU_ADP_EXT.1, FAU_STG.4

services, ineffective security mechanisms, or unintended circumvention of security mechanisms.	events.	
	O.AUDIT_PROTECTION The TOE will protect audit records against unauthorized access, modification, or deletion to ensure accountability of user actions.	FAU_ADP_EXT.1
	O.TOE_ADMINISTRATION The TOE will provide mechanisms to ensure that only privileged users are able to log in and configure the TOE, and provide protections for logged-in users. The TOE will ensure that administrative responsibilities are separated across different roles in order to mitigate the impact of improper administrative activities or unauthorized administrative access.	FIA_UAU_EXT.1, FIA_UIA_EXT.1, FMT_MOF.1(1), FMT_MOF.1(2), FMT_MOF.1(3), FMT_MOF.1(4), FMT_MOF.1(5), FMT_MTD.1, FMT_SMF.1, FMT_SMR.2, FTA_SSL.3, FTA_SSL.4
	OE.AUDIT_GENERATION The Operational Environment provides a mechanism for the generation of portions of the audit data.	N/A
	OE.AUDIT_STORAGE The Operational Environment provides a mechanism for the storage of specified audit data.	N/A
	OE.AUDIT_REVIEW The Operational Environment provides a mechanism for the review of specified audit data.	N/A
	OE.AUDIT_RETENTION The Operational Environment provides mechanisms for retention of audit records for both normal and extended retention periods.	N/A
	OE.SESSION_PROTECTION_LOCAL The Operational Environment provides the ability to lock or terminate local administrative sessions.	N/A
	OE.TOE_ADMINISTRATION The Operational Environment provides specified management capabilities	N/A

	required for the overall operation of a Certificate Authority, and the ability to restrict access to a subset of the capabilities as specified in the ST	
T.TSF_FAILURE **Security mechanisms of the TOE may fail, leading to a compromise of the TSF.**	**O.TSF_SELF_TEST** The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly. The TOE will provide integrity protection to detect modifications to firmware, software, and archived data.	FPT_TST_EXT.2
	OE.TRUSTED_PLATFORM The operating system on which the TOE has been installed is securely configured, regularly patched, and not subject to unauthorized access.	N/A
T.UNAUTHENTICATED_TRANSACTIONS **Relying parties within an information system depend on the TOE to accurately bind subjects to their credentials for use in authenticating and providing privacy for transactions. Without the proper binding provided by the TOE, relying parties cannot ensure adequate access controls on sensitive information, ensure transactional integrity, ensure proper accountability, and/or enforce non-repudiation.**	**O.CERTIFICATES** The TSF must ensure that certificates, certificate revocation lists, and certificate status information are valid.	FDP_CER_EXT.1, FDP_CER_EXT.2, FDP_CER_EXT.3, FDP_CRL_EXT.1, FDP_CSI_EXT.1, FDP_OCSPG_EXT.1, FDP_STG_EXT.1, FIA_ESTS_EXT.1, FIA_X509_EXT.1, FIA_X509_EXT.2
	O.CONFIGURATION_MANAGEMENT The TOE will conduct configuration management to assure identification of system connectivity (software, hardware, and firmware), and components (software, hardware, and firmware), auditing of configuration data, and controlling changes to configuration items.	FDP_CER_EXT.1,FDP_CRL_EXT.1, FDP_OCSPG_EXT.1, FMT_MOF.1(1), FMT_MOF.1(2), FMT_MOF.1(3), FMT_MOF.1(4), FMT_MOF.1(5), FMT_MTD.1
	O.INTEGRITY_PROTECTION The TOE will provide appropriate integrity protection for TSF data and software and any user data stored by the TOE.	FCS_CDP_EXT.1, FCS_CKM_EXT.5, FPT_TST_EXT.2
	O.NON_REPUDIATION The TOE will prevent a subscriber from avoiding accountability for sending a message by providing evidence that the subscriber sent the message; and control communications from unknown source.	FCO_NRO_EXT.2
	OE.TOE_ADMINISTRATION The Operational Environment provides specified management capabilities	N/A

	required for the overall operation of a Certificate Authority, and the ability to restrict access to a subset of the capabilities as specified in the ST	
T.UNAUTHORIZED_ACCESS **A malicious user, process, or external IT entity intentionally circumvents TOE security mechanisms.**	**O.PROTECTED_COMMUNICATIONS** The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities. The TOE will protect data assets when they are being transmitted to and from the TOE, including through intervening untrusted components.	FCS_CDP_EXT.1, FCS_CKM.1, FCS_CKM.2, FCS_CKM_EXT.1(1), FCS_CKM_EXT.1(2), FCS_CKM_EXT.4, 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_STG_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, FPT_KST_EXT.1, FPT_KST_EXT.2, FPT_SKP_EXT.1, FTP_ITC.1, FTP_TRP.1
	O.TOE_ADMINISTRATION The TOE will provide mechanisms to ensure that only privileged users are able to log in and configure the TOE, and provide protections for logged-in users. The TOE will ensure that administrative responsibilities are separated across different roles in order to mitigate the impact of improper administrative activities or unauthorized administrative access.	FIA_UAU_EXT.1, FIA_UIA_EXT.1, FMT_MOF.1(1), FMT_MOF.1(2), FMT_MOF.1(3), FMT_MOF.1(4), FMT_MOF.1(5), FMT_MTD.1, FMT_SMF.1, FMT_SMR.2, FTA_SSL.3, FTA_SSL.4
	OE.CRYPTOGRAPHY The Operational Environment provides cryptographic services that can be invoked by the TSF in order to perform security functionality.	N/A
	OE.KEY_ARCHIVAL The Operational Environment provides the ability to use split knowledge procedures to enforce two-party control to export keys necessary to resume CA functionality if the TSF should fail.	N/A
	OE.SESSION_PROTECTION_LOCAL The Operational Environment provides the ability to lock or terminate local administrative sessions.	N/A
	OE.TOE_ADMINISTRATION The Operational Environment provides specified management capabilities required for the overall operation of a	N/A

	Certificate Authority, and the ability to restrict access to a subset of the capabilities as specified in the ST	
T.UNAUTHORIZED_UPDATE **A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.**	**O.VERIFIABLE_UPDATES** The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and from a trusted source.	FCS_CDP_EXT.1, FCS_COP.1(3), FIA_X509_EXT.2, FPT_TUD_EXT.1
T.UNDETECTED_ACTIONS **Remote users or external IT entities may take actions that adversely affect the security of the TOE.**	**O.AUDIT_LOSS_RESPONSE** The TOE will respond to possible loss of audit records when audit trail storage is full or nearly full by restricting auditable events.	FAU_ADP_EXT.1, FAU_STG.4
	O.AUDIT_PROTECTION The TOE will protect audit records against unauthorized access, modification, or deletion to ensure accountability of user actions.	FAU_ADP_EXT.1,
	O.SYSTEM_MONITORING The TOE will provide the capability to generate audit data and send those data to an external IT entity. The TOE will record in audit records: date and time of action and the entity responsible for the action.	FAU_ADP_EXT.1, FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.3, FAU_GCR_EXT.1, FAU_SCR_EXT.1, FAU_SEL.1, FAU_STG_EXT.1, FIA_UIA_EXT.1, FPT_STM.1
	OE.AUDIT_GENERATION The Operational Environment provides a mechanism for the generation of portions of the audit data.	N/A
	OE.AUDIT_STORAGE The Operational Environment provides a mechanism for the storage of specified audit data.	N/A
	OE.AUDIT_REVIEW The Operational Environment provides a mechanism for the review of specified audit data.	N/A
	OE.AUDIT_RETENTION The Operational Environment provides mechanisms for retention of audit records for both normal and extended retention	N/A

	periods.	
	OE.CERT_REPOSITORY The Operational Environment provides a certificate repository for storage of certificates (and optionally CRLs) issued by the TSF.	N/A
T.USER_DATA_REUSE **A malicious user, process, or external IT entity may gain access to user data that is not cleared when resources are reallocated.**	**O.RESIDUAL_INFORMATION_CLEARING** The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.	FDP_RIP.1
T.WEAK_CRYPTO **A weak hash or signature scheme may be compromised by an attacker and used to apply integrity checks to malicious content so that it appears legitimate.**	**O.PROTECTED_COMMUNICATIONS** The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities. The TOE will protect data assets when they are being transmitted to and from the TOE, including through intervening untrusted components.	FCS_CDP_EXT.1, FCS_CKM.1, FCS_CKM.2, FCS_CKM_EXT.1(1), FCS_CKM_EXT.1(2), FCS_CKM_EXT.4, 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_STG_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, FPT_KST_EXT.1, FPT_KST_EXT.2, FPT_SKP_EXT.1, FTP_ITC.1, FTP_TRP.1
	O.VERIFIABLE_UPDATES The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and from a trusted source.	FCS_CDP_EXT.1, FCS_COP.1(3), FIA_X509_EXT.2, FPT_TUD_EXT.1
	OE.CRYPTOGRAPHY The Operational Environment provides cryptographic services that can be invoked by the TSF in order to perform security functionality.	N/A
	OE.KEY_ARCHIVAL The Operational Environment provides the ability to use split knowledge procedures to enforce two-party control to export keys necessary to resume CA functionality if the TSF should fail.	N/A
P.ACCESS_BANNER **The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by**	**O.DISPLAY_BANNER** The TOE will display an advisory warning regarding use of the TOE.	FTA_TAB.1

accessing the TOE.		

TABLE 11 TOE SECURITY OBJECTIVE MAPPING

5. Extended Components Definition

All of the extended requirements in this ST have been drawn from the CA PP. The CA PP defines the following extended requirements and since they are not refined in this ST the CA PP should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FAU_ADP_EXT.1 Audit Dependencies
- FAU_GCR_EXT.1 Generation of Certificate Repository
- FAU_SCR_EXT.1 Certificate Repository Review**
- FAU_STG_EXT.1 External Audit Trail Storage**
- FCO_NRO_EXT.2 Certificate-based proof of origin
- FCS_CDP_EXT.1 Cryptographic Dependencies
- FCS_STG_EXT.1 Cryptographic Key Storage
- FCS_CKM_EXT.1(1) Symmetric Key Generation for DEKs**
- FCS_CKM_EXT.4 Cryptographic Key Destruction**
- FCS_CKM_EXT.5 Public Key Integrity**
- FCS_RBG_EXT.1 Cryptographic Random Bit Generation**
- FCS_HTTPS_EXT.1 HTTPS Protocol**
- FCS_TLSS_EXT.1 TLS Server Protocol**
- FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication**
- FCS_CKM_EXT.1(2) Key Generation Key Encryption Keys**
- FCS_CKM_EXT.8 Key Hierarchy Entropy**
- FDP_CER_EXT.1 Certificate Profiles

- FDP_CER_EXT.2 Certificate Request Matching
- FDP_CER_EXT.3 Certificate Issuance Approval
- FDP_CSI_EXT.1 Certificate Status Information
- FDP_STG_EXT.1 Public Key Protection**
- FDP_CRL_EXT.1 Certificate Revocation List Validation**
- FDP_OCSPG_EXT.1 OCSP Basic Response Generation**
- FIA_X509_EXT.1 Certificate Validation
- FIA_X509_EXT.2 Certificate-Based Authentication
- FIA_UAU_EXT.1 Authentication Mechanism
- FIA_UIA_EXT.1 User Identification and Authentication
- FIA_ESTS_EXT.1 Enrollment over Secure Transport (EST) Server**
- FIA_X509_EXT.3 Certificate Request**
- FIA_ENR_EXT.1 Certificate Enrollment***
- FPT_KST_EXT.1 No Plaintext Key Export
- FPT_KST_EXT.2 TSF Key Protection
- FPT_SKP_EXT.1 Protection of Keys
- FPT_TUD_EXT.1 Trusted Update
- FPT_TST_EXT.2 Integrity Test*

Items marked with * are optional requirements. Items marked with ** are required based on selections. Items marked with *** are objective requirements.

6. Security Functional Requirements

This section identifies the Security Functional Requirements (SFRs) that are claimed for the TOE. The SFRs which are claimed are consistent with the SFRs that are defined in the claimed Protection Profile.

6.1 Conventions

The common criteria define four operations – assignment, refinement, selection, and iteration –that may be performed on functional requirements. This ST will highlight the operations in the following manner:

- Assignment: allows the specification of an identified parameter. Indicated with **bold text** inside square brackets.
- Refinement: allows the addition of details. Indicated with *italicized text*.
- Selection: allows the specification of one or more elements from a list. Indicated with underlined text inside square brackets.
- Iteration: allows a component to be used more than once with varying operations. Indicated with a sequential letter in parentheses following the element number of the iterated SFR. For example, FCS_CKM.1.1(1)(a) and FCS_CKM.1.1(1)(b).

6.2 Security Functional Requirements Summary

The following table lists the SFRs claimed by the TOE:

Class	Component
Security Audit (FAU)	FAU_ADP_EXT.1 Audit Dependencies FAU_GEN.1 Audit Data Generation FAU_GEN.2 User Identity Association FAU_GCR_EXT.1 Generation of Certificate Repository FAU_STG.4 Prevention of Audit Data Loss FAU_SCR_EXT.1 Certificate Repository Review FAU_SAR.1 Audit Review FAU_SAR.3 Selectable Audit Review FAU_SEL.1 Selective Audit FAU_STG_EXT.1 External Audit Trail Storage
Communication (FCO)	FCO_NRO_EXT.2 Certificate-Based Proof of Origin
Cryptographic Support (FCS)	FCS_CDP_EXT.1 Cryptographic Dependencies FCS_STG_EXT.1 Cryptographic Key Storage FCS_COP.1(5) Cryptographic Operation (Password-Based Key Derivation Function) FCS_CKM_EXT.1(1) Symmetric Key Generation for DEKs FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_CKM_EXT.4 Cryptographic Key Destruction FCS_CKM_EXT.5 Public Key Integrity FCS_COP.1(1) Cryptographic Operation (AES Encryption/Decryption) FCS_COP.1(2) Cryptographic Operation (Cryptographic Signature) FCS_COP.1(3) Cryptographic Operation (Cryptographic Hashing) FCS_COP.1(4) Cryptographic Operation (Keyed-Hash Message Authentication) FCS_RBG_EXT.1 Cryptographic Random Bit Generation FCS_HTTPS_EXT.1 HTTPS Protocol FCS_TLSS_EXT.1 TLS Server Protocol FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication FCS_CKM_EXT.1(2) Key Generation Key Encryption Keys FCS_CKM_EXT.8 Key Hierarchy Entropy
User Data Protection (FDP)	FDP_CER_EXT.1 Certificate Profiles FDP_CER_EXT.2 Certificate Request Matching FDP_CER_EXT.3 Certificate Issuance Approval FDP_CSI_EXT.1 Certificate Status Information FDP_RIP.1 Subset Residual Information Protection FDP_STG_EXT.1 Public Key Protection FDP_CRL_EXT.1 Certificate Revocation List Validation FDP_OCSPG_EXT.1 OCSP Basic Response Generation
Identification and Authentication (FIA)	FIA_X509_EXT.1 Certificate Validation FIA_X509_EXT.2 Certificate-Based Authentication FIA_UAU_EXT.1 Authentication Mechanism FIA_UIA_EXT.1 User Identification and Authentication FIA_ESTS_EXT.1 Enrollment over Secure Transport (EST) Server FIA_X509_EXT.3 Certificate Request FIA_ENR_EXT.1 Certificate Enrollment
Security Management (FMT)	FMT_MOF.1(1) Management of Security Functions Behavior (Administrator Functions) FMT_MOF.1(2) Management of Security Functions Behavior (CA/RA Functions) FMT_MOF.1(3) Management of Security Functions Behavior (CA Operations Functions) FMT_MOF.1(4) Management of Security Functions Behavior (Admin/Officer Functions) FMT_MOF.1(5) Management of Security Functions Behavior (Auditor Functions) FMT_MTD.1 Management of TSF Data FMT_SMF.1 Specification of Management Functions

	FMT_SMR.2 Restrictions on Security Roles
Protection of the TSF (FPT)	FPT_FLS.1 Failure with Preservation of Secure State FPT_KST_EXT.1 No Plaintext Key Export FPT_KST_EXT.2 TSF Key Protection FPT_RCV.1 Manual Trusted Recovery FPT_SKP_EXT.1 Protection of Keys FPT_STM.1 Reliable Time Stamps FPT_TUD_EXT.1 Trusted Update FPT_TST_EXT.2 Integrity Test
TOE Access (FTA)	FTA_SSL.4 User-Initiated Termination FTA_TAB.1 Default TOE Access Banners FTA_SSL.3 TSF-Initiated Termination
Trusted Path/Channels (FTP)	FTP_TRP.1 Trusted Path FTP_ITC.1 Inter-TSF Trusted Channel

TABLE 12 TOE SECUIRTY FUNCTIONAL COMPONENTS

6.3 Security Audit (FAU)

6.3.1 FAU_ADP_EXT.1 Audit Dependencies

FAU_ADP_EXT.1.1 The TSF shall implement audit functionality and [no additional audit functionality] in order to perform audit operations on the following audit data: [**auditable events listed in the table below that require persistent storage**]

Requirement	Auditable Events	Additional Audit Record Contents	Retention	Responsible Component
FAU_SEL.1	All modifications to the audit configuration that occur while the audit collection functions are operating.	None.	Normal	TOE
FCS_CKM.1	All occurrences of non-ephemeral and [no other] key generation for TOE related functions.	Success: public key generated	Normal	TOE
FCS_CKM.2	All occurrences of non-ephemeral and [no other] key establishment for TOE related functions.	Success: key established	Normal	TOE
FCS_CKM_EXT.4	Failure of the key destruction process for TOE related keys.	Identity of object or entity being cleared.	Normal	TOE
FCS_CKM_EXT.5	Detection of integrity violation for stored TSF data.	None.	Normal	TOE
FCS_COP.1(2)	All occurrences of signature generation using a CA signing key. Failure in signature	Name/identifier of object being signed Identifier of key used for signing. None	Extended Normal	TOE

	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	TOE
FCS_TLSS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. None	Normal	TOE
FCS_TLSS_EXT.2	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. None	Normal	TOE
FDP_CER_EXT.1	Certificate generation.	Success: [certificate object identified]	Extended	TOE
FDP_CER_EXT.2	Linking of certificate to certificate request.	Success: [certificate object identifier], [link to certificate request object identifier] Failure: Reason for failure, [link to certificate request object identifier].	Extended	TOE
FDP_CER_EXT.3	Failed certificate approvals.	Reason for failure, [link to certificate request object identifier].	Normal	TOE
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	TOE
FDP_CRL_EXT.1	Failure to generate a CRL.	None.	Normal	TOE
FDP_OCSPG_EXT.1	Failure to generate certificate status information.	None.	Extended	TOE
FIA_X509_EXT.1	Failed certificate validations.	None.	Normal	TOE
FIA_X509_EXT.2	Failed authentications.	None.	Normal	TOE
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	TOE
FIA_UIA_EXT.1	All use of the identification and authentication mechanism used for TOE	Provided user identity. Origin of the attempt (e.g., IP address).	Normal	TOE

	related roles.			
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	TOE
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	TOE
FPT_FLS.1	Invocation of failures under this requirement.	Indication that the TSF has failed with the type of failure that occurred.	Normal	TOE
FPT_KST_EXT.2	All unauthorized attempts to use TOE secret and private keys.	Identifier of user or process that attempted access.	Normal	TOE
FPT_RCV.1	The fact that a failure or service discontinuity occurred; resumption of the regular operation	The type of failure or service discontinuity.	Extended	TOE
FPT_STM.1	Changes to the time.	The old and new values for the time.	Normal	Operating system
FPT_TUD_EXT.1	Initiation of update.	Version number	Extended	TOE
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	TOE
FTA_SSL.4	The termination of an interactive session.	None.	Normal	TOE
FTA_SSL.3	The termination of a remote session by the session termination mechanism.	None.	Normal	TOE
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.	Normal	TOE
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	TOE

TABLE 13 AUDITABLE EVENTS

6.3.2 FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1 *Refinement:* The TSF shall generate *and [**no other actions**]* an audit record of the following auditable events:

a) Start-up of the *TSF* audit functions;

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

c) **All administrative actions invoked through the TSF interface**;

d) [**auditable events listed in Table 13**]]

FAU_GEN.1.2 *Refinement***:** The TSF shall *[*include*]* within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [**information specified in column three of Table 13**].

6.3.3 FAU_GEN.2 User Identity Association

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

6.3.4 FAU_GCR_EXT.1 Generation of Certificate Repository

FAU_GCR_EXT.1.1 The TSF shall [invoke the Operational Environment to store] certificates and [CRLs] issued by the TSF.

6.3.5 FAU_STG.4 Prevention of Audit Data Loss

FAU_STG.4.1: *Refinement:* The TSF shall [prevent audited events] and [**no additional actions**] if the audit trail *cannot be written to*.

6.3.6 FAU_SCR_EXT.1 Certificate Repository Review

FAU_SAR_EXT.1.1 The TSF shall [provide] the capability to search for certificates containing specified values of the following certificate fields: [

- subject name,
- serial number

] returning all matching certificates and [**their certificate request identifiers necessary to search the audit trail for events involving those certificates**].

6.3.7 FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide [Auditors] with the capability to read all information from the audit records.

FAU_SAR.1.2: *Refinement:* The TSF shall provide the audit records in a manner suitable for the *Auditor* to interpret the information.

6.3.8 FAU_SAR.3 Selectable audit review

FAU_SAR.3.1: The TSF shall provide the ability to apply [**searches**] of audit data based on [**certificate request identifiers**] associated with the event.

6.3.9 FAU_SEL.1 Selective Audit

FAU_SEL.1.1 *Refinement:* The TSF shall be able to select the set of events to be audited *by specific mechanisms* from the set of all auditable events based on the following attributes:

a) [event type]
b) [**No additional attributes**].

6.3.10 FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.1.1 The TSF shall maintain availability and integrity of audit data by storing it [locally on the TOE platform].

6.4 Communication (FCO)

6.4.1 FCO_NRO_EXT.2 Certificate-based proof of origin

FCO_NRO_EXT.2.1 The TSF shall provide proof of origin for certificates it issues in accordance with the digital signature requirements using mechanism in accordance with RFC 5280 and FCS_COP.1(2).

FCO_NRO_EXT.2.2 The TSF shall provide proof of origin for certificate status information it issues in accordance with the digital signature requirements in [CRLs (RFC 5280), OCSP (RFC 6960)] and FCS_COP.1(2).

FCO_NRO_EXT.2.3 The TSF shall require and verify proof of origin for certificate requests it receives [EST using mechanisms in accordance with FIA_ESTS_EXT.1].

FCO_NRO_EXT.2.4 The TSF shall require and verify proof of origin for public keys contained in certificate requests it receives via [proof-of-possession mechanisms in EST in accordance with FIA_ESTS_EXT.1].

FCO_NRO_EXT.2.5 The TSF shall require and verify proof of origin for revocation requests it receives in via [**the subscriber self-service revocation page of the public site**].

6.5 Cryptographic Support (FCS)

6.5.1 FCS_CDP_EXT.1(a) Cryptographic Dependencies

FCS_CDP_EXT.1.1(a): The TSF shall [implement cryptographic functionality] in order to perform [[FCS_STG_EXT.1(b), FCS_COP.1(5), FCS_CKM_EXT.1.1(1), FCS_CKM.1, FCS_CKM.2, FCS_CKM_EXT.4, FCS_CKM_EXT.5, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1, FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, FCS_CKM_EXT.8]] cryptographic operations.

6.5.2 FCS_CDP_EXT.1(b) Cryptographic Dependencies

FCS_CDP_EXT.1.1(b): The TSF shall [invoke interfaces provided by the Operational Environment] in order to perform [[FCS_STG_EXT.1(a), FCS_CKM.1, FCS_CKM_EXT.4, FCS_COP.1(2), FCS_RBG_EXT.1, FCS_CKM_EXT.1(2)]] cryptographic operations.

6.5.3 FCS_STG_EXT.1(a) Cryptographic Key Storage

FCS_STG_EXT.1.1(a) Persistent private and secret keys shall be stored within the[Operational Environment] [in an hardware cryptographic module].

Application Note:

The CA issuer keys, the CA "System" credential, and the TLS/HTTPS server key are stored in the Operational Environment's PKCS#11 Cryptographic Module.

6.5.4 FCS_STG_EXT.1(b) Cryptographic Key Storage

FCS_STG_EXT.1.1(b) Persistent private and secret keys shall be stored within the [Operational Environment] [encrypted within a key hierarchy established in accordance with [FCS_CKM_EXT.1.1(2)]].

Application Note:

The DEKS used for encrypting data stored in the Operational Environment's database are encrypted with the CA "System" credential.

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

FCS_COP.1(5) *Refinement:* The TSF shall [perform][**PBKDF2 as in RFC 2898**] in accordance with a specified cryptographic algorithm [*HMAC-SHA-256*] and *output* cryptographic key sizes [256-bit] that meet the following: [**NIST 800-132**].

Application Note:

This requirement is included solely because we use PBKDF2 to store a check value of EST user passwords as described in FCS_COP.1(4).

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

FCS_CKM_EXT.1.1(1) The TSF shall [generate] data encryption keys (DEKs) of size [256-bit] using

[

- an RBG that meets this profile (as specified in FCS_RBG_EXT.1),

].

6.5.7 FCS_CKM.1(a) Cryptographic Key Generation

FCS_CKM.1.1(a): *Refinement:* The TSF shall *[*generate, invoke interfaces provided by the Operational Environment to generate*]* *asymmetric* cryptographic keys in accordance with the specified key generation algorithm:

[

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*

]

and specified cryptographic key sizes [**3072-, 4096-, 8192- bit**].

Application Note:

The TOE generates the initial set of authentication credentials using RSA-3072. The TOE uses the environmental PKCS#11 Cryptographic Module to generate CA, System, and TLS keys which may be either RSA or ECC, but default to RSA-3072.

6.5.8 FCS_CKM.1(b) Cryptographic Key Generation

FCS_CKM.1.1(b): *Refinement:* The TSF shall *[*generate, invoke interfaces provided by the Operational Environment to generate*]* *asymmetric* cryptographic keys in accordance with the specified key generation algorithm:

[

- *ECC Schemes using "NIST curves" [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;*

]

and specified cryptographic key sizes [**256-, 384-, and 521-bit ECC**].

Application Note:

The TOE and the Operational Environment's web browser will generate ephemeral ECDH (ECDHE) keys during TLS setup since an ECDHE cipher suite is used. The TOE uses the environmental PKCS#11 Cryptographic Module to generate CA, System, and TLS keys which may be either RSA or ECC, but default to RSA-3072.

6.5.9 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1: *Refinement:* The TSF shall *[*perform*] key establishment* in accordance with the specified key *establishment* algorithm:

[

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

]

6.5.10 FCS_CKM_EXT.4(a) Cryptographic Key Destruction

FCS_CKM_EXT.4.1(a) The TSF shall [destroy] all cryptographic keys and critical security parameters which are not permanently protected from export by hardware when no longer required, in accordance with the specified cryptographic key destruction method

[

- for volatile memory, the destruction shall be executed by a [
    - destruction of reference to the key directly followed by a request for garbage collection

  ]

].

FCS_CKM_EXT.4.2(a) The TSF shall [destroy] all plaintext keying material cryptographic security parameters when no longer needed.

Application Note:

TOE critical security parameters (DEKs, PKCS#11Cryptographic Module PIN) in memory are explicitly zeroized when no longer in use. Other CSPs in the TOE (passwords associated with EST requests, LDAP, email, and database) are Java objects and a request for garbage collection is made after destroying their reference. The PKCS#11 Cryptographic Module is responsible for secure erasure of the critical security parameters it manages.

6.5.11 FCS_CKM_EXT.4(b) Cryptographic Key Destruction

FCS_CKM_EXT.4.1(b) The TSF shall [destroy] all cryptographic keys and critical security parameters which are not permanently protected from export by hardware when no longer required, in accordance with the specified cryptographic key destruction method

[

- For volatile memory, the destruction shall be executed by a [
    - a single direct overwrite consisting of [
        - zeroes

        ]

    ]

].

FCS_CKM_EXT.4.2(b) The TSF shall [destroy] all plaintext keying material cryptographic security parameters when no longer needed.

Application Note:

TOE critical security parameters (DEKs, PKCS#11Cryptographic Module PIN) in memory are explicitly zeroized when no longer in use. Other CSPs in the TOE (passwords associated with EST requests and database) are Java objects and a request for garbage collection is made after destroying their reference. The PKCS#11 Cryptographic Module is responsible for secure erasure of the critical security parameters it manages.

6.5.12 FCS_CKM_EXT.5 Public Key Integrity

FCS_CKM_EXT.5.1 The TSF shall [protect] public keys used to meet CA requirements against undetected modification through the use of [digital signatures (in accordance with FCS_COP.1(2)].

FCS_CKM_EXT.5.2 The [digital signature] used to protect a public key shall be verified upon each access to the key.

Application Note:

The TOE protects the Trust Anchor database table and the ACL database table using digital signatures. The Operational Environment protects the issuer credentials, the Apache Tomcat trust anchor key store, and the TLS server credential.

All public keys stored by the TOE are contained in standard data formats which encapsulate the public key in a digitally signed message body (X.509 v3 certificate, PKCS#10 certificate request, CMS).

Certificates in the trust anchor databases are validated when added to the trust anchor database and are validated when used in path validation.

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

FCS_COP.1.1(1): *Refinement:* The TSF shall *[perform]* [**encryption and decryption**] in accordance with a specified cryptographic algorithm:

[

- *AES-CBC (as defined in NIST SP 800-38A) mode*

]

and cryptographic key size [256-bit].

Application Note:

The TOE encrypts sensitive data using AES-CBC with 256-bit keys, using the CMS format, prior to storage in the database. The TOE uses the ISC CDK to support AES-CBC in 256-bit key sizes for TLS/HTTPS session establishment.

6.5.14 FCS_COP.1(2)(a) Cryptographic Operation (Cryptographic Signature)

FCS_COP.1.1(2)(a): *Refinement:* The TSF shall *[invoke interfaces in the operational environment to perform]* [**cryptographic signature services**] in accordance with the following specified cryptographic algorithms *[*

- *RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 3072 bits or greater that meets FIPS-PUB 186-4, "Digital Signature Standard",*
- *Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater that meets FIPS PUB 186-4, "Digital Signature Standard" with "NIST curves" P-256, P-384, and [P-521] (as defined in FIPS PUB 186-4, "Digital Signature Standard")*

].

Application Note:

Both RSA and ECDSA are supported in the evaluated configuration. The TOE uses the PKCS#11 Cryptographic Module to perform cryptographic signatures for certificate signing, CRL signing, OCSP response signing, and signatures requiring the TLS/HTTPS server certificate. The Web Browser performs cryptographic signatures for identifying the user authenticating to the TOE. The TOE itself performs signature verification operations (of TLS client certificates, certificates on ACLs, and software update packages) and signs the initial authentication certificate requests.

6.5.15 FCS_COP.1(2)(b) Cryptographic Operation (Cryptographic Signature)

FCS_COP.1.1(2)(b): *Refinement:* The TSF shall *[perform]* [**cryptographic signature and signature verification services**] in accordance with the following specified cryptographic algorithms *[*

- *RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 3072 bits or greater that meets FIPS-PUB 186-4, "Digital Signature Standard",*
- *Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater that meets FIPS PUB 186-4, "Digital Signature Standard" with "NIST curves" P-256, P-384, and [P-521] (as defined in FIPS PUB 186-4, "Digital Signature Standard")*

].

Application Note:

Both RSA and ECDSA are supported in the evaluated configuration. The TOE uses the PKCS#11 Cryptographic Module to perform cryptographic signatures for certificate signing, CRL signing, OCSP response signing, and signatures requiring the TLS/HTTPS server certificate. The Web Browser performs cryptographic signatures for identifying the user authenticating to the TOE. The TOE itself performs signature verification operations (of TLS client certificates, certificates on ACLs, and software update packages) and signs the initial authentication certificate requests.

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

FCS_COP.1.1(3): *Refinement:* The TSF shall *[*perform*]* [**cryptographic hashing services**] in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512] and message digest sizes [160, 256, 384, 512] bits that meet the following: [*FIPS Pub 180-4, "Secure Hash Standard"*].

Application Note:

The TOE performs cryptographic hashing services when creating certificate requests, issuing certificates (SHA-384, SHA-512 only), CRLs (SHA-384, SHA-512 only), and OCSP responses (SHA-1, SHA-256, SHA-384, or SHA-512). The TOE performs cryptographic hashing services when validating certificate requests, certificates, CRLs, OCSP requests (SHA-1, SHA-256, SHA-384, SHA-512 as chosen by the requester), and performing TLS/HTTPS (SHA-1). All modes are byte oriented. The Web Browser performs cryptographic hashing services when authenticating via TLS/HTTPS to the CA.

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

FCS_COP.1.1(4): *Refinement:* The TSF shall *[*perform*]* [**keyed hash message authentication**] in accordance with a specified cryptographic algorithm *HMAC-[SHA-1, SHA-256], key size [**120-800 bits**], and message digest sizes [160, 256]* bits that meet the following: [*FIPS Pub 198-1, "The Keyed Hash Message Authentication Code, and FIPS Pub 180-4, "Secure Hash Standard"*].

Application Note:

The ISC CDK uses HMAC-256 for its DRBG implementation.

The TOE, using the ISC CDK, uses PBKDF2 with HMAC-SHA-256 to store a check value for EST passwords (in other words, instead of storing the EST password or an iterated hash of the password, the TOE stores the output of the PBKDF2 function along with the required salt value, iteration count, etc.). The password is the HMAC key and can be from 15 to 100 bytes in length.

The TOE uses HMAC-SHA-1 when establishing TLS/HTTPS connections.

6.5.18 FCS_RBG_EXT.1 Cryptographic Random Bit Generation

FCS_RBG_EXT.1.1 The TSF shall [perform] all deterministic random bit generation (RBG) services in accordance with NIST Special Publication 800-90A using [HMAC_DRBG (SHA-256) and CTR_DRBG(AES-256)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [a software-based noise source, Operational Environment-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and authorization factors that it will generate.

Application Note:

The TOE uses ISC CDK's DRBG which uses HMAC_DRBG(SHA-256). Third-party entropy sources (including the PKCS#11 Cryptographic Module's DRBG) are used. It is assumed that they provide 8 bits of entropy for each byte they output. The ISC CDK's DRBG obtains at least 384-bits of entropy from the third-party source or it fails to initialize and enters its error state.

The PKCS#11 Cryptographic Module uses CTR_DRBG(AES-256) with 384-bits of entropy from a hardware source.

6.5.19 FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

6.5.20 FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1 The TSF shall implement [TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

[

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

].

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSl 3.0, TLS 1.0, and [no other TLS versions]

FCS_TLSS_EXT.1.3 The TSF shall [generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves].

6.5.21 FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication

FCS_TLSS_EXT.2.1 The TSF shall implement [TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

[

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

].

FCS_TLSS_EXT.2.2 The TSF shall deny connections from clients requesting SSL 2.0, SSl 3.0, TLS 1.0, and [no other TLS versions]

FCS_TLSS_EXT.2.3 The TSF shall [generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves].

FCS_TLSS_EXT.2.4 The TSF shall support mutual authentication of TLS clients using X.509 certificates.

FCS_TLSS_EXT.2.5 For communications configured to require TLS with mutual authentication, the TOE shall not establish a trusted channel if the client certificate is invalid.

FCS_TLSS_EXT.2.5 The TSF shall respond with a fatal TLS error if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate presented for client authentication does not match the expected identifier for the client.

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

FCS_CKM_EXT.1.1(2) The TSF shall be able to [invoke interfaces in the Operational Environment to generate] [asymmetric KEKs of [type RSA-2048, RSA-3072, ECDH-P-256, ECDH-P-384, ECDH-P-521] security strength in accordance with FCS_CKM.1 from [an RBG that meets this profile (as specified in FCS_RBG_EXT.1)]].

Application Note:

The TOE uses the PKCS#11 Cryptographic Module to generate these keys.

6.5.23 FCS_CKM_EXT.8 Key Hierarchy Entropy

FCS_CKM_EXT.8.1 Keys (DEKS or KEKS) formed from combinations or by encrypting one key with another form shall be traceable through a hierarchy of keys to a REK generated in accordance with FCS_RBG_EXT.1 using a hardware-based mechanism.

FCS_CKM_EXT.8.2 Key entropy for KEKs shall be preserved according to the sensitivity of the DEK, KEK, or key it encrypts.

FCS_CKM_EXT.8.3 Key entropy for DEKs shall be [256] bits in accordance with the sensitivity of the data encrypted.

6.6 User Data Protection (FDP)

6.6.1 FDP_CER_EXT.1 Certificate Profiles

FDP_CER_EXT.1.1 The TSF shall implement a certificate profile function and shall ensure that issued certificates are consistent with configured profiles.

FDP_CER_EXT.1.2 The TSF shall generate certificates using profiles that comply with requirements for certificates as specified in IEFT RFC 5280, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", while ensuring that the following conditions are met:

a) The version field shall contain the integer 2.

b) The issuerUniqueID or subjectUniqueID fields are not populated.

c) The serialNumber shall be unique with respect to the issuing Certification Authority.

d) The validity field shall specify a notBefore value that does not precede the current time and a notAfter value that does not precede the value specified in notBefore.

e) The issuer field is not empty.

f) The signature field and the algorithm in the subjectPublicKeyInfo field shall contain the OID for a signature algorithm specified in FCS_COP.1(2).

g) The following extensions are supported:

   a. subjectKeyIdentifier

   b. authorityKeyIdentifier

   c. basicConstraints

   d. keyUsage

   e. extendedKeyUsage

   f. certificatePolicy

h) A subject field containing a null Name (e.g., a sequence of zero relative distinguished names) is accompanied by an populated critical subjectAltName extension.

i) The subjectKeyIdentifier extension is populated with a value unique for each public key contained in a certificate issued by the TSF.

j) The authorityKeyIdentifier extension in any certificate issued by the TOE must be populated and must be the same as the subjectKeyIdentifier extension contained in the issuer's signing certificate.

k) Populated keyUsage and extendedKeyUsage fields in the same certificate contain consistent values.

FDP_CER_EXT.1.3 The TSF shall be able to generate at least 20 bits of random for use in issued certificates to be included in [serialNumber] fields, where the random values are generated in accordance with FCS_RGB_EXT.1.

6.6.2 FDP_CER_EXT.2 Certificate Request Matching

FDP_CER_EXT.2.1 The TSF shall establish a linkage from certificate requests to issued certificates.

6.6.3 FDP_CER_EXT.3 Certificate Issuance Approval

FDP_CER_EXT.3.1 The TSF shall support the approval of certificates by [CA Operations Staff, rules] issued according to a configured certificate profile.

6.6.4 FDP_CSI_EXT.1 Certificate Status Information

FDP_CSI_EXT.1.1 The TSF shall provide certificate status information whose format complies with [ITU-T Recommendation X.509v2 CRL, the OCSP standard as defined by RFC 6960].

FDP_CSI_EXT.1.2 The TSF shall support the approval of changes to the status of a certificate by [CA Operations Staff, rules].

6.6.5 FDP_RIP.1 Subset Residual Information Protection

FDP_RIP.1.1 The TSF and [Operational Environment] shall ensure that any previous information content of a resource is made unavailable upon the [allocation of the resource to, deallocation of the resource from] the following objects: [**EST passwords, TLS Session Object**].

6.6.6 FDP_STG_EXT.1 Public Key Protection

FDP_STG_EXT.1.1 The TSF shall use [an integrity mechanism] to protect the trusted public keys and certificates (trust store elements) used to validate local login, trusted channel, and external communication to the CA.

Application Note:

The TOE protects the Trust Anchor database table and the ACL database table using an integrity mechanism. The Operational Environment protects the issuer credentials, and the TLS server credential.

6.6.7 FDP_CRL_EXT.1 Certificate Revocation List Validation

FDP_CRL_EXT.1.1 A TSF that issues CRLs shall verify that all mandatory fields in any CRL issued contain values in accordance with ITU-T Recommendation X.509. At a minimum, the following items shall be validated:

a) If the version field is present, then it shall contain a 1.

b) If the CRL contains any critical extensions, then the version field shall be present and contain the integer 1.

c) If the issuer field contains a null Name (e.g., a sequence of zero relative distinguished names), then the CRL shall contain a critical issuerAltName extension.

d) The signature and signatureAlgorithm fields shall contain the OID for a digital signature algorithm in accordance with FCS_COP.1(2).

e) The thisUpdate field shall indicate the issue date of the CRL.

f) The time specified in the nextUpdate field (if populated) shall not precede the time specified in the thisUpdate field.

6.6.8 FDP_OCSPG_EXT.1 OCSP Basic Response Generation

FDP_OCSPG_EXT.1.1 The TSF shall ensure that all mandatory fields in the OCSP response contain values in accordance with FDP_CSI_EXT.1. At a minimum, the following items shall be validated:

a) The version field shall indicate a current version.

b) The signatureAlgorithm field shall contain the object identifier (OID) for a digital signature algorithm in accordance with FCS_COP.1(2).

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

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

e) The time specified in the nextUpdate field (if populated) shall not precede the time specified in the thisUpdate field.

6.7 Identification and Authentication (FIA)

6.7.1 FIA_X509_EXT.1 Certificate Validation

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

- IETF RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a certificate in the Trust Anchor Database.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the cA is set to TRUE for all CA certificates.

- The TSF shall validate the revocation status of the certificate using [a Certificate Revocation List (CRL) as specified in FDP_CSI_EXT.1].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3),
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field,
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.

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

6.7.2 FIA_X509_EXT.2 Certificate-Based Authentication

FIA_X509_EXT.2.1 TSF shall [use] X.509v3 certificates as defined by RFC 5280 to support [authentication for code signing for TOE updates, HTTPS], and [access to the Admin Site, access to the CA Site, access to the Public Site for self-service revocation, certificate access for EST, access to the RAMI interface, and access to the DBAccess interface].

FIA_X509_EXT.2.2 When the TSF cannot determine the validity of a certificate the TSF shall [not accept the certificate].

FIA_X509_EXT.2.3 The TSF shall not establish a trusted communication channel if the peer certificate is deemed invalid.

6.7.3 FIA_UAU_EXT.1 Authentication Mechanism

FIA_UAU_EXT.1.1 The TSF shall [provide] a [**certificate-based authentication mechanism**] to perform privileged user authentication.

6.7.4 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow 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;
- [Download certificate from repository;
- Respond to EST cacerts requests;
- Submit certificate requests;

- Obtain information about the TOE (version, current time, operating system type).]

FIA_UIA_EXT.1.2 The TSF shall require each user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that user, including subscriber certificate renewal, subscriber revocation requests, privileged user access, and [no other actions].

FIA_UIA_EXT.1.3 For subscriber actions, the TSF shall verify that the DN of the certificate presented by the subscriber for authentication matches that of the certificate being affected by the subscriber's actions.

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

FIA_ESTS_EXT.1.1 The TSF shall use the Enrollment over Secure Transport (EST) protocol as specified in RFC 7030 to receive, process, and respond to certificate enrollment requests from authorized clients.

FIA_ESTS_EXT.1.2 The TSF shall authenticate EST clients for re-enrollment via TLS certificate-based mutual authentication in accordance with RFC 7030 Section 3.3.2 and FCS_TLSS_EXT.2

FIA_ESTS_EXT.1.3 The TSF shall authenticate EST clients for initial enrollment and for supplemental authentication via [HTTP basic authentication in accordance with RFC 7030 section 3.2.3; TLS certificate-based mutual authentication in accordance with RFC 7030 section 3.3.2 and FCS_TLSS_EXT.2]

FIA_ESTS_EXT.1.4 The TSF shall authorize EST clients based on [the authenticated client certificate is issued by the same issuer and asserts id-kp-cmcRA in its extended key usage extension as specified by RFC 7030 Section 3.7, [**the authenticated client certificate is issued by the same issuer and its DN is identical to that in the request, the authenticated EST username and password match those held by the TOE account for the issuer and the request's subject DN's CN or the request's requested subjectAltName extension contain a name matching the EST username**]].

Application Note:

For clarity, the TSF will automatically issue a certificate in response to an EST request

- *If the certificate used to authenticate has the same issuer DN as the issuer to which the request is submitted and the certificate used to authenticate asserts id-kp-cmcRA in its extended key usage extension.*
- *If the certificate used to authenticate has the same issuer DN as the issuer to which the request is submitted and the certificate request's subject DN matches the one in the certificate request submitted.*
- *If the EST username and password match those configured in the server and the EST username matches either the CN in the request's subject DN or the EST username matches a name in the request's requested subjectAltName.*

6.7.6 FIA_X509_EXT.3 Certificate Request

FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key, CA's distinguished name, [**no other information**].

FIA_X509_EXT.3.2 The TSF validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

6.7.7 FIA_ENR_EXT.1 Certificate Enrollment

FIA_ENR_EXT.1.1 The TSF shall be able to generate a certificate request to an external certification authority to receive a CA certificate for a CA's signing key using [PKCS#10 in accordance with FIA_X509_EXT.3]

6.8 Security Management (FMT)

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

FMT_MOF.1.1(1) *Refinement:* The *[TSF, Operational Environment]* shall restrict the ability to

1. *manage the TOE locally (OE) and remotely (TSF);*
2. *configure the audit mechanism; (TSF)*
3. *configure and manage certificate profiles; (TSF)*
4. *modify revocation configuration; (TSF)*
5. *perform updates to the TOE; (OE)*
6. *perform on-demand integrity tests; (TSF)*
7. *Import and remove X.509v3 certificates into/from the Trust Anchor Database; (TSF)*

[

8. *configure certificate revocation list function; (TSF)*
9. *configure OCSP function; (TSF)*
10. *export PKCS#10 certificate request;*
11. *import CA certificate;*
12. *[**manage the ACL of the Admin Site and CA Account Sites; (TSF)***
13. ***manage the CRL store for path validation; (TSF)***
14. ***configure the default TOE access banner; (TSF)***

15. ***disable CA accounts; (TSF)***

16. ***generate certificate request for issuer; (TSF)****]*

]

to *[Administrators]*.

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

FMT_MOF.1.1(2) *Refinement:* The *[TSF]* shall restrict the ability to

1. *approve and execute the issuance of certificates;*

2. *configure subscriber self-service request constraints;*

[

3. *configure automated certificate approval management;*

]

to *[CA Operations Staff]*.

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

FMT_MOF.1.1(3) *Refinement:* The *[TSF]* shall restrict the ability to

1. *approve certificate revocation;*

[

2. *no other function*

]

to [CA Operations Staff].

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

FMT_MOF.1.1(4) *Refinement:* The *[Operational Environment]* shall restrict the ability to

1. *perform destruction of sensitive data when no longer needed;*

2. *[participate as a second party for archival and recovery;*

3. *perform private or secret key or critical data export]*

to *[Administrators]*.

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

FMT_MOF.1.1(5) *Refinement:* The *[TSF]* shall restrict the ability to

- Delete entries from the audit trail
- [Search the audit trail]

to [Auditors].

Application Note:

The TOE does not include a capability to remove entries from the audit trail. Entries in the audit trail remain in perpetuity.

6.8.6 FMT_MTD.1 Management of TSF Data

FMT_MTD.1.1 The TSF shall restrict the ability to *manage the TSF data* to [**Administrators, CA operations Staff, and Auditors**].

6.8.7 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1: *Refinement:* The *[TSF, Operational Environment]* shall be capable of performing the following management functions: [

1. Ability to manage the TOE locally (OE) and remotely (TOE);
2. Ability to perform updates to the TOE; (TOE)
3. Ability to perform archival and recovery; (OE)
4. Ability to manage the audit mechanism; (TOE)
5. Ability to configure and manage certificate profiles; (TOE)
6. Ability to approve and execute the issuance of certificates; (TOE)
7. Ability to approve certificate revocation; (TOE)
8. Ability to modify revocation configuration; (TOE)
9. Ability to configure subscriber self-service request constraints; (TOE)
10. Ability to perform on-demand integrity tests; (TOE)
11. Ability to destroy sensitive user data when no longer needed; (OE)
12. Ability to import and remove X.509v3 certificates into/from the Trust Anchor Database; (TOE)

[

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

14. [Ability to modify the CRL configuration, Ability to modify the OCSP configuration]; (TOE)

15. Ability to configure the cryptographic functionality; (TOE)

]]

Application Note:

The TOE provides 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, and 15. For 11 there is no sensitive user data that needs to be destroyed.

The Operational Environment provides 1 (Operating System), 3 (PKCS#11 Cryptographic Module), 15 (PKCS#11 Cryptographic Module).

6.8.8 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1: *Refinement:* The TSF *and [Operational Environment]* shall maintain the roles:

- Administrator,
- Auditor,
- CA Operations Staff,
- *[no other roles]*

Application Note:

The TOE defines 5 roles that can be categorized in to the three roles above as described in Section 9.6.

The Operational Environment maintains part of the administrator role by controlling login access to the operating system, and access to the PKCS#11 Cryptographic Module.

FMT_SMR.2.2: *Refinement:* The TSF *and [Operational Environment]* shall be able to associate users with roles.

Application Note:

The TOE associates users with roles using certificates.

The Operational Environment maintains part of the administrator role by controlling login access to the operating system and the TOE's local console.

FMT_SMR.2.3: *Refinement:* The TSF *and [no other component]* shall ensure that the conditions

[

- No identity is authorized to assume both an Auditor role and any of the other roles in FMT_SMR.2.1; and
- No identity is authorized to assume both a CA Operations Staff role and any of the other roles in FMT_SMR.2.1

}

are satisfied.

6.9 Protection of the TSF (FPT)

6.9.1 FPT_FLS.1 Failure with Preservation of Secure State

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [DRBG failure, integrity failure on Trust Anchor database, [**ISC CDK failure**], [**PKCS#11 Cryptographic Module failure**, **Database inaccessible**]].

6.9.2 FPT_KST_EXT.1 No Plaintext Key Export

FPT_KST_EXT.1.1 The TSF and [Operational Environment] shall prevent the plaintext export of [

- **CA issuer keys**
- **CA "System" credential (asymmetric key)**
- **TLS/HTTPS Server key**
- **TLS ECDHE keys**
- **CA CMS DEKs**].

Application Note:

The Operational Environment's PKCS#11 Cryptographic Module supplies this functionality for all keys except for CA CMS DEKs, which are encrypted with the CA "System" credential's public key, and TLS ECDHE keys which are ephemeral. In the evaluated configuration, the PKCS#11 Cryptographic Module provides no mechanism for plaintext export of the keys. The TOE protects the CA CMS DEKs by encrypting them with the "System" credential's public key and provides no mechanism to export them as plaintext. The TOE protects the TLS ECDHE keys by erasing them from memory immediately after use.

6.9.3 FPT_KST_EXT.2 TSF Key Protection

FPT_KST_EXT.1.1 The TSF and [Operational Environment] shall prevent unauthorized use of all TSF private and secret keys.

Application Note:

The TOE prevents unauthorized use of the keys in the Operational Environment's PKCS#11 Cryptographic Module by not providing user accessible interfaces that allow unauthorized use of the TOE. The Operational Environment (operating system and PKCS#11 Cryptographic Module) prevent the unauthorized use of the keys in the PKCS#11 Cryptographic Module when accessed outside of the TOE's interfaces.

6.9.4 FPT_RCV.1 Manual Trusted Recovery

FPT_RCV.1.1 After [**integrity failure**] the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

6.9.5 FPT_SKP_EXT.1 Protection of Keys

FPT_SKP_EXT.1.1 The TSF shall [interface with the Operational Environment to implement] the ability to prevent reading of all pre-shared keys, private, and secret keys (e.g., KEKs, DEKs, session keys).

Application Note:

The TSF provides no mechanisms allowing the reading of any such 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 any such keys.

6.9.6 FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1 *Refinement:* The TSF shall *[interface with the Operational Environment to provide]* reliable time stamps.

Application Note:

The TOE uses the operating system's API calls to obtain time information.

6.9.7 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The TSF shall [implement] the ability to check for updates and patches to the TOE.

FPT_TUD_EXT.1.2 The TSF shall [implement] the ability to provide Administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3 The TSF shall [implement] the ability to verify firmware/software updates to the TOE using a digital signature prior to installing those updates.

FPT_TUD_EXT.1.4 The TSF shall [implement] the ability to verify the digital signature whenever the software or firmware is externally loaded into the TOE and if verification fails, the TSF shall [**stop the update process**].

6.9.8 FPT_TST_EXT.2 Integrity Test

FPT_TST_EXT.2.1 A [digital signature algorithm according to FCS_COP.1(2)] shall be applied to the [Trust Anchor Database element(s), [**Access Control Lists containing certificates**]].

FPT_TST_EXT.2.2 Integrity shall be verified at [initialization, on-demand by a privileged user].

6.10 TOE Access (FTA)

6.10.1 FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1: *Refinement:* The TSF shall *[implement] the ability to* allow *privileged* user-initiated termination of the *privileged* user's own interactive session.

Application Note:

The TOE's web pages include a logout link to terminate the session.

6.10.2 FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1: *Refinement:* Before establishing *a privileged* user session the TSF shall display *an Administrator-configured* advisory *notice and consent* warning message regarding unauthorized use of the TOE.

6.10.3 FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1: *Refinement:* The TSF shall terminate *a remote* interactive session after an [***Administrator-configurable*** **time interval of session inactivity**].

6.11 Trusted Path/Channels (FTP)

6.11.1 FTP_TRP.1 Trusted Path

FTP_TRP.1.1: *Refinement:* The TSF shall *use [HTTPS, TLS] to* provide *a trusted* communication path between itself and *remote subscribers and privileged users* that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [disclosure].

FTP_TRP.1.2: *Refinement:* The TSF shall permit *remote subscribers* and privileged users to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [initial subscriber and privileged user authentication and all remote administration actions].

6.11.2 FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1: *Refinement:* The TSF shall *use [HTTPS, TLS] to* provide a *trusted* communication channel between itself and *authorized external IT entities supporting the following capabilities: [RA, [**Audit***

Server]] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

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

FTP_ ITC.1.3 The TSF shall initiate communication via the trusted channel for [**no services**].

7. Security Assurance Requirements

This section identifies the Security Assurance Requirements (SARs) claimed for the TOE. The SARs claimed are consistent with the SARs that are defined in the Protection Profile.

7.1 Class ADV: Development

7.1.1 ADV_FSP.1 Basic Functional Specification

Developer action elements:

ADV_FSP.1.1D	The developer shall provide a functional specification.
ADV_FSP.1.2D	The developer shall provide a tracing from the functional specification to the SFRs.
Developer Note:	As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPE and AGD_PRE documentation, coupled with the information provided in the TSS of the ST. 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.

Content and presentation elements:

ADV_FSP.1.1C	The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.2C	The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.3C	The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

ADV_ FSP.1.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
ADV_ FSP.1.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

7.2 Class AGD: Guidance Documents

7.2.1 AGD_OPE.1 Operational User Guidance

Developer action elements:

AGD_OPE.1.1D	The developer shall provide operational user guidance.
Developer Note:	Rather than repeat information here, the developer should review the assurance activities for this component to ascertain the specifics of the guidance that the evaluator will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

Content and presentation elements:

AGD_OPE.1.1C	The operational user guidance shall describe, for each privileged user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.
AGD_OPE.1.2C	The operational user guidance shall describe, for each privileged user role, how to use the available interfaces provided by the TOE in a secure manner.
AGD_OPE.1.3C	The operational user guidance shall describe, for each privileged user role, the available functions and interfaces, in particular all security parameters under the control of the privileged user, indicating secure values as appropriate.
AGD_OPE.1.4C	The operational user guidance shall, for each privileged user role, clearly present each type of security-relevant event relative to the privileged user-accessible functions that need to be performed, including changing the security

	characteristics of entities under the control of the TSF.
AGD_OPE.1.5C	The operational user guidance shall identify all possible modes of operation of
AGD_OPE.1.6C	The operational user guidance shall, for each privileged user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.
AGD_OPE.1.7C	The operational guidance shall be clear and reasonable.

Evaluator action elements:

AGD_OPE.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

7.2.2 AGD_PRE.1 Preparative Procedures

Developer action elements:

AGD_PRE.1.1D	The developer shall provide the TOE, including its preparative procedures.
Developer Note:	As with the operational guidance, the developer should look to the assurance activities to determine the required content with respect to preparative procedures.

Content and presentation elements:

AGD_PRE.1.1C	The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.
AGD_PRE.1.2C	The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

AGD_PRE.1.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
AGD_PRE.1.2E	The evaluator *shall apply* the preparative procedures to confirm that the TOE can be prepared securely for operation.

7.3 Class ALC: Life-Cycle Support

7.3.1 ALC_CMC.1 Labeling of the TOE

Developer action elements:

ALC_CMC.1.1D	The developer shall provide the TOE and a reference for the TOE.
Content and presentation elements:	
ALC_CMC.1.1C	The TOE shall be labeled with its unique reference.
Evaluator action elements:	
ALC_CMC.2.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

7.3.2 ALC_CMS.1 TOE CM Coverage

Developer action elements:	
ALC_CMS.2.1D	The developer shall provide a configuration list for the TOE.
Content and presentation elements:	
ALC_CMS.2.1C	The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.
ALC_CMS.2.2C	The configuration list shall uniquely identify the configuration items.
Evaluator action elements:	
ALC_CMS.2.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

7.4 Class ASE: Security Target Evaluation

As per activities defined in Common Methodology for Information Technology Security Evaluation, Version 3.1 Revision 5, April 2017.

7.5 Class ATE: Tests

7.5.1 ATE_IND.1 Independent Testing – Conformance

Developer action elements:	
ATE_IND.1.1D	The developer shall provide the TOE for testing.
Content and presentation elements:	
ATE_IND.1.1C	The TOE shall be suitable for testing.
Evaluator action elements:	
ATE_IND.1.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.1.2E	The evaluation shall test a subset of the TSF

	to confirm that the TSF operates as specified.

7.6 Class AVA: Vulnerability Assessment

7.6.1 AVA_VAN.1 Vulnerability Survey

Developer action elements:	
AVA_VAN.1.1D	The developer shall provide the TOE for testing.
Content and presentation elements:	
AVA_VAN.1.1C	The TOE shall be suitable for testing.
Evaluator action elements:	
AVA_VAN.1.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
AVA_VAN.1.2E	The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE.
AVA_VAN.1.3E	The evaluator *shall conduct* penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

8. Security Requirements Rationale

The SFRs and SARs which are claimed in Sections 6 and 7 are consistent with the SFRs that are defined in the claimed Protection Profile.

9. TOE Summary Specification

The following sections identify the security functions of the TOE and describe how the TSF meets each claimed SFR.

Definitions

The following terms are used throughout the TSS

- Issuer – The entity that issues the certificate. We use it to refer to the CA Account, its credentials (certificate and private key), ACLs, profiles, and other associated data.
- CA Account – Within the TOE each issuer is maintained in a CA account. An administrator will create the CA account to hold the issuer's credentials, ACLs, profiles, and other associated data. Each CA account has a set of tables in the database in which its data, including its associated audit data, is stored.

- Privileged User – A privileged user is someone holding a role within the TOE's authentication system (Administrator, CA Operations Staff, or Auditor).

9.1 Security Audit (FAU)

9.1.1 FAU_ADP_EXT.1 Audit Dependencies

Auditing of the TSF is performed by two (2) components: the TOE and the Operating System. The auditable events are listed in Table 13 Auditable Events and each is labeled with the responsible component.

All data requiring extended retention is audited by the TOE, which stores the events in the environmental database, and access is limited to the auditor role. Events requiring normal retention are collected by the TOE except for those events related to FPT_STM.1. Events related to FPT_STM.1 are audited by the environmental Operating System as the it controls access to, and provides the capability to, change the time of the system upon which the TOE relies for reliable timestamps completely independent of the TOE.

The TOE stores its events in the environmental database as described in Section 9.1.2 unless the database is unavailable. In the event of this critical error, the TOE stores the database failure event in a text based diagnostic log file in the environmental file system and shuts down.

The database and diagnostic log file are protected by the environmental Operating System. Auditors view the TOE's audit logs using the TOE's web interface which provides the ability to read and search the audit record.

The TOE uses the local diagnostic text file (`/usr/local/certagent7/ca.log, c:\Program Files\CertAgent 7\ca.log`) to store diagnostic information about the TOE's startup and shut down processes and the health of the database.

The environmental Operating Systems maintain their own event log systems. In the evaluated configuration, the event logs can be viewed as follows:

- Using the Windows Server 2012R2 Windows Event Viewer application found under the Tools menu of the Server Manager application that starts upon login.
- Using the CentOS 6.7 the system log files are located in `/var/log` and can be viewed using an editor of the administrator's choice.

Local logon to the Operating System is required to view the operating system log files or the local diagnostic file.

If the TOE cannot understand an incoming certificate request it will reject the data; the TOE will only audit failures that occur after the TOE has accepted the incoming data as a legitimate certificate request to which it has assigned a request ID.

9.1.2 FAU_GEN.1 Audit Data Generation

The TOE audits the startup of its audit functions, and the use of all the administrator functions required by the FMT SFRs. The TOE also generates audit records for all the auditable events identified in Table 14 below. For each auditable event, the date, time, type, subject identity (usually the DN from the certificate used to authenticate), and outcome of the event is recorded. Additional data is collected as listed in Table 14. The TOE does not include different levels of audit. All audit data is generated by the TOE in response to actions taken by the TOE itself, privileged users, subscribers, or relying parties.

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). As described in Section 9.1.1, these records are stored by the Operating System in its own audit system. The Operating System also protects these records and controls access to them.

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.

There is one administrative audit log table in the database (Admin table) and each issuer (CA account) has its own audit log table (CA table) in the database. In general, the administrative log contains actions performed by administrators and HTTPS/TLS records while the issuer audit log contains actions performed by the CA Operations Staff. The table in below details which items go into which table or audit trail. If the database is not available for some reason, the TOE will place diagnostic information in a local text file as described in Section 9.1.1.

The TOE generates audit records for the auditable events identified in the second column of Table 14 below:

Requirement	Auditable Events	Additional Audit Record Contents	Retention	Event Type	Audit Location
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. Failure in signature generation	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
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] Failure: Reason for failure, [link to certificate request object identifier].	Extended	Request	CA table
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
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
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	The fact that a failure or service discontinuity occurred Resumption of the regular operation	TSF failure types that are available on recovery.	Normal	System, Login, NIAP	Admin table, local log file for failure notices
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. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.	Normal	TLS Session	Admin table
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

TABLE 14 TOE AUDITABLE EVENT TYPES AND STORAGE LOCATIONS

The TOE defines the following event types for audit events in the log:

- ACL
- Audit
- CA Account
- Certificate
- Configuration
- Credential
- CRL
- Database
- DBAccess
- Email
- EST
- Job
- Login
- NIAP
- OCSP
- PIN
- RAMI

- Request
- System
- TLS Session
- User

Each audit entry type can be created by many different events and with additional data as shown in the following tables:

Event Type	Auditable Events
Credential	• Generated “System” credential from UI ○ FDP_CER_EXT.1 • Updated “System” credential from UI
PIN	• Entered system PIN
ACL	• Added/Removed cert to/from Admin ACL, or update permission of a cert in Admin ACL - <permission>, <cert DN>, and <issuer DN> ○ FMT_SMR.2
Audit	• Updated audit trail configuration - <changes> ○ FAU_SEL.1 • Audit function started • Added/Remove/Updated saved audit search - <saved name>

Login	• Logged in by user to Admin site
	  ○ FIA_X509_EXT.1
	  ○ FIA_X509_EXT.2
	  ○ FIA_UAU_EXT.1
	  ○ FIA_UIA_EXT.1
	• Failed to login
	  ○ FPT_RCV.1
	  ○ FCS_CKM_EXT.5
	• Logged out by user to Admin site
	  ○ FTA_SSL.4
	• Logged out due to session timeout
	  ○ FTA_SSL.3

Database	• Updated database configuration - <changes>
Job	• Added/Removed/Updated/Ran/Locked/Unlocked job - <job description>, [<ca account>]

CA Account	• Created CA account - <ca name>
	• Added/Removed certs to/from CA ACL, or update permission of a cert in CA ACL - <ca name>, <permission>, <cert DN>, and <b64-encoded cert>
	  ○ FMT_SMR.2
	  ○ FPT_FLS.1
	• Enabled/Disabled CA account - <ca name>
	• Invalid RAMI request when invalid CA account specified - <error>
	• Invalid DBAccess request when invalid CA account specified - <error>, [<post>]
	• Invalid EST request when invalid CA account specified - <ca name>, <error>, [<client auth info>], [<EST request>],[<return HTTP code>]
	• Unauthorized to login to the CA Account Site because user not in the ACL of any CA accounts
	  ○ FPT_KST_EXT.2

Email	• Sent error alert email - <to address> and <error> • Updated email configuration - <changes>

System	• Fatal error occurred - <error>
	  ○ FPT_FLS.1
	  ○ FPT_RCV.1
	  ○ FCS_CKM_EXT.5
	• Events triggered from installer
	  ○ Started installation - <version>
	  ○ Generated key pair, self-signed certificate on HSM for the system – <serial>, <subject DN>, <b64-encoded cert>
	  ▪ FDP_CER_EXT.1
	  ○ Generated key pair, self-signed cert on HSM for the initial CA account and applied changes - <ca name>, <serial>, <subject DN>, <b64-encoded cert>
	  ▪ FDP_CER_EXT.1
	  ▪ FCS_COP.1(2)
	  ○ Saved CA cert to file - <file path>
	  ○ Imported Root CA cert into Java trust keystore - <ks path>, <alias>
	  ○ Issued a CRL by initial CA account - <ca name>
	  ○ Saved CRL to file - <file path>
	  ○ Generated key pair and certificate request for Admin/Auditor/CA operations staff - <dn>, <b64-encoded p10>
	  ▪ FCS_CKM.2
	  ○ Submitted Admin/Auditor/CA operations staff's p10 to CA account - <ca name>, <request ID>
	  ○ Issued Admin/Auditor/CA operations staff certificate - <serial>, <subject DN>, <b64-encoded cert>, <signer: serial, subject>
	  ▪ FCS_CKM.2
	  ○ Saved cert to file - <file path>
	  ○ Saved PKCS12 to file - <file path>
	  ○ Generated key pair and certificate request on HSM for TLS - <dn>, <b64-encoded p10>
	  ▪ FCS_CKM.1
	  ▪ FCS_CKM.2
	  ○ Submitted TLS request to CA account - <ca name>, <request ID>
	  ○ Issued TLS certificate - <serial>, <subject DN>, <b64-encoded cert>, <signer: serial, subject>
	  ▪ FCS_CKM.2
	  ○ Installed TLS certificate into HSM
	  ○ Renamed TLS cert label in HSM - <current label>, <new label>
	  ○ Renamed CA cert label in HSM - <current label>, <new label>
	  ○ Updated configuration file
	  ○ Completed installation
	• Events triggered from update process
	  ○ Verified update package - <package version>, <signer serial, subject, issuer, validity>
	  ○ Started update process - <signed updated file>,
	  ○ Failed to update - <error>
	  ○ Running update script - <version>
	  ▪ FPT_TUD_EXT.1
	  ○ Update details - <output from the update script in the package>
	  ○ Completed update
	• Stopping CA

DBAccess	• Queried audit trail of Admin site via DBAccess API- <sql> • Listed CA account names via DBAccess API • Queried DN for CA account via DBAccess API • Validated if user is authorized to use the DBAccess API

NIAP	• Started integrity test on startup
	  ○ FPT_FLS.1
	• Ran integrity test on rust anchor/ACL - <result>
	  ○ FPT_RCV.1
	  ○ FCS_CKM_EXT.5
	  ○ FPT_FLS.1
	• Completed integrity test on startup
	• Signed trust anchor/ACL database when integrity setting changed from disabled to enabled
	  ○ FPT_FLS.1
	• Updated NIAP settings - <changes>
	• Added/Removed CRL used for path validation - <issuer DN>
	• Added/Removed trust anchor for path validation - <serial>, <dn>, <b64-encoded cert>
	  ○ FDP_STG_EXT.1

TLS Session	• Failed certificate validations during TLS connection establishment • Failed authentications during TLS connection establishment • Initiation of the trusted channel (IP address of the client) • Termination of the trusted channel (IP address of the client) • Failure of the trusted channel functions (IP address of the client) • Failure to establish a HTTPS/TLS session (IP address of the client) • Establishment/Termination of a HTTPS/TLS session (IP address of the client)

TABLE 15 ADMIN TABLE AUDITABLE EVENT DETAILS

Event Type	Auditable Events

Request	• Processed a request - <req ID>, <serial>, <dn>, <signer subject DN>, <signer serial>
	  ○ FDP_CER_EXT.1
	  ○ FDP_CER_EXT.2
	  ○ FDP_CER_EXT.3
	  ○ FCS_CKM_EXT.5
	  ○ FCS_COP.1(2)
	• Rejected a request [and sent email to subscriber] - <req ID>, <reason>, [<email>]
	• Reconsidered a request - <req ID>
	• Sent email to subscriber after certificate issuance - <email>, <reqID>
	• Assigned a request to other profile - <new profile>, <req ID>
	• Assigned all requests from a deleted profile to master profile - <deleted profile>
	• Updated contact email addresses associated with the request - <req ID>
	• Failed to link a certificate with a request - <serial>, <req ID>
	  ○ FDP_CER_EXT.2

Certificate	• Changed cert status - <serial>, <status> • Assigned a cert to other profile - <new profile>, <serial> • Ran process to check for expired cert - <# of cert found>, <elapsed time> • Exported certs via CACLI - <notBeforeDate range> • Imported cert via CACLI [and update next serial number] - <serial>, [<next serial number changes>] • Assigned all certs from a deleted profile to master profile - <deleted profile> • Sent cert expiration reminder - <email>, <serial> • Sent certificate retrieval notification - <email>, <serial> • Updated contact email addresses associated with the cert - <serial> • Published/Removed a cert to/from an LDAP - <serial>, <LDAP host and port>, <LDAP entry DN> • Changed status of affected certificates from pending revocation to revoked after issuing a CRL

CRL	• Issued a CRL [and deleted old CRL] - <signer serial>, <signer DN>, [<# of CRL deleted>]
	  ○ FCS_COP.1(2)
	  ○ FDP_CRL_EXT.1
	• Imported a CRL via CACLI
	• Updated CRLNum via CACLI after importing a CRL
	• Publish a CRL to an LDAP - <ldap host and port>, <LDAP entry DN>
	• Started automated CRL issuance - <frequency>
	• Stopped automated CRL issuance

OCSP	• Processed OCSP request - <serial>, <returned status>, <signer: serial, subject DN>
	  ○ FCS_COP.1(2)
	  ○ FDP_OCSP_EXT.1

User	• Submitted certificate request - <from source: browser, upload, sub CA, CACLI, installer>, <req ID> • Certificate retrieved by subscriber - <req ID> • Processed self-service cert revocation request - <serial>, <revocation status>, <revocation date>

Login	• Logged in by user to CA Account Site
	  ○ FIA_X509_EXT.1
	  ○ FIA_X509_EXT.2
	  ○ FIA_UAU_EXT.1
	  ○ FIA_UIA_EXT.1
	  ○ FPT_KST_EXT.2
	• Failed to login
	  ○ FPT_KST_EXT.2
	• Logged out by user to CA Account Site
	  ○ FTA_SSL.4
	• Logged out due to session timeout
	  ○ FTA_SSL.3

Credential	• Published CA cert to an LDAP - <ldap host and port>, <LDAP entry DN>
	• Decrypted encrypted password/PIN using “System” credential
	• Created a certificate request from current cert
	• Installed CA cert onto HSM - <subject DN>
	• Applied a new credential to the account
	• Generated a new key pair on HSM and cert request - <DN>, <b64-encoded P10>
	  ○ FCS_CKM.1, FCS_CKM.2
	• Generated a new root credentials - <serial>, <issuer DN>, <subject DN>, <b64-encoded cert>
	  ○ FCS_CKM.1
	  ○ FCS_COP.1(2)
	• Submitted cert request to superior CA - <submitted time and date>
	• Configured CA to use an existing credential on HSM - <serial>, <issuer DN>, <subject DN>, <b64-encoded cert>

RAMI	• Unauthorized access
	  ○ FPT_KST_EXT.2
	• Processed a request - <req ID>, <serial>, <subject DN>, <signer serial, subject DN>
	  ○ FCS_COP.1(2)
	• Issued a CRL [and deleted old CRL] - <signer serial>, <signer DN>, [<# of CRL deleted>]
	• Changed status of a cert - <serial>, <status>

DBAccess	• Unauthorized access • Queried audit trail of this CA account - <sql> • Queried certificate table of this CA account - <sql> • Got index information of certificate table of this CA account • Updated index of the certificate table of this CA account - <sql> • Replaced contact email address list associated with the certificate entries of this CA account - <old address>, <new address>, <update count> • Validated if user is authorized to use the DBAccess API of this CA account

Configuration	• Updated account configuration - <changes>
	  ○ Revocation policy
	  ○ Enrollment – Configuration/Web/RAMI/EST
	  ○ Public Site setting
	  ○ Certificate Issuance – Extension/Properties/Filter
	  ○ OCSP Responder Setting
	  ○ CRL Processing Setting
	  ○ Email Configuration
	• Added/Removed/Updated an LDAP configuration
	• Enabled/Disabled this CA account
	• Added EST subscriber - <subscriber name>
	• Removed EST subscriber - <subscriber name>
	• Updated EST password for subscriber - <subscriber name>
	• Added/Updated/Removed saved audit search
	• Created a profile [and copied setting from other profile] - <profile name>, <copied profile name>
	• Removed a profile - <profile name>
	• Deleted profile's configuration - <profile name>
	• Deleted the ACL for profile - <profile name>
	• Added/Removed cert to/from profile ACL, or update permission of a cert in profile ACL - <profile name>, <permission>, <cert DN>, and <issuer DN>
	  ○ FMT_SMR.2
	• Updated profile's display name - <profile name>, <changes>
	• Updated next serial number - <changes>

EST	• Unauthorized access
	  ○ FPT_KST_EXT.2
	• Processed EST simpleenroll/simplereenroll event - <issued cert info: req ID, serial, subject DN>, <signer: serial, subject DN>, <EST subscriber info: <subscriber name> or <client: serial, subject DN, issuer DN>>, <EST request: b64-encoded p10>, <EST response: <HTTP code>: <b64-encode P7>>
	  ○ FCS_COP.1(2)
	  ○ FIA_EST_EXT.1

TABLE 16 CA TABLE AUDITABLE EVENT DETAILS

9.1.3 FAU_GEN.2 User Identity Association

As stated in Section 9.1.2, each audit event has an associated subject identity which is audited as the Client ID.

For the TOE, if the event results from the actions of a user or entity authenticated by using a certificate, this is always the Subject DN of the certificate used to authenticate. For events that are generated by the TOE during installation or through automated tasks, the Client ID field contains `(n/a)` and the Client field contains `[system]`. For events generated via unauthenticated pages, such CRL retrieval via the public site, the Client ID field contains `(n/a)` and the Client field contains the IP address that accessed the page.

The environmental Operating System uses the login name of the user account causing the event.

9.1.4 FAU_GCR_EXT.1 Generation of Certificate Repository

The TOE stores certificates and CRLs it issues in tables in the environmental database. When an issuer account is created, the TOE uses the Java JDBC API to creates tables in the database to hold the certificates, CRLs, and audit information for the new issuer.

9.1.5 FAU_STG.4 Prevention of Audit Data Loss

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.

9.1.6 FAU_SCR_EXT.1 Certificate Repository Review

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.

9.1.7 FAU_SAR.1 Audit review

Any privileged user who appears on an auditor ACL within the TOE is a member of the auditor role for that portion of the TOE to which the ACL applies. There is an ACL for the administrative site and an ACL for each CA account. Someone may be an auditor on the ACL for the administrative site and/or one or more CA accounts sites. Restrictions on security roles are described in Section 9.6.8.

The auditor role has the ability to read all the information from the TOE audit records (except those accessible only by local login to the OS) and perform searches of the TOE's audit data using the web interface. The auditor role can also export audit data in a comma delimited file for processing in another program or access audit data through the DBAccess interface.

There is one administrative audit log that is stored in a table in the database and each issuer (CA account) has its own audit log that is stored in a separate table for each issuer in the database. In general, the administrative log contains actions performed by administrators and the issuer audit log contains actions performed by the CA Operations Staff. The table in Section 9.1.2 details which items go into which table.

No user, including members of the auditor role, has the ability to delete or modify the audit data via the TOE. The TOE does not provide any interface or mechanism to complete such actions. The files making up the database and local diagnostic audit log are protected by the environmental Operating System and its access controls.

9.1.8 FAU_SAR.3 Selectable audit review

As noted above in Section 9.1.4 auditors can search the certificate database to obtain the Request ID. The auditor can then search the audit trail and specify the request ID in the event field.

It is also possible to obtain the request ID solely by searching the audit logs. It is not obvious from the interface that this is possible, but by using the Event matches field and wildcard searches this can be accomplished. For example, specifying Event matches: *serial number* will return all audit entries containing that serial number. At least one of those results will contain the Request Identifier that uniquely identifies and links requests to certificates within the TOE.

9.1.9 FAU_SEL.1 Selective Audit

Specific auditable events can be selected by event type in the audit configuration portion of the CertAgent Administrative Site. As the TOE's audit trail retains everything for an extended period of time, this option allows the auditor to enable and disable collection of events of based on their perceived value. For example, events generated by TLS Session activity may only be of interest when there is a connection issue, and an administrator may elect to enable them only when something is not working properly.

9.1.10 FAU_STG_EXT.1 External Audit Trail Storage

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.

Audit trail data may be transferred to an external IT entity by having that entity use the DBAccess REST API. This connection is client-authenticated and encrypted using TLS/HTTPS. The external IT entity is expected to initiate the connection to the TOE and poll the TOE periodically to obtain updated audit entries.

9.2 Communication (FCO)

The TOE uses the PKCS#11 Cryptographic Module to digitally sign certificates, CRLs, and OCSP responses it creates using one of the algorithms in the table below as determined by the issuing certificate's key type. The TOE also uses the PKCS#11 Cryptographic Module to digitally sign the Trust Anchor database table, the ACL database table, and to decrypt DEKs used to encrypt sensitive data that the TOE persists in the database. The choice of key type and size is made when the issuer credential or "System" credential is created and remains so until a new credential is created. The table differentiates between key sizes that can be generated vs those that it can support (*i.e.,* the PKCS#11 Cryptographic Module can be used to generate a public/private key pair using its own tools that may support different key sizes than the TOE itself can generate on the PKCS#11 Cryptographic Module).

Key	Supported Key Sizes	Generates Key Sizes

Type		
RSA	2048-bits or greater	3072, 4096, 8192
ECDSA	All NIST defined B, K, and P curves with key length 256-bits or higher	NIST curves: P-256, P-384, P-521

TABLE 17 SUPPORTED KEY TYPES AND SIZES

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

The TOE supports EST's simple enrollment as defined in FIA_ESTS_EXT.1. Requests received via EST must be digitally signed. The TOE's EST responses contain the issued and signed certificate matching the request. Simple EST doesn't support revocation. 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.

For the purposes of cross-certification or to become subordinate to another issuer, the TOE can generate certificate requests in the PKCS#10/RFC2898 format.

Associated SFR: FCO_NRO_EXT.2 Certificate-based proof of origin

9.3 Cryptographic Support (FCS)

9.3.1 FCS_CDP_EXT.1 Cryptographic Dependencies

As explained earlier in this document, the TOE includes its own cryptographic module (the ISC CDK) and also uses an external component (the PKCS#11 Cryptographic Module) for cryptographic functions. The easiest summary is that the PKCS#11 Cryptographic Module performs all sensitive private key operations, and the TOE uses the ISC CDK for everything else. Both the ISC CDK and the PKCS#11 Cryptographic Module in the evaluated configuration are FIPS 140-2 validated.

The following table attempts to break this down further:

Functionality/SFRs	TOE Component Invoking the Functionality	Cryptographic Provider Performing the Function
TLS/HTTPS session establishment (includes ephemeral key gen and key derivation, hashing, HMAC, and symmetric encryption as required by TLS/HTTPS) FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1, FCS_CKM.2, FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1	Apache Tomcat	ISC CDK
TLS/HTTPS session establishment (digital signature required by TLS/HTTPS) FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1, FCS_COP.1(2), FCS_RBG_EXT.1	Apache Tomcat	PKCS#11 Cryptographic Module
Asymmetric key generation of TLS/HTTPS server credentials FCS_CKM.1, FCS_RBG_EXT.1	CertAgent	PKCS#11 Cryptographic Module
Asymmetric key generation of TLS/HTTPS client authentication credentials FCS_CKM.1, FCS_RBG_EXT.1	CertAgent	ISC CDK
Certificate validation, certificate request validation, trust anchor table validation FCS_COP.1(2), FCS_COP.1(3), FCS_CKM_EXT.5	CertAgent	ISC CDK
Asymmetric key generation of the TOE's REK (referred to as the "System" credential) used to protect TOE data FCS_CKM_EXT.1.1(2), FCS_RBG_EXT.1	CertAgent	PKCS#11 Cryptographic Module
Symmetric key generation of DEKs used to protect TOE data prior to storage in the database FCS_CKM_EXT.1.1(1) , FCS_RBG_EXT.1	CertAgent	ISC CDK
Symmetric encryption/decryption with DEKs to encrypt/decrypt TOE data prior to storage in the database FCS_COP.1(1)	CertAgent	ISC CDK
Encryption of DEKs using the public key portion of the TOE's asymmetric REK ("System" credential) FCS_RBG_EXT.1, FCS_CKM_EXT.8, FCS_CKM.2	CertAgent	ISC CDK
Decryption of DEKs using the private key portion of the TOE's asymmetric REK ("System" credential) FCS_CKM_EXT.8	CertAgent	PKCS#11 Cryptographic Module
Asymmetric key generation of issuer credentials FCS_CKM.1, FCS_RBG_EXT.1	CertAgent	PKCS#11 Cryptographic Module
Hashing of certificate request, certificate, CRL, and OCSP 'to be signed' data FCS_COP.1(3)	CertAgent	ISC CDK

Digital signature over certificate, CRL, and OCSP responses FCS_COP.1(2), FCS_RBG_EXT.1	CertAgent	PKCS#11 Cryptographic Module
Digital signature verification of certificate requests, certificates, and CRLs FCS_COP.1(2), FCS_CKM_EXT.5	CertAgent	ISC CDK
Digital signature over certificate requests for the initial set of authentication credentials FCS_COP.1(2), FCS_RBG_EXT.1	CertAgent	ISC CDK
Keyed-hash message authentication for secure storage of EST password check values FCS_COP.1(5)	CertAgent	ISC CDK

TABLE 18 CRYPTOGRAPHIC DUTIES

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). During installation the TOE prompts for this library and stores the library path and name for future use. As described below, the TOE creates a number of certificates and private keys using the PKCS#11 Cryptographic Module during installation. One of these keys is the "System" credential which is used to encrypt data needed by the TOE in order to start. When the TOE starts, it initializes the PKCS#11 API and logs into the module using the password supplied by the privileged user starting the TOE. If the TOE starts and the PKCS#11 Cryptographic Module is incorrect, the "System" credential will be missing and the TOE will fail to start. The TOE keeps the initial PKCS#11 login session open while running so that it doesn't need to call the login function repeatedly. The table below lists the PKCS#11 API calls used by the TOE.

Operation	PKCS#11 Functions
TLS/HTTPS session establishment (digital signature and asymmetric decryption required by TLS/HTTPS) FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1, FCS_COP.1(2), FCS_RBG_EXT.1	C_SignInit, C_Sign
Asymmetric key generation of TLS/HTTPS server credentials FCS_CKM.1, FCS_CKM.2, FCS_RBG_EXT.1	C_GenerateKeyPair
Asymmetric key generation of the TOE's REK (referred to as the "System" credential) used to protect TOE data FCS_CKM_EXT.1.1(2), FCS_RBG_EXT.1	C_GenerateKeyPair
Decryption of DEKs using the private key portion of the TOE's asymmetric REK ("System" credential) FCS_CKM_EXT.8	RSA: C_DecryptInit, C_Decrypt ECC: C_DeriveKey, C_DigestInit, C_DigestUpdate, C_DigestKey, C_DigestFinal
Asymmetric key generation of issuer credentials FCS_CKM.1, FCS_RBG_EXT.1	C_GenerateKeyPair
Digital signature over certificate, CRL, and OCSP responses	C_SignInit, C_Sign

FCS_COP.1(2), FCS_RBG_EXT.1	
Random number generation	C_GenerateRandom
Session management	C_OpenSession, C_GetSessionInfo, C_CloseSession, C_GetSlotList, C_GetTokenInfo, C_GetMechanismInfo, C_GetInfo
Authentication to the device	C_Login, C_Logout
Object management	C_FindObjectsInit, C_FindObjects, C_FindObjectsFinal, C_GetAttributeValue, C_SetAttributeValue, C_CreateObject
Startup/Shutdown	C_Initialize, C_Finalize

TABLE 19 PKCS#11 API FUNCTIONS

The TOE installer creates a number of certificates and private keys during installation to enable the system to start and allow the administrator to configure, and issue production certificates. To do this, the TOE installer first creates a self-signed root certificate (RSA-3072/SHA-384) using the PKCS#11 Cryptographic Module and then uses that root to issue the other certificates (TLS/HTTPS server, administrator, CA operations staff, and auditors) whose private keys are store in password protected files on the environment's local file system. These keys are generated by the TOE using the ISC CDK and are all RSA-3072/SHA-384 credentials.

During installation the TOE uses the PKCS#11 Cryptographic Module to generate an initial "System" credential (certificate and private key) which is self-signed. The "System" credential is an asymmetric REK used by the TOE to encrypt/decrypt DEKs that encrypt sensitive information the TOE stores in the database. This key is also RSA-3072/SHA-384. The data format, used by the TOE when encrypting the sensitive information, is the Cryptographic Message Syntax (CMS) as defined in RFC 5652. CMS is the basis of S/MIME, and provides a format for storing data encrypted using a traditional hybrid encryption scheme in which a symmetric DEK is used to encrypt the bulk of the data and the DEK is encrypted with an asymmetric KEK or REK.

The TOE uses the ISC CDK to create ephemeral ECDHE keys during TLS/HTTPS negotiation and when encrypting sensitive data if the "System" credential has been changed to an ECC key pair.

9.3.2 FCS_STG_EXT.1 Cryptographic Key Storage

The table below lists 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.)	PKCS#11 Cryptographic Module	Protected by the PKCS#11 Cryptographic Module.
TLS/HTTPS Client Key	Authentication	Web Browser key	Protected by the web browser.

(asymmetric)		store	
CA CMS DEKs	Data encryption	Stored with the encrypted data in the database	Encrypted using the CA's "System" credential's public key.

TABLE 20 CRYPTOGRAPHIC KEY STORAGE AND PROTECTION

During installation the initial set of TLS/HTTPS asymmetric client authentication keys and certificates (that correspond to the supported roles) are generated by the TOE and saved as password protected PKCS#12 files, whose private keys are encrypted using AES-256, to be imported into a web browser in order to access the TOE.

9.3.3 Cryptographic Validation Information

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.

Gemalto SafeNet USB HSM	ISC CDK Certificate
2403	3105

TABLE 21 FIPS 14-2 VALIDATION CERTIFICATES

Operation(s)	CAVP Algorithm List	CAVP Certificate(s)	NIST Standard(s)	SFR(s)
Generate Random	DRBG	428	SP800-90	FCS_RBG_EXT.1
Key Pair Generation Signature Generation	ECDSA	461, 464	FIPS 186-4	FCS_CKM.1(b), FCS_COP.1(2)(a)
Key Generation Signature Generation	RSA	1369, 1372	FIPS 186-4	FCS_CKM.1(a), FCS_COP.1(2)(a)

TABLE 22 GEMALTO SAFENET USB HSM CAVP CERTIFICATES

Operation(s)	CAVP Algorithm List	CAVP Certificate(s)	NIST Standard(s)	SFR(s)
Encrypt (CBC) Decrypt (CBC)	AES	4002 (CentOS 6.7) 4396 (Win 2012R2)	FIPS 197; SP800-38A	FCS_COP.1(1)
Generate Random	DRBG	1192 (CentOS 6.7) 1416 (Win 2012R2)	SP800-90	FCS_RBG_EXT.1
Key Pair Generation Signature Verification	ECDSA	892 (CentOS 6.7) 1055 (Win 2012R2)	FIPS 186-4	FCS_CKM.1(b), FCS_COP.1(2)(b)

PBKDF2 TLS PRF Generate Random	HMAC	2615 (CentOS 6.7) 2914 (Win 2012R2)	FIPS 198-1	FCS_COP.1(4)
ECDH (Ephemeral Unified)	KAS	85 (CentOS 6.7) 186 (Win 2012R2)	SP800-56A R3	FCS_CKM.2
Key Generation Signature Generation Signature Verification	RSA	2065 (CentOS 6.7) 2378 (Win 2012R2)	FIPS 186-4	FCS_CKM.1(a), FCS_COP.1(2)(b)
Hash Data	SHS	3307 (CentOS 6.7) 3625 (Win 2012R2)	FIPS 180-4	FCS_COP.1(3)

TABLE 23 ISC CDK CAVP CERTIFICATES

The ISC CDK FIPS validation was performed on platforms other than those used for this validation and separate algorithm tests were performed using the CAVP process to cover the missing platforms.

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

The TOE uses PBKDF2 as defined in RFC 2898 and NIST SP800-132 to create the check values for EST user passwords using HMAC-SHA-256, an 8-byte salt, and 2048 iterations. The 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 protect key shares per FPT_SKY_EXT.2; it 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[1].

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

The TOE generates DEKs of 256-bit length using the ISC CDK to encrypt sensitive data prior to storage in the database.

DEK	Information
CMS Symmetric Key	This key is generated by calling getrand2(32) method in the ISC CDK which returns a 32-byte (256-bit) random value from the ISC CDK's approved DRBG which is used directly as the key.

TABLE 24 DATA ENCRYPTION KEYS

[1] NIST SP 800-57 Part 1, Revision 4, states that the comparable security strength of HMAC-SHA-256 for key derivation functions is at least 256-bit

9.3.6 FCS_CKM.1 Cryptographic Key Generation, FCS_CKM.2 Cryptographic Key Establishment

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.

During installation the TOE uses the PKCS#11 Cryptographic Module to generate the initial TLS/HTTPS server key for HTTPS/TLS operations as described in the introduction.

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 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 post-installation using the TOE's Admin Site. The options presented when selecting the key type to generate are RSA-3072, RSA-4096, RSA-8192, NIST curve P-256, NIST curve P-384, or NIST curve P-521.

TABLE 25 KEY ESTABLISHMENT KEYS

The TOE uses the PKCS#11 Cryptographic Module to generate the initial set of credentials as described in the introduction to this section. The specific SFRs that use or rely on these keys are: FAU_GEN.2.1, FCO_NRO_EXT.2.1, FCO_NRO_EXT.2.2, FCS_COP.1.1(2), FIA_X509_EXT.2.1, FTP_ITC.1, FTP_TRP.1.1

Key	Information
HTTPS/TLS Server Credential	The TOE installer uses the PKCS#11 Cryptographic Module to create an RSA-3072 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). Each are RSA-3072.
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, RSA-4096, RSA-8192, 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, RSA-4096, RSA-8192, US-P-256, US-P-384, US-P-521.

TABLE 26 OTHER ASYMMETRIC KEYS

As the tables indicate the TOE supports RSA-4096 and larger key sizes but they were not tested for use in the evaluated configuration.

9.3.7 FCS_CKM_EXT.4 Cryptographic Key Destruction

The TSF destroys all cryptographic keys and critical security parameters as follows:

Key	Type	Information
CMS DEK	Symmetric	Resides as plaintext in memory only. Cleared by the ISC CDK when the operation completes.
TLS ECDHE	Asymmetric	Resides as plaintext in memory only. Cleared by the ISC CDK when the HTTPS/TLS operation requiring them completes.

TABLE 27 KEY DESTRUCTION

The TSF destroys all plaintext keying material in memory by clearing the memory. No plaintext keying material is kept outside of memory. Other data considered sensitive by the TOE is cleared as described in the following table.

Password	Information
PKCS#11 Cryptographic Module PIN for the “System” credential	This PIN is required to be injected by an administrator after system startup. The PIN is converted in to an object managed by the ISC CDK and the ISC CDK overwrites this memory with zeroes when the value is no longer needed.
Email password	Encrypted using “System” credential’s public key and stored in the database. If an email connection is required, the password will be decrypted and stored in memory. After closing the connection, the reference to the plaintext password will be destroyed and a Java call immediately made to request garbage collection.
Database password	Encrypted by “System” credential’s public key and stored in a local database configuration file. After the system PIN has been entered, the database password will be decrypted and stored in memory. At system shut down, the reference to plaintext password will be destroyed and a Java call immediately made to request garbage collection.
PKCS#11 Cryptographic Module PIN for CA issuer credentials	Encrypted by “System” credential’s public key and stored in the database. If a signing operation is required, and the PKCS#11 access information is different from that of the “System” credential, the PIN will be decrypted and stored in memory managed by the ISC CDK. When the operation completes the ISC CDK will overwrite this memory with zeroes.
EST subscriber password	A PBKDF2 (HMAC-SHA-256, 8-byte salt, 2048 iterations) generated check value is stored in the database. When an EST request uses a password for authentication its PBKDF2/HMAC-SHA-256 check value is computed and compared to that in the database. When the operation completes the reference to the plaintext EST password will be destroyed and a Java call immediately made to request garbage collection.

TABLE 28 SENSITIVE DATA DESTRUCTION

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 is simply clearing memory allocated by the environmental Operating System using APIs provided by the same.

9.3.8 FCS_CKM_EXT.5 Public Key Integrity

Public keys used by the TSF to meet CA requirements are protected against undetected modification through the use of digital signatures and by the Operational Environment. For those public keys protected by the TOE, a digital signature is used and it is verified upon each access to the public key.

The table below shows the public keys used by the TOE to meet CA requirements that are protected by the TOE or the Operational Environment.

Key	Storage	Protection	Integrity	Purpose
Issuer certificates	Database and PKCS#11 Cryptographic Module	Operational Environment	Operational Environment	Public signature key for certificate and CRL signing, and for signing OCSP responses
CertAgent trust anchor certificates	Database	Digital signature	Verification of digital signature	Certificate path validation
Apache Tomcat trust anchor certificates	Local file system	Operational Environment	Operational Environment	Certificate path validation
ACL certificates	Database	Digital signature	Verification of digital signature	Access control
HTTPS/TLS Server Key	PKCS#11 Cryptographic Module	Operational Environment	Operational Environment	Server authentication

TABLE 29 PUBLIC KEY INTEGRITY

Operational Environment Protection

Issuer certificates are protected by the PKCS#11 Cryptographic Module. The TOE maintains a copy of each issuer certificate in the environmental database. Access to the database is controlled by the environmental Operating System.

The HTTPS/TLS server certificate is protected by the PKCS#11 Cryptographic Module.

Access to the Apache Tomcat trust anchor key store is controlled by the environmental Operating System.

Signature Validation

Signatures in certificate requests are validated when:

- submitted via the Public site; the certificate request will be rejected if validation fails

- submitted via EST or RAMI; the operation will fail if the certificate request is invalid
- viewed in the Advanced Certificate Request dialog; an error message will be displayed and the issuance option will be disabled if validation fails
- issued; the action will be aborted if validation fails

Signatures in certificates issued by a CA account and stored by the TOE will be validated when:

- viewed in the Advanced Certificate dialog; an error message is displayed on failure
- parsed for display; an error message is displayed on failure

Signatures in issuer certificates will be validated when:

- assigned as a CA credential to a CA account; an error message is displayed and the operation aborted on failure
- used to issue a certificate; an error message is displayed and the operation aborted on failure
- used to issue a CRL; an error message is displayed and the operation aborted on failure
- used to create an OCSP response; an error message is displayed and the operation aborted on failure
- parsed for display; an error message is displayed and the operation aborted on failure

Signatures in trust anchor certificates will be validated when:

- added to the trust anchor list; an error message is displayed and the action aborted if validation failed
- parsed for display; an error message will be displayed if validation failed
- when used to validate a certificate path; on failure access using that trust anchor will be denied

The signatures over the ACL table is validated when:

- the TOE starts; the TOE will shut down if validation fails
- when requested by a privileged user through the TOE's web interface; an error message is displayed and the TOE shuts down on failure
- when the ACL is checked for membership; the TOE will shut down if validation fails

The signature over the CertAgent Trust Anchor table is validated when:

- the TOE starts; the TOE will shut down if validation fails
- when requested by a privileged user through the TOE's web interface; an error message is displayed and the TOE shuts down on failure
- when a certificate in the Trust Anchor database is used; the TOE will shut down if validation fails

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

The TOE performs encryption and decryption as follows:

- The TOE uses the ISC CDK to perform AES-CBC encryption/decrypt with 256-bit keys during TLS/HTTPS negotiation.
- The TOE uses the ISC CDK to perform AES-CBC encryption/decryption with 256-bit keys when storing/retrieving sensitive data in the database. The encrypted data is in the CMS format (RFC 5652). The AES DEK is encrypted using asymmetric encryption with the "System" credential's public key.

The ISC CDK's API includes a C++ object for AES operations and, on all platforms, the TOE uses the `init` and `crypt` methods in that object for AES encryption/decryption. The `init` function takes as input the type of operation (encrypt or decrypt), the size of the key in bytes (32), the key's value as an array of bytes, a mode of operation (CBC), and an initialization vector as an array of bytes.

9.3.10 FCS_COP.1(2) Cryptographic Operation (Cryptographic Signature)

The TOE invokes the PKCS#11 Cryptographic Module to perform signature operations (issuing certificates, creating CRLs, or creating OCSP responses). The TOE uses the ISC CDK to digitally sign RSA certificate requests as part of the installation process that generates the initial set of authentication credentials. The key sizes supported are those listed in Table 17 Section 9.2. The table below shows the details.

Object Signed	Cryptographic Module/API	Details	Function Invoked (on all platforms, for all supported algorithms, modes, and key sizes)[2]
Certificate request	PKCS#11 Cryptographic Module	The TOE creates certificate requests for cross-certificate signing and subordinate issuer creation. The requests are signed using the PKCS#11 Cryptographic Module.	C_SignInit, C_Sign
Certificate request	ISC CDK	The TOE creates the initial set of authentication credentials as RSA certificate requests which are signed by an initial root certificate. The requests are signed by the private key generated by the ISC CDK.	Key::Sign
Issued certificate	PKCS#11 Cryptographic Module	The TOE signs certificates using the PKCS#11 Cryptographic Module.	C_SignInit, C_Sign
CRL	PKCS#11 Cryptographic Module	The TOE signs CRLs using the PKCS#11 Cryptographic Module.	C_SignInit, C_Sign
OCSP response	PKCS#11 Cryptographic Module	The TOE signs OCSP responses using the PKCS#11 Cryptographic Module.	C_SignInit, C_Sign
Root certificate	PKCS#11 Cryptographic Module	The TOE signs root certificates using the PKCS#11 Cryptographic Module.	C_SignInit, C_Sign
"System" certificate	PKCS#11 Cryptographic Module	The TOE signs the "System" certificate using the PKCS#11 Cryptographic Module.	C_SignInit, C_Sign
TLS/HTTPS ServerKeyExchange message	PKCS#11 Cryptographic Module	The TLS/HTTPS ciphersuite uses ECDHE for key agreement and the TOE uses the PKCS#11 Cryptographic Module to sign a hash value as part of the TLS negotiation process.	C_SignInit, C_Sign
TLS/HTTPS client authentication message verification	ISC CDK	If the TLS/HTTPS connection requires client authentication, the signature sent by the client is verified by the TOE using the ISC CDK.	For certificates containing an RSA public key: RSA::encrypt For certificates containing an ECC public key: Key::SignCheck

[2] 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.

Trust Anchor and ACL tables	PKCS#11 Cryptographic Module	The TOE uses the "System" credential to sign the serialized contents of the trust anchor and ACL database tables using the PKCS#11 Cryptographic Module.	C_SignInit, C_Sign

TABLE 30 SIGNED OBJECTS

Signature verification is detailed in Section 9.3.8 for those keys that the TOE stores. The TOE also validates signatures on certificates presented for client authentication purposes in HTTPS/TLS connections, and used to digitally sign trusted updates. To verify any digital signature, the TOE uses the ISC CDK and either the `RSA::encrypt` or `Key::SignCheck` function depending on the key type.

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

The TOE uses the ISC CDK to perform cryptographic hashing as follows:

- When establishing a TLS/HTTPS connection
  - Supports the TLS 1.1 PRF and message digest size 288-bits
  - Supports SHA-1 for signature validation as required by TLS 1.1
- When digitally signing the Trust Anchor and ACL database tables for integrity protection.
  - Uses SHA-384
- When creating certificates, certificate requests, and CRLs
  - Supports SHA-384 and SHA-512 and message digest size 384 and 512
- When creating OCSP responses
  - Supports SHA-1, SHA-256, SHA-384, SHA-512 and message digest sizes 160, 256, 384, and 512
- When verifying certificates, CRLs, and certificate requests
  - Supports SHA-1, SHA-256, SHA-384, SHA-512 and message digest sizes 160, 256, 384, and 512

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

The TOE uses the ISC CDK to perform key hash message authentication as follows:

- When establishing a TLS/HTTPS connection
  - Supports the TLS 1.1 PRF (which uses HMAC-SHA-1 for parts of the computation) with key size 256-521-bits and output size of 288-bits

- When performing PBKDF2 for creating the EST password check values (Note: PBKDF2 is performed when a privileged user in the CA Operations Staff role creates the password or when a subscriber authenticates with the password)
    - Uses HMAC-SHA-256 with key size 120-800-bits and output size of 256-bits
- When generating random numbers
    - Uses HMAC-SHA-256 with key size 256-bits and output size of 256-bits

9.3.13 FCS_RBG_EXT.1 Cryptographic Random Bit Generation

The TOE uses the two cryptographic modules to generate random numbers in the following ways.

- The PKCS#11 Cryptographic Module
    - CTR_DRBG(AES-256)
    - Provider: Gemalto SafeNet USB HSM
    - Entropy Source: hardware-based noise source providing 384-bits of entropy
- The TOE using the ISC CDK
    - HMAC_DRBG(SHA-256)
    - Provider: ISC CDK
    - Entropy Source: third-party software-based noise source providing more than 256-bits of entropy

The entropy document explains, in detail, the sources and amounts of entropy. As stated in the entropy assay, the TOE provides additional entropy to the ISC CDK DRBG using the PKCS#11 Cryptographic Module's `C_GenerateRandom` function.

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

The TOE provides TLS/HTTPS protected access to its external interfaces and remote IT entities.

The following ciphersuites are supported:

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

The default, initial, TLS/HTTPS server key is RSA-3072. The TLS/HTTPS server key can be replaced with a RSA-4096, or larger, key pair if so desired, but this was not tested for use in the evaluated configuration. For TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA, 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.

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

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.

See Section 9.9 for additional details of the TLS/HTTPS trusted channels/trusted path provided by the TOE.

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

As described elsewhere, the TOE uses the PKCS#11 Cryptographic Module to generate an asymmetric key pair (certificate and private key) that is designated as the "System" credential. There is only one (1) active "System" credential at a time (see Section 9.3.16). When the TOE uses the PKCS#11 Cryptographic Module, the PKCS#11 Cryptographic Module uses its random number generation routines to generate the random values required for key generation. To our knowledge, the PKCS#11 Cryptographic Module used in the evaluated configuration generates the key randomly and its DRBG provides 256-bits of security. In the evaluated configuration, the default "System" credential is an RSA 3072-bit asymmetric key, but can be changed using the TOE's Admin Site to another key type as listed in Table 17 and described in Section 9.2. The "System" credential's public key is used to encrypt DEKs which are used to protect sensitive data stored in the database.

Please see Section 9.3.1 and Table 19 for details on which functions in the PKCS#11 API are used to generate these keys.

Please see Section 9.3.16 for details on how the size of this key is identical to the key size used for the encryption/decryption of data by this key.

9.3.16 FCS_CKM_EXT.8 Key Hierarchy Entropy

The TOE maintains a single REK called the "System" credential. As stated in Section 9.3.1, the "System" credential is used to encrypt DEKs. The "System" credential is an asymmetric REK and thus the encryption of a DEK is traceable through a hierarchy to a REK. The REK is generated as described in Section 9.3.15.

DEKs are created when the HSM password or database password are encrypted by the TOE (i.e., during installation or later if the HSM password or database password are updated using the TOE's interfaces). Encrypted DEKs are stored in the database and used when the TOE starts, or the first time the TOE requires access to the encrypted data.

As stated in Section 9.3.5, the DEK is a 256-bit key generated randomly using the ISC CDK's DRBG's `getrand2()` method, with an input parameter of 32 requesting 32 random bytes (i.e. 256-bits). Thus the REK must provide the equivalent of 256-bit security. In the evaluated configuration, a RSA 3072-bit key is used which matches the Commercial National Security Algorithm Suite recommendation for the protection of 256-bit AES keys.

9.4 User Data Protection (FDP)

9.4.1 FDP_CER_EXT.1 Certificate Profiles

The TOE implements a certificate profile function and issues certificates consistent with the profile's configuration. A CA account can have one or more profiles which are configured by an administrator using the TOE's web interface. Using that interface the administrator can assign a name (account ID), access control list, extensions, extensionRequest filter rule, and default properties to the profile.

The default certificate properties include RDN, validity period, message digest, and DN encoding format. Extensions include subjectKeyIdentifier, authorityKeyIdentifier, basicConstraints, keyUsage, extendedKeyUsage, certificatePolicy, etc. The extensionRequest filter rule allows administrator to control how requested extensions that appear in certificate requests are handled. The extensionRequest filter controls, on a per extension basis:

- if the extension request will be allowed to override the profile value
- if the extension request will be ignored
- if the presence of the extension in the request will cause the request to be rejected
- if the presence of the extension in the request will flag it for manual issuance

For keyUsage and extendedKeyUsage extensions, there is a special "required and matched" option. If this option is specified, and the certificate request doesn't contain precisely the same extension and value as the profile to which it is submitted, the certificate request will be rejected.

Subscribers select the profile to submit their request to through

- the TOE's publicly accessible web enrollment pages
- the EST URL which includes the profile's name
- the RAMI interface's POST parameters

The TOE does not support or allow null names. The TOE can generate certificates that comply with the IETF RFC 5280 profile and can ensure that the requirements listed in Section 6.6.1 are met. Specifically:

- The TOE issues only X.509 v3 certificates whose version field contains the integer 2.

- The TOE never populates the issuerUniqueID or subjectUniqueID fields.
- The TOE uses a sequential serial number with 3 random leading bytes ensuring that the serialNumber is unqiue for a given issuer.
- The TOE will not issue a certificate whose notBefore value precedes the current time or a certificate whose notAfter value precedes the notBefore value.
- The TOE never issues a certificate with an empty issuer field.
- The TOE only allows the use of the key types and sizes selected in FCS_COP.1(2) and always populates the signature field and the algorithm in the subjectPublicKeyInfo field using an OID for one of the selected algorithms. The allowed key types and sizes are: RSA with a key size greater than or equal to 3072-bits, ECDSA with a key size 256-bits or greater and NIST curve type P-256, P-384, or P-521.
- The TOE supports a number of extensions including: subjectKeyIdentifier, authorityKeyIdentifier, basicConstraints, keyUsage, extendedKeyUsage, and certificatePolicy.
- The TOE never issues a certificate with a subject field containing a null name that isn't accompanied by a populated critical subjectAltName extension because the TOE requires the subject field to contain a non-null name. The TOE requires that the Name contain at least one relative distinguished name.
- When the profile includes a subjectKeyIdentifier extension it is populated according to RFC 5280, section 4.2.1.2, option 1: "The keyIdentifier is composed of the 160-bit SHA-1 hash of the value of the BIT STRING subjectPublicKey (excluding the tag, length, and number of unused bits)."
- When the profile includes a authorityKeyIdentifier extension it is populated with the subjectKeyIdentifier value from the certificate being used to issue the certificate.
- The TOE requires that keyUsage and extendedKeyUsage values in an issued certificate contain consistent values. The TOE will not allow a profile to be created or updated if it contains inconsistent values for keyUsage and extendedKeyUsage. If the profile's extensionRequest filter is configured to allow a certificate request to override the profile's keyUsage and/or extendedKeyUsage setting, the TOE will reject requests with inconsistent values.

Proof-of-possession of the private key corresponding to the request is addressed in Section 9.2.

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.

9.4.2 FDP_CER_EXT.2 Certificate Request Matching

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.

9.4.3 FDP_CER_EXT.3 Certificate Issuance Approval

The 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' permission can approve certificates via the CA account web interface (which is the only interface through which manual issuance occurs).

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.

9.4.4 FDP_CSI_EXT.1 Certificate Status Information

The TOE provides certificate status information whose format complies with ITU-T Recommendation X.509v2 CRL, and the OCSP standard as defined by IETF RFC 6960.

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. Please see Section 9.4.7 for additional details.

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.

Privileged users with 'CA Operations Staff' role and 'revoke' or 'RAMI' permission, or subscribers, can approve changes to the status of a certificate.

This can be done via:

- the TOE's CA account web interface
- the Registration Authority Management Interface (RAMI)
- The TOE's public web interface's subscriber self-management web interface

The process varies based on the interface. Within either web interface, a list of valid certificates is presented along with a button labeled Revoke. The rule for subscribers, is that they 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. Clicking the button, and confirming the action in the pop up, changes the status of the certificate. Via RAMI, submitting a revoke request will change the status of the certificate.

9.4.5 FDP_RIP.1 Subset Residual Information Protection

The TOE does not store any personally identifiable information, that does not also appear in a certificate. The TOE does handle EST passwords and the TLS session object.

EST passwords enter the TOE via the web interface or the EST interface. For either path, the EST password is immediately converted into a check value and the memory related to them cleared and freed with possible garbage collection at some point. The conversion process passes the value into the ISC CDK which performs PBKDF2 (as described in Section 9.3.12) and returns the check value. The CDK's buffers are zeroized when the check value is returned.

1. A EST password is entered into the TOE.
2. The EST password is passed to the ISC CDK.
3. The ISC CDK computes the check value, returns the check value, and clears the CDK buffers.
4. The TOE's Java servlet erases the EST password from its memory and frees the related object.
5. Sometime later the Operational Environment's Java Runtime Environment executes its garbage collection routines and frees the memory back to the Operational Environment's Operating System.
6. The Operational Environment's Operating System clears each memory page before allocating it to a process.

Memory holding EST password values is cleared prior to deallocation by the TOE and prior to allocation by the Operating Environment.

TLS session objects are a Java objects, the primary members which store data that should not be reused (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.

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.

9.4.6 FDP_STG_EXT.1 Public Key Protection

The TSF implements an integrity mechanism to protect some of the trusted public keys and certificates used to validate trusted channel. Other public keys are protected by the Operational Environment which provides access controlled storage.

The CertAgent trust anchor list and various ACLs are maintained in the environmental database in tables that store the subject DN, the DER-encoded X.509 certificate, and, for ACLs, a set of permissions, for each certificate on the list. 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 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.

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).

The TOE's command line interface tool allows the trust anchor list, the Admin Site ACL, CA Account Sites' ACLs, and profile ACLs to be modified. Access to the TOE's command line tool is controlled by the environmental Operating System. Once an administrator has logged into the Operating System using their username and password, they can use the TOE's command line tool to modify the lists. Access to the environmental database, and its storage (files backing the database), is also controlled by the environmental Operating System.

To add or remove a certificate using the TOE's Admin Site, an administrator uses a web browser to connect to the TOE using the trusted path (a client authenticated TLS connection). Once connected, the TOE checks the certificate used to authenticate to determine on which ACL it appears, if any, and which permissions, if any, it has been granted. If their certificate is on the ACL for the site they are accessing, and has been assigned administrator permissions, they can then use the web interface to access the trust anchor list or ACL to be modified by clicking the appropriate link. The TOE then displays the current list with check boxes allowing certificates to be selected for deletion. At the bottom of the page there is an Add button allowing the privileged user to upload a X.509 v3 certificate to be added to the list.

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. As described in Section 2.4.6, only privileged users in the administrator role within the TOE should hold administrator rights in the environmental Operating System. 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.

The TOE component, Apache Tomcat, has its own trust anchor key store which is used as described in Section 9.9. Access to the Apache Tomcat trust anchor key store is controlled by the environmental Operation System as described in Section 9.3.8. Too add or remove a certificate from the Apache Tomcat, 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.

As described elsewhere (see Section 9.3.8), the environmental PKCS#11 Cryptographic Module stores, protects, and maintains the integrity of the issuer, “System”, and TLS server credentials, including their public keys, used by the TOE.

9.4.7 FDP_CRL_EXT.1 Certificate Revocation List Validation

The TOE supports CRL generation on demand, on schedule, or when certain revocation reasons are used depending on its configuration. Issued CRLs contain values in accordance with ITU-T Recommendation X.509 as follows:

a) The version field is always present and contains a 1.

b) The authorityKeyIdentifier and CRLNumber extensions are supported.

c) The issuer field is always present and never contains a null name.

d) The signature and signatureAlgorithm fields contain the OID for a digital signature algorithm in accordance with FCS_COP.1(2). Consistent with this ST, which states that only SHA-384 or SHA-512 will be used when issuing CRLs the supported OIDs are: ecdsa-with-SHA384 (1.2.840.10045.4.3.3), ecdsa-with-SHA512 (1.2.840.10045.4.3.4), sha384WithRSAEncryption (1.2.840.113549.1.1.12), sha512WithRSAEncryption (1.2.840.113549.1.1.13).

e) The thisUpdate field indicates the issue date of the CRL.

f) The nextUpdate field is always present and the time specified in this field does not precede the time specified in the thisUpdate field

Using the TOE’s CA Account site a member of the CA Operations Staff role of a CA Account can configure CRLs to be automatically issued:

- Periodically based on the number of hours, days, months, or years specified.
- Whenever a certificate hold is removed.
- Whenever a certificate is revoked for a specified revocation reason code by CA Operations Staff or via RAMI. The reason codes that can be selected are:
    - On hold: None, Call Issuer, Reject, Pick-up Token
    - Revoked: Unspecified, Key Compromise, CA Compromise, Affiliation Changed, Superseded, Cessation of Operation, Remove from CRL, Privilege Withdrawn, AA Compromise

The CA Administrator can configure the CA Account to limit the number of issued CRLs retained and to use a particular message digest algorithm (SHA-384 or SHA-512) when issuing a CRL.

The TOE runs a daily job for each CA Account to change the status of revoked certificates to expired and revoked when the certificates expire. The next CRL issued will no longer include those certificate

serial numbers on the revocation list. This Admin Site Administrator can configure the settings associated with this job.

9.4.8 FDP_OCSPG_EXT.1 OCSP Basic Response Generation

The TOE produces OCSP basic responses as described in IETF RFC 6960. The OCSP responses are signed by the CA's issuer private key which resides in the PKCS#11 Cryptographic Module. SHA-1, SHA-256, SHA-384 and SHA-512 are supported and can be configured via the TOE's CA Account web interface by an administrator.

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.

The TOE's OCSP Responder is limited. Administrators can enable or disable it by checking or unchecking the Enable OCSP Responder checkbox for the CA account. Administrators may also select the hash algorithm that is used when signing the responses. If the Enable OCSP Responder checkbox is checked, the TOE's OCSP Responder will generate OCSP responses for the issuer. If the Enable OCSP Responder checkbox is not check, the TOE's OCSP Responder will not generate OCSP responses for the issuer. Since the TOE allows more than one issuer per instance of the TOE, this allows an administrator to control which issuers on the TOE provide OCSP services and which do not.

9.5 Identification and Authentication (FIA)

9.5.1 FIA_X509_EXT.1 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.

Note, the TOE never acts as a TLS client and so never validates a TLS server certificate to verify that it contains the Server Authentication purpose 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.

9.5.2 FIA_X509_EXT.2 Certificate-Based Authentication

The TOE uses X.509v3 certificates, as defined by RFC 5280, to authenticate privileged users, subscribers, RAs, and DBAccess clients over HTTPS, and to verify the integrity of software updates. When the TOE cannot determine the validity of a certificate the TOE will not accept the certificate. If the certificate used for the software update is not successfully validated, the TOE refuses to allow the update to install. 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.

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. Please see the TOE's guidance documents for details on replacing the TLS credential.

9.5.3 FIA_UAU_EXT.1 Authentication Mechanism

The TOE uses certificate-based access control to perform privileged user authentication through its web interfaces. Privileged users are those that have a role in the TOE: Administrators, Auditors, and CA Operations Staff. Subscribers and relying parties are not privileged users and are discussed in the next section.

A privileged user (as well as an RA or DBAccess client) is authorized to act in a role, if they hold the private key matching a certificate and permission on one of the TOE's ACLs when they authenticate to the TOE. If the client successfully establishes a client authenticated HTTPS connection with the TOE, the TOE validates the client's certificate as described in Section 9.5.1 and checks the TOEs ACLs to determine what access the user should be granted. The ACLs are tables in the database consisting of lists of certificates and the rights granted to the holder of the private key matching that certificate as proved when they successfully perform the digital signature required by the TLS negotiation process. A certificate can be added to the ACL as described in Section 9.4.6. That section lists no restrictions on the origin of the certificates and they may be issued by a CA other than the TOE.

There is one ACL table in the database and it is created during installation of the TOE. During installation the table is populated with the initial set of authentication credentials created by the installer as described in Sections 2.4.3, 9.3.2, 9.3.6, and 9.3.10. The auditor's credential is assigned auditor permissions and the administrator's credential is assigned admin rights on the Admin Site's

ACL. Each time a CA account is created, an empty ACL for that account is also created in the ACL table. Each time a CA profile is created, an ACL of that profile is also created in the ACL table with the same certificates and permissions as the CA account ACL. During installation, a CA account is created (it is used to issue the TLS server credential and the initial set of authentication credentials). The ACL for this CA is populated by the installer, with the auditor's, administrator's and CA operation staff's certificates each holding the matching permissions for their role.

Local Administrators of the environmental operating system, that log in to the local console using a username and password, can manage the TOE locally (using the TOE's command line tools) and perform updates to the TOE. See Section 9.6 for details on the TOE interfaces available from the environmental Operating System's console. The authentication mechanism of local administrators is controlled by the environmental operating system using a username and password mechanism.

9.5.4 FIA_UIA_EXT.1 User Identification and Authentication

The TSF allows 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. There are no other actions that may be taken by relying parties.

Privileged Users, RAs, and DBAccess Clients

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. For details on the actions, and by whom these actions can be performed, see Section 9.6.

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. This process is further described in Section 9.5.3.

The TOE's EST implementation supports client authorization defined in section 3.7 of RFC 7030 that states that the client is a Registration Authority if the client authentication certificate used was issued by the EST CA, and it includes the id-kp-cmcRA OID in its extendedKeyUsage extension. In this case the EST client is treated as an RA, not a subscriber, and the TOE allows the issuance of a certificate whose Subject DN does not match that of the client certificate used to authenticate. With the exception of the fourth bullet above, the same requirements apply to this client certificate.

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.

Subscribers

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 self-service revocation portion of the Public site, and EST enrollment, require subscribers to successfully authenticate to the TOE using TLS/HTTPS with client authentication using their valid certificate issued in a manner matching that of privileged users. The only difference is that rather than checking the supplied certificate against an ACL, the Subject DN of the certificate is used to determine whether or not to allow the request. If the Subject DN of the certificate matches that of the certificate requested through EST then the renewal operation is allowed. For self-revocation,

only certificates matching the Subject DN in the certificate used to authenticate are presented to the subscriber for self-revocation.

Subscribers without a valid certificate that want to subscribe using EST can authenticate using an EST subscriber name and password created by a CA Operations Staff member of the CA account for them. They are only able to obtain a certificate that contains the subscriber name in its Subject DN or Subject Alternative Name extension.

Relying Parties

As noted above, relying parties are never authenticated or identified. Relying parties may use the TOE's Public site to obtain certificate status information, by downloading CRLs, or certificates, by downloading root certificates, issuer certificates, or subscriber certificates. Relying parties may also use the OCSP interface to obtain certificate status information.

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

The TOE supports Enrollment over Secure Transport (EST) protocol as described 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 (see Section 9.5.4).

The cipher suite supported is TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492.

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. EST passwords must be at least 15 characters in length and be composed of any combination of upper and lower case letters, numbers, and at least one of the following special characters: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")". EST passwords are converted into check values using the mechanisms described in Section 9.3.12 and the check values are stored in the database table for the CA account.

When a subscriber connects to the TOE using EST basic authentication, it passes the subscriber name and password to the TOE. The TOE then computes the check value for the presented password and compares it to the value in the database for the supplied subscriber name. If the values match, the subscriber name used matches one of the fields below, the profile allows EST, the username and password haven't already been used to obtain a certificate, and all other profile compliance checks pass, a certificate will be issued to the entity automatically. The subscriber name used to authenticate is compared to the following values in the certificate request submitted:

- Common Name (CN) in the request's Subject DN
- Email address in the request's subjectAltName
- DNSname in the request's subjectAltName

If the subscriber name matches one of those items, the TOE will issue a certificate matching the request.

9.5.6 FIA_X509_EXT.3 Certificate Request, FIA_ENR_EXT.1 Certificate Enrollment

The TOE supports the generation of a PKCS#10 certificate request as specified by RFC 2986 when establishing an issuer or when cross-certification with another issuer is desired. The generated request includes both the public key and the CA's distinguished name.

When establishing an issuer, a user in the Administrator role selects PKCS#10 certificate request as the desired type, obtains the request, and submits it to the other issuer. Once issued, the response is imported into the TOE which validates the chain of certificates up to a root certificate in the trust anchor list.

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.

9.6 Security Management (FMT)

The TOE has two administrator web sites, each with its set of roles and access control list. The CertAgent Administrative webpages, known as the Admin Site, support the following roles and responsibilities:

Role	Permission	Responsibility
Administrator	admin	manage "System" credential, database configuration settings, manage CA accounts, manage ACLs, trust anchor database, CRL store for path validations, NIAP configuration, run integrity tests, configure audit trails and manage jobs, submit queries via the DBAccess service
Auditor	audit	view and export audit trails; submit queries via the DBAccess service

TABLE 31 ADMIN SITE ROLES AND PERMISSIONS

The CertAgent CA Account webpages, known as the CA Account Site, support the following roles and responsibilities:

Role	Permission	Responsibility
Administrator	admin	manage account configurations; manage issuer credential, submit queries via the DBAccess service
Auditor	audit	view and export audit trails and search certificates; submit queries via the DBAccess service
CA Operations Staff	certify	issue certificates and reject invalid certificate requests
	revoke	revoke certificates and issue CRLs
	RAMI	submit requests via the RA management interface (RAMI)
	DBAccess	submit queries via the DBAccess service

TABLE 32 CA ACCOUNT SITE ROLES AND PERMISSIONS

Besides remote web interfaces (which can, of course, be accessed using a local web browser), the TOE supports local interface via command line tools. These tools are used by the TOE administrator. TOE administrators are assumed to have administrator privileges to login to the Operating System. The table below describes the tools, location, and their functionalities:

Tool and Location	Function
CertAgent script : certagent.[bat|sh]	start/stop the CA, set the system PIN, display version information
Tomcat script: tomcat.[bat|sh]	start/stop Tomcat

HyperSQL script: hsqldb.[bat|sh]	start/stop HyperSQL database server
Update tool: update/update.[bat|sh]	check for an update, verify an update, or install an update package
CACLI tool: tools/cacli/cacli.[bat|sh]	create CA accounts, create profiles, generate and manage CA credentials, create and disable CA accounts, display CA account names, display and update CA account configuration, display slot/label on an HSM, ACL management for Admin Site and CA Account Sites, trust anchor management, CRL management, profile management, import issued CRL/certs, export certs, and submit cert requests
Report generator: tools/reportgenerator.[bat|sh]	generate reports on certificates matching search criteria

TABLE 33 LOCAL TOOLS AND FUNCTIONS

DBAccess supports the following roles and responsibilities:

Role	Permission	Responsibility
Administrator	admin	query the list of CA account names query the distinguished name of a given CA account's issuer certificate
Auditor	audit	query the last X number of the Admin Site's audit records query the last X number of a CA Account's audit records
CA Operations Staff	DBAccess	query a certificate table index[3] create a certificate table index delete a certificate table index query the certificate table update the point of contact information for a certificate table entry

TABLE 34 DBACCESS ROLES AND PERMISSIONS

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

The environmental Operating System restricts the following tasks to the local administrator:

1. manage the TOE locally using the local tools and functions listed in Table 33;
2. perform updates to the TOE using the update tool listed in Table 33;

The TOE restricts the following tasks, performed through the TOE web interface Admin Site, to the Administrator role:

1. manage the TOE locally and remotely;
2. configure the audit mechanism;

[3] A database table "index" enables a query to more efficiently retrieve data from a database table.

3. perform on-demand integrity tests;
4. import and remove X.509v3 certificates into/from the Trust Anchor Database;
5. manage the ACL of the Admin Site and CA Account Site;
6. manage the CRL store for path validation;
7. configure the default TOE access banner;
8. disable CA accounts;

The TOE restricts the following tasks, performed through the TOE web interface CA Account Site, to Administrator role:

1. configure and manage certificate profiles;
2. modify revocation configuration;
3. configure certificate revocation list function;
4. configure OCSP function;
5. export PKCS#10 certificate request;
6. import CA certificate;
7. generate certificate request for issuer;

Both the Admin Site and the CA Account Site maintain their own access control lists (ACL) containing authorized privileged user certificates and permissions. Only users with the appropriate permissions can execute the defined functions. The role restrictions can be disabled, allowing one certificate to appear on all access control lists and have rights, by using the options on the NIAP conformance configuration web page.

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

The TOE restricts the following tasks to CA Operations Staff with 'certify' permission of the given CA Account Site:

1. approve and execute the issuance of certificates;
2. configure subscriber self-service request constraints;
3. configure automated certificate approval management;

The TOE restricts the following tasks to CA Operations Staff with 'RAMI' permission of the given CA Account Site:

1. approve and execute the issuance of certificates;

The TOE restricts the following tasks to CA Operations Staff with 'revoke' permission of the given CA Account Site:

1. configure subscriber self-service request constraints;

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

The TOE restricts the following tasks to CA Operations Staff with 'revoke' permission of the given CA Account Site:

1. approve certificate revocation; including the ability to configure automatic CRL issuance as described in Section 9.4.7

The TOE restricts the following tasks to CA Operations Staff with 'RAMI' permission of the given CA Account Site:

1. approve certificate revocation

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

The Operational Environment restricts the following tasks to the local administrator:

1. perform destruction of sensitive data when no longer needed;
2. participate as a second party for archival and recovery;
3. perform private or secret key or critical data export.

All three of these tasks operate on the PKCS#11 Cryptographic Module. Access to the module is controlled by the environmental operating system, which provides protection of TOE services and assets, under the control of the OS, from unauthorized access (OE.TRUSTED_ADMIN), and the PKCS#11 Cryptographic Module itself (via its own authentication mechanisms).

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.

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

The TOE restricts the following tasks to Auditors with access to the given site:

1. Delete entries from the audit trail
2. Search the audit trail

The TOE provides no way to delete individual entries from the audit trail to any role.

9.6.6 FMT_MTD.1 Management of TSF Data

None of the administrative functions listed in the section above 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.

9.6.7 FMT_SMF.1 Specification of Management Functions

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 described in Table 33.
    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 described in Table 33.
3. Ability to perform archival and recovery;
    a. The PKCS#11 Cryptographic Device (Gemalto SafeNet USB 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 Gematlo SafeNet USB HSM to the computer and configure it per HSM vendor guidance.
        iii. Use the Gemalto SafeNet 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 described in Table 33.

    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 Gemalto SafeNet USB HSM:

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

        ii. Use the Gemalto SafeNet 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 described in Table 33.

    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 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 1.1 and the only supported ciphersuite is TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA.

9.6.8 FMT_SMR.2 Restrictions on Security Roles

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[4]
Auditor	audit	Admin Site
		CA Account Site
		DBAccess service access[5]
CA Operations Staff	certify	CA Account Site
	revoke	CA Account Site
	RAMI	RAMI Interface
	DBAccess	DBAccess service access

TABLE 35 ROLE RESTRICTIONS

A user takes on the Administrator role when granted admin permission for any of the interfaces listed for the admin permission. A user takes on the Auditor role when granted audit permission for any of interface listed for the audit permission. A user takes on the CA Operations Staff role when granted permission for any of the four interfaces listed. The TOE associates users with roles by uploading users' certificates into the access control list and assigning permissions to the certificates as described in Sections 9.5.3 and 9.5.4.

The TOE's Admin Site, CA Account Site, RAMI, and DBAccess interfaces restrict available operations based on permissions granted to the certificate used to authenticate to the interface. For example, an Administrator accessing the CA Account Site is presented a different menu than a member of the CA Operations Staff and, for each operation attempted, a permission check is performed (in case direct URL is used). Similarly, an Administrator of the Admin Site who accesses the DBAccess interface is only allowed to successfully execute the queries listed in Table 33 associated with the admin permission, while a member of CA Operations Staff with DBAccess permission is restricted to successfully executing the operations listed in Table 33 associated with the DBAccess permission.

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.

[4] Implicitly granted by the admin permission

[5] Implicitly granted by the auditor permission

- 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.

Recall that each issuer (root or intermediate) has its own CA Account within the TOE. There is an ACL for the Admin Site, and a separate ACL for each CA Account. Within any of these ACLs, a single certificate cannot be given permissions that conflict. Within the Admin Site, the TOE refuses to allow a single certificate to be granted both admin and audit permissions to the Admin Site. Within a CA Account, the TOE refuses to allow a single certificate to be granted permissions for multiple roles. Within a CA Account, a certificate can have: just admin permission, just audit permission, or one or more of certify, revoke, RAMI, and DBAccess. However, within the TOE itself, a single certificate can be granted conflicting permissions across multiple CA Accounts. In other words, the role restrictions apply within a given account but not within the entirety of the TOE. 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)

TABLE 36 ROLE RESTRICTION EXAMPLE

The Administrator role can perform most administrative tasks through either the local console or through the Admin Site (provided an appropriate authentication certificate is available to the web browser used to access the Admin Site).

9.7 Protection of the TSF (FPT)

9.7.1 FPT_FLS.1 Failure with Preservation of Secure State

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.

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

TABLE 37 FAILURE STATES

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.

The TOE detects failures in the Operational Environment by checking the status indicator of the PKCS#11 Cryptographic Module (API function return codes) and by catching exceptions thrown by the JDBC API used to communicate with the database. When the TOE detects a failure in the Operational Environment that can be corrected while the TOE is still running, it creates audit records and returns an error message. If the TOE detects a failure in the Operational Environment that cannot be corrected while running, it shuts itself shown in an orderly fashion as described above.

9.7.2 FPT_KST_EXT.1 No Plaintext Key Export

All keys are listed in 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.

9.7.3 FPT_KST_EXT.2 TSF Key Protection

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 through the TOE, 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 further restrict privileged user actions based on roles as described in Section 9.5.

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. 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.

9.7.4 FPT_RCV.1 Manual Trusted Recovery

After a failure of integrity is detected, the TOE shuts itself down in an orderly manner. To return the TOE to a secure state:

- An administrator must start the TOE in maintenance mode by logging into the environmental Operating System and starting the TOE with the `start-maintenance` option.
- An administrator must login to the Admin Site and, for each CA account, remove and reimport all certificates in the ACLs or confirm (by certificate fingerprint information) that the certificates are correct.
- An administrator must login to the CA Account site and, for each profile in each account, remove and reimport all certificates in the ACLs or confirm (by certificate fingerprint information) that the certificates are correct.
- An administrator must login to the Admin Site, remove and reimport (or confirm, by certificate fingerprint information) all certificates in the trust list and ACLs and enable the NIAP options disabled in maintenance mode. Enabling the NIAP options causes the signatures over the trust list and ACLs to be computed and stored restoring the integrity of the system.
- An administrator must log into the environmental Operating System, stop the TOE using the `stop` option, and then start the TOE with the `start` option.

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 issue(s) as described above. 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

9.7.5 FPT_SKP_EXT.1 Protection of Keys

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.

See Section 9.3.2 for details on each key and its protection. All keys are controlled by FIPS 140-2 validated cryptographic modules.

9.7.6 FPT_STM.1 Reliable Time Stamps

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.

Depending on the module obtaining the time value one of the following functions is used to obtain the current time from the Operating System:

- The POSIX `time()` function
- The Java `java.util.Calendar.getTime().getTime()` method
- The Java `java.lang.System.currentTimeMillis()` method

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_OCSP_EXT.1.1, FIA_X509_EXT.1, FIA_X509_EXT.2, FTA_SSL.3, and FTA_SSL_EXT.1.

9.7.7 FPT_TUD_EXT.1 Trusted Update

The TOE includes a single tool that users in the Administrator role use to

- check for updates on demand and
- verify and install update packages.

The tool is called the update tool and it is a command line program included with the TOE that provides a command line interface to check, verify, and install updates to the TOE. The update tool is available to an Administrator after they successfully authenticate to the environmental Operating System's local console.

A user in the Administrator role, can execute the update tool to determine whether or not an update package is available. If the tool indicates that a package is available, the user must obtain the update package from Information Security Corporation in the same fashion that they obtained the TOE originally. The TOE update package is delivered in a zipped archive via a download link. A valid serial number is required to download the package. Licensed customers receive a signed email with a download link (URL) and serial number.

An 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 format of this file is as described in *RFC 5652 Cryptographic Message Syntax (CMS)*. The update data is hashed (FCS_COP.1(3)), the hash is digitally signed (FCS_COP.1(2)), and the data and signature are then combined with the certificate chain needed to verify the signature (FIA_X509_EXT.1) to create the output archive file.

When updating the TOE, the tool verifies the validity of the update's digital signature by hashing the contents of the package (FCS_COP.1(3)), verifying the digital signature matches the computed hash value (FCS_COP.1(2)), and validating the certificate per FIA_X509_EXT.1. If the package is valid, the update tool stops the TOE, installs the update, and restarts the TOE. If the signature on the update package is invalid the update tools stops the process and refuses to install the update. 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

9.7.8 FPT_TST_EXT.2 Integrity Test

See Section 9.4.6 FDP_STG_EXT.1 Public Key Protection paragraph three for details. 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. If the integrity check fails, the TOE behaves as described in Section 9.7.4.

9.8 TOE Access (FTA)

The TOE's primary management interface is a client-authenticated web interface that uses HTTPS. Before establishing a privileged user session to the Admin Site and CA Account Site via HTTPS, the TSF displays an Administrator-configured advisory notice and consent warning message regarding use of the TOE.

The TOE supports an administratively defined HTTPS/TLS session timeout (default is 30 minutes). This timeout applies to the Admin Site, the CA Account Site, and the self-service portion of the Public Site. When the timeout occurs the session is terminated and re-authentication is required for access privileges to be granted again. The TOE also supplies a logout link (Admin Site, CA Account Site) or end session button (Self-Service Site) on interactive pages enabling the privileged user or authenticated subscriber to terminate their sessions.

Access to the local console interface of the TOE is managed by the environmental Operating System. The operating system provides mechanisms allowing the user to terminate an interactive session as well as locking a session after a period of inactivity.

To terminate an interactive session in the environmental Operating System, the user selects Sign Out or Log Out from within the Operating System's user interface. When the Operating System locks a session it blanks, or otherwise renders the display unreadable, and waits for input. When the Operating System receives input, it requires re-authentication (username/password) to unlock the session.

Associated SFRs: FTA_SSL.4 User-Initiated Termination, FTA_TAB.1 Default TOE Access Banners, FTA_SSL.3 TSF-Initiated Termination.

9.9 Trusted Path/Channels (FTP)

The TOE requires HTTPS/TLS for any trusted communication between itself and authorized external network based IT entities, remote subscribers, and privileged users. Each HTTPS/TLS session is logically distinct (uses a different cryptographic key to encrypt the communications) from other communication paths and provides assured identification of its end points. HTTPS/TLS protects communicated data from disclosure using encryption and detects modification of communication using a keyed-hash message authentication code. The initial HTTPS/TLS server credential is an RSA-3072 asymmetric key and is generated during installation. The TOE supports TLS v1.1 with TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA only.

The TOE allows the following entities to initiate communication with the TOE:

- Registration Authorities using the RAMI REST-based API
- Audit servers and other clients using the DBAccess REST-based API
- Remote subscribers using EST
- Remote subscribers using a web browser
- Privileged users using a web browser
- Relying parties

A trusted path/channel using HTTPS/TLS with mutual authentication is required for:

- Privileged users
- RAMI clients
- DBAccess clients
- Remote subscribers using certificate-based authentication to EST
- Remote subscribers using a web browser for self-service revocation

A trusted path/channel using HTTPS/TLS without client authentication is required for:

- Remote subscribers using a web browser to upload a certificate request (no authentication)
- Remote subscribers using a web browser to generate and submit a certificate request (no authentication)
- Remote subscribers using EST without certificate-based authentication (EST basic authentication)

The TOE requires trusted communication for privileged user authentication and all remote administration actions including clients using the RAMI or the DBAccess interface. The TOE requires trusted communication for subscriber actions via either the Public site or EST. The TOE does not initiate communication with any external IT entities.

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.

When mutual authentication is used, the TOE performs two path validation operations to verify the authenticity and trust of the client's certificate. The first path validation is performed by the Apache

Tomcat TOE component using Java's path building and validation processing backed by the ISC CDK cryptographic module for performing cryptographic signature validation. This path validation terminates to trust anchors residing in the Apache Tomcat trust store. The second path validation is performed by the CertAgent TOE component on the certificate already verified by Apache Tomcat. This path validation is done using CertAgent's path building and validation processing backed by the ISC CDK cryptographic module for performing cryptographic signature validation. CertAgent maintains its own trust anchor list stored in the database. Additional detail on the Apache Tomcat and CertAgent trust anchor stores can be found in Sections 9.3.8 and 9.4.6.

The cryptographic functionality to support TLS connections is provided by the ISC CDK and the PKCS#11 Cryptographic Module. The TOE uses the ISC CDK for most of the TLS protocol's cryptographic needs and only uses the PKCS#11 Cryptographic Module for operations involving the TLS server private key. The cryptographic functions specified in FCS_CKM.1, FCS_CKM.2, FCS_RBG_EXT.1, and FCS_COP.1(*) are used during HTTPS/TLS session establishment for:

- Ephemeral key generation for key establishment (ISC CDK)
- Key exchange (ISC CDK)
- Cryptographic random bit generation (ISC CDK)
- AES encryption/decryption (ISC CDK)
- Cryptographic hashing (ISC CDK)
- Keyed-Hash message authentication (ISC CDK)
- Cryptographic signature validation (ISC CDK)
- Cryptographic signature (PKCS#11 Cryptographic Module)

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.

Associated SFRs: FTP_TRP.1 Trusted Path, FTP_ITC.1 Inter-TSF trusted channel