Security Target
ADSS PKI Server

Version	V11
Date	2024-07-22
Classification	PUBLIC

.

Version control

Version	Date	Author	Description
v1	2022-03-16	Ascertia LTD	Initial Version.
V2	2022-07-08	Ascertia LTD	Update during the evaluation process.
V3	2022-08-31	Ascertia LTD	Update during the evaluation process.
V4	2022-10-20	Ascertia LTD	Update during the evaluation process.
V5	2022-11-08	Ascertia LTD	Updated section 2
V6	2022-11-30	Ascertia LTD	Update during the evaluation process.
V7	2022-12-13	Ascertia LTD	Update during the evaluation process.
V8	2022-12-21	Ascertia LTD	Update during the evaluation process.
V9	2023-01-09	Ascertia LTD	Update during the evaluation process.
V10	2024-03-05	Ascertia LTD	ALC_FLR.3 augmentation added.
V11	2024-07-22	Ascertia LTD	Public release

Table of Contents

1 Introduction

1.1. Description of ADSS Server

Ascertia's ADSS PKI Server provides a modular trust services framework that delivers all of the components required to issue, validate and manage X.509 and ISO 7816 Card Verifiable Digital Certificates. Ascertia's ADSS Server also provides a high performance OCSP solution for the validation of digital certificates and can provide real-time revocation and certificate whitelisting and can be leveraged as a certificate validation hub for multiple CA's.

ADSS Server offers the highest levels of flexibility for enterprises, governments, and trust services providers, and has been designed to deliver all the services needed to provide highly performant, scalable PKI services.

ADSS Server exposes an intuitive web interface for system operators and a collection of API's and standards based interfaces for the issuance, lifecycle management and validation of certificates and digital signatures.

This document details the ADSS PKI Server Target of Evaluation (TOE) that is in scope for this Common Criteria evaluation for the corresponding Security Target (ST)

1.2. Security Target and TOE references

ST Title	ADSS PKI Server Security Target
ST Version	V11
ST Creation Date	2024-07-22
TOE Reference	ADSS PKI Server v8
TOE Name	ADSS PKI Server
TOE Version	8.0.0

1.3. TOE Overview

Within a PKI, the CA is responsible for issuing and managing public-key certificates for subjects to prove their identities; these subjects are typically called subscribers and can be people, devices, applications, or servers. A public-key certificate is a credential that contains the public key for that subscriber bound with other identifying information using a CA's digital signature. To obtain a certificate, subscribers register with the PKI. Depending on how the PKI is designed, this is done through a Registration Authority (RA) service that verifies the requester's identity before the request is handled by the CA. Part of the registration process is the generation of a private/public key pair that occurs on the subscriber's system. The public key is transmitted to the CA during the registration process. The CA signs the certificate with a digital signature (using its own private key) that binds the public key and other identifying information to the subscriber. In this capacity, the CA acts as a trusted third party by asserting the authenticity of the subscriber, the public key, and the binding of the subscriber to the public key. This allows relying parties (e.g., individuals or applications) to verify and trust signatures or assertions made by the subscriber using the private key that corresponds to the public key contained in the certificate. This also allows the relying parties to use the public key in the certificate to carry out encrypted communication with the subscriber.

ADSS PKI Server is a next-generation PKI solution that manages the complete life cycle of public key certificates based on X.509 v3 and CVC BSI TR-03110 and Certificate Revocation List based on X.509 v2 standards. It is suitable for enterprise and Internet-based Public PKIs as well as national

CAs and Qualified Trust Service Providers (QTSPs) enabling the issuance, validation and lifecycle management of digital certificates for a variety of use cases, including:

- Digitally signing documents
- Code Signing
- Email Security
- Strong authentication of web servers and client via TLS
- Secure logical access to desktops and servers via smartcard logon
- Secure access to VPN Networks
- Secure physical access to buildings
- Issuing identity cards to employees, ID cards or electronic passports to citizens
- Creating and managing Country Verifying, Document Verifying CA's and inspection system certificates for the inspection of EAC enabled machine readable travel documents.

TOE is a web application software implementing a PKI CA infrastructure running on an application server. TOE mainly provides certificate issuance and revocation status services.
TOE consists of three components named:

1. ADSS PKI Server Service

   ADSS PKI Server Service provides different web services to perform various operations e.g. certificate generation/revocation requests etc. it is further divided into two sub-components:

   - ADSS CA Service: Provides APIs to generate/renew/revoke certificates
   - ADSS OCSP Service: Provides APIs to get the revocation status of a certificate

   Both ADSS CA Service and ADSS OCSP Service can be deployed on same machine running on same application server and they can also be deployed on separate machines with their own application server.

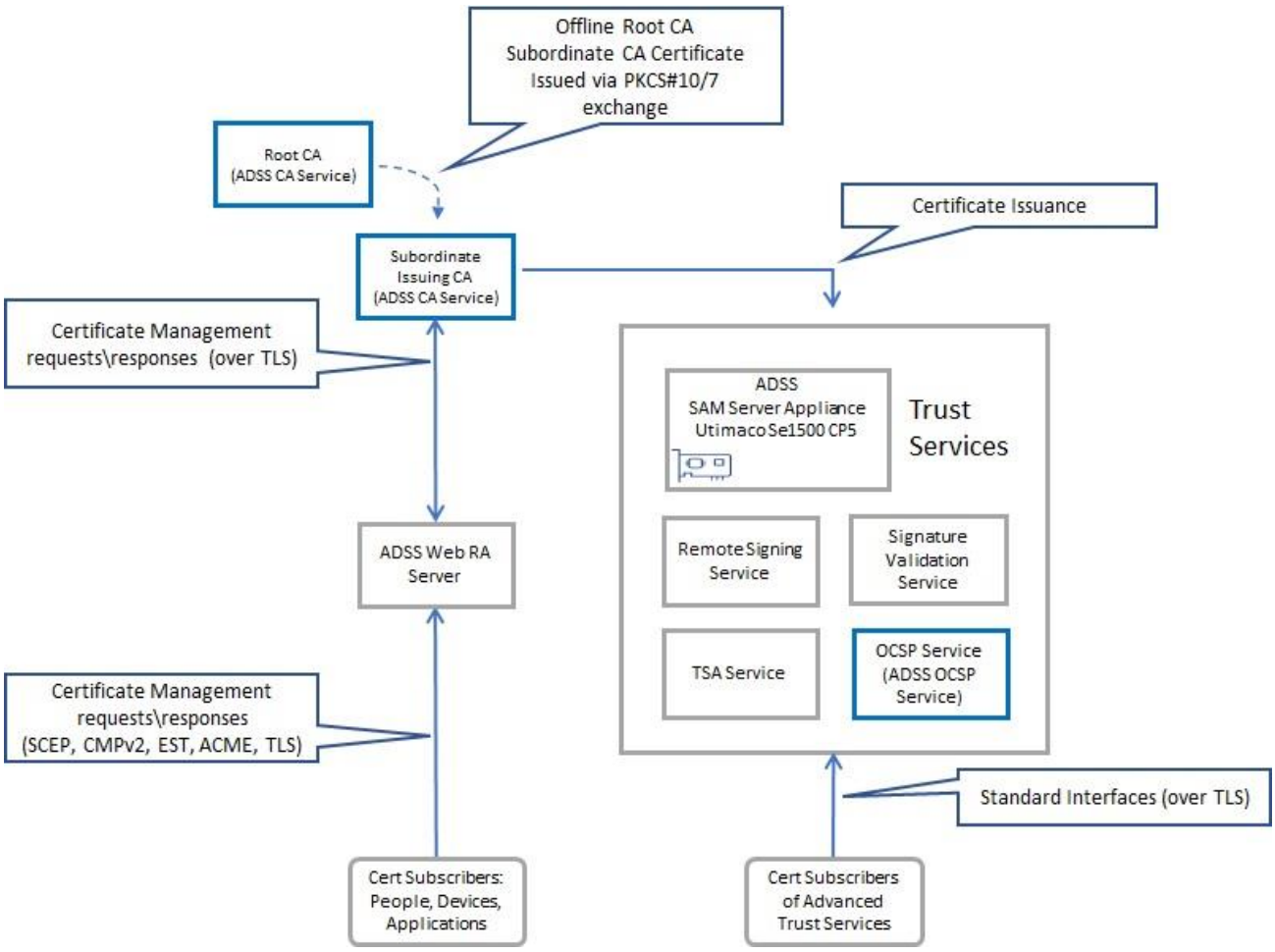
2. ADSS PKI Server Admin Console

   ADSS PKI Server Admin Console allows administrators to configure the product, e.g. define access control, certification profile management, certificate template management, configuring crypto source i.e. HSM etc.

3. ADSS PKI Server Core

   ADSS PKI Server Core performs various background tasks e.g. Logs archiving, DB monitoring, HSM monitoring etc.

In a typical PKI infrastructure, one TOE can be deployed as a Root CA and another TOE as a subordinate CA. The ADSS OCSP Service can also be deployed separately. This is shown in the image below:



ADSS CA Service

ADSS CA Service meets the requirements of X.509, PKIX, ETSI, CA/B Forum, ICAO and CWA 14167-1 requirements for trustworthy systems making it suitable for use by Qualified Trust Service Providers (TSPs), National Root CA, Intermediate CAs, Issuing CAs for global public trust as well as for Enterprise CAs in a closed environment.

ADSS CA Server can be used to set-up off-line X509v3 based Root CA's, multiple Subordinate CAs, ISO 7816 based Country Verifying and multiple Document Verifying CA's from the same instance.

ADSS CA Server uses certificate templates to manage the certificate contents for common certificate types e.g. document signing, TLS/SSL client certificates, email security, code signing, archive signing etc.

It supports FIPS 140-2 level 3 and Common Criteria certified HSMs to store and protect all cryptographic keys.

Support for the common cryptographic algorithms is provided including SHA1, SHA-2, SHA-3 (SHA-256, SHA-384, SHA-512, SHA3-256, SHA3-384, SHA3-512), RSA keys up to 8192 bits and ECDSA up to 521 bits (NIST and Brainpool).

Offers certificate lifecycle services using a flexible web services interface

- It supports local key generation and certification of the public key via ADSS Web RA Server or 3rd party business application to generate keys and certificates in cryptographic smartcards.

- It also supports remote key generation and certification within network-based Hardware Security Modules or Remote Signature Creation Devices.

ADSS Server implements strong operator authentication, detailed and HMAC-protected transaction logging, fine-grain access control, dual control and many other security features for a high-trust PKI system.

ADSS OCSP Service

It is an advanced x.509 certificate Validation Authority service that fully conforms to the IETF RFC 6960, RFC 5019 standard and CA/B Forum white-list checking requirements.

It is also FIPS 201-2 Certified (APL #1411) and approved for use by US federal agencies for HSPD-12 implementations.

Operate as a robust validation hub solution capable of providing OCSP certificate validation services for multiple Certificate Authorities (CAs) concurrently.

Simple or sophisticated validation policies are supported for each individual CA and ADSS OCSP Server provides a detailed historical record of all transactions.

Retrieve certificate status information from the CAs using multiple methods, e.g. HTTP/S CRLs, LDAP/S CRLs, peer OCSP responders and real-time revocation information using the CA's database.

The OCSP Server supports whitelist checking, i.e. a check to see if the certificate was actually issued by the CA (supports the Extended Revoked Definition extension of RFC 6960).

- Supports FIPS 140-2 level 3 and Common Criteria certified HSMs to store and protect all cryptographic keys.

TOE Usage and Major Security Features

The Ascertia ADSS PKI Server manages the complete life cycle of public key certificates based on X.509 standard. The TOE functionality includes but is not limited to:-

- Certificate Generation
- Return Certificate Status Information in form of X509 CRL & OCSP Response

TOE Type

The TOE is the ADSS PKI Server, that consists of software components: ADSS CA Server and ADSS OCSP Server. The TOE is a web application software implementing a PKI CA infrastructure running on an application server.

TOE Hardware/Software/Firmware

1. **Root CA:** The Root CA service is the final point of trust in a PKI (aka trust anchor). It's a self-signed CA and implemented by installing ADSS CA Server on a tempest laptop with an offline HSM. The main role of the Root CA is to certify subordinate CAs (intermediate or issuing CAs) and to manage their revocation.

2. **Subordinate CAs:** These are typically online CAs which issue certificates to subscribers and trust service provider components e.g. OCSP Servers. The ADSS CA Server is used to implement Subordinate CAs together with online HSMs (network attached or PCIe-based).

3. **OCSP Service:** Component that provides the certificate status information of the requested certificate.

Non-TOE Hardware/Software/Firmware

The following hardware, firmware and software are supplied by the IT environment, and are therefore excluded from the TOE boundary: -

Advance Trust Services: These are additional that complement the timestamping service, as well as remote signing and signature validation services. These trust services are issued certificates by the subordinate CA. The relevant ADSS Server modules which implement these trust services are ADSS TSA Server, ADSS Signing Server, including ADSS Signature Activation Module (SAM) Server, and ADSS Verification Server.

Web RA Service: This service component provides the complete certificate lifecycle management to human users through a web interface over HTTPS. The Web RA Service then communicates with the relevant Subordinate CA to request the certificate (or its revocation/suspension. The relevant ADSS Server modules which implement this service is the ADSS Web RA Server.

SigningHub: Ascertia's SigningHub is a web-based application that provides document signing and certificate lifecycle management capabilities. It interacts with ADSS PKI Server for complete certificate lifecycle management through web APIs over HTTPS. It communicates with the relevant Subordinate CA to request certificates or their revocation/suspension.

Operating System: Windows Server, Linux

Database: e.g. Microsoft SQL Server, Azure SQL Database, Oracle, PostgreSQL, MySQL etc.

HSMs: e.g. Thales Luna, Entrust nShield, Utimaco etc.

NTP Servers: Any local or remote NTP Server

Administrator Clients: Any modern internet browser with operator TLS client authentication (smartcard/token based)

1.4. TOE Description

ADSS PKI Server is a next-generation PKI solution that manages the complete life cycle of public key certificates based on the X.509 standard and IETF Internet Standard protocols. It is suitable for enterprise and Internet-based Public PKIs as well as national CAs and Qualified Trust Service Providers (QTSPs).

The TOE can be used to deliver the following services:

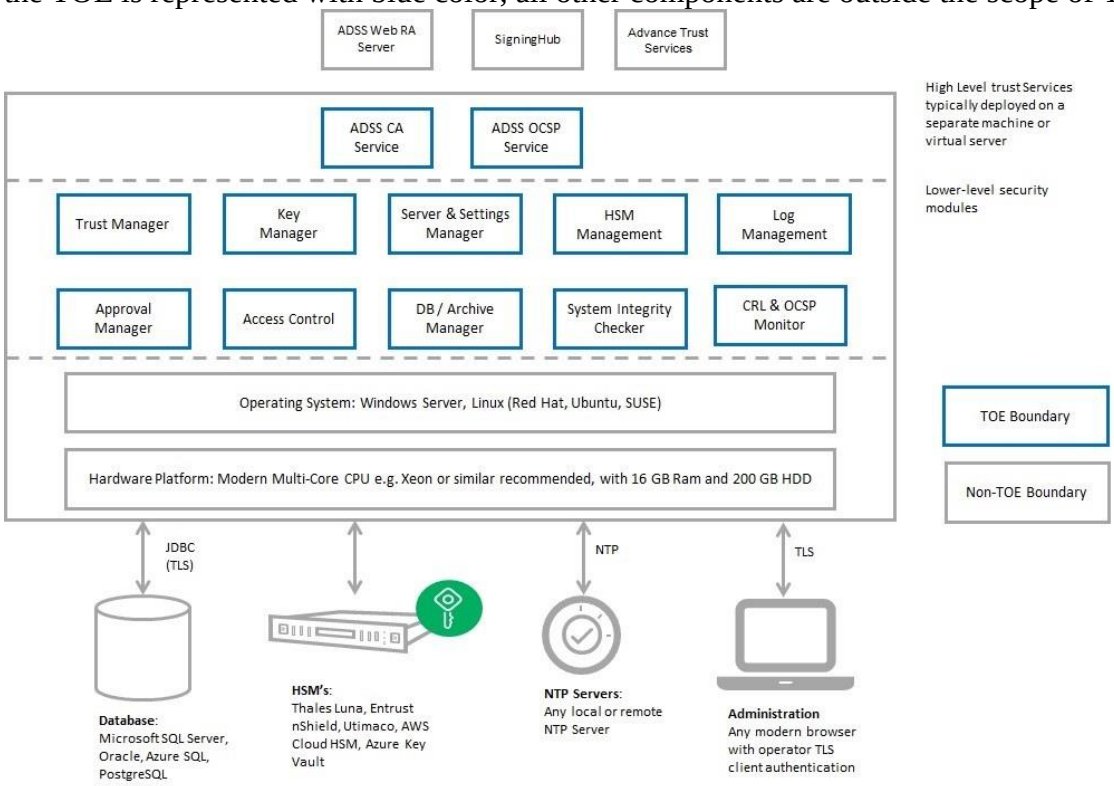
- Strong entity authentication, e.g.:
    - Authentication of users/servers over the Internet through the issuance of TLS (previous versions known as SSL) client and server certificates
    - Authentication of user for secure logon to Windows and/or Linux by provisioning certificates on smartcards
    - Authentication of users and devices as part of establishing a VPN connection

    - Authentication of users as part of Single Sign On (SSO) to web applications
    - Authentication of users of mobile devices (iOS and Android)
- Digital signatures and encryption e.g.:
    - Signing and encrypting documents (e.g. PDFs, Office documents etc.)
    - Signing and encrypting emails
    - Signing of software code for trusted delivery and execution
- Establishing Trusted Service Provider (TSP) components, e.g.:
    - Setting up Certificate Authorities (CAs), including Root CAs, Intermediate CAs and Subordinate CAs
    - Issuing certificates to Time Stamp Authorities
    - Issuing certificates to Validation Authorities (OCSP, SCVP, XKMS certificate validation Servers and document signature verification providers)
    - Issuing user certificates as part of Remote Signing Service
    - Issuing certificates to Long-Term Archives & Notary Service providers

TOE Boundary

TOE exists as Java EE 11 supported on a range of 64-bit operating systems as illustrated below. The high-level trust services are typically deployed on separate machines in an n-tier architecture. The lower level security services are available to the higher-level trust services where applicable.

The TOE interacts with other components to implement its security functions. In the below figure, the TOE is represented with blue color, all other components are outside the scope of TOE.



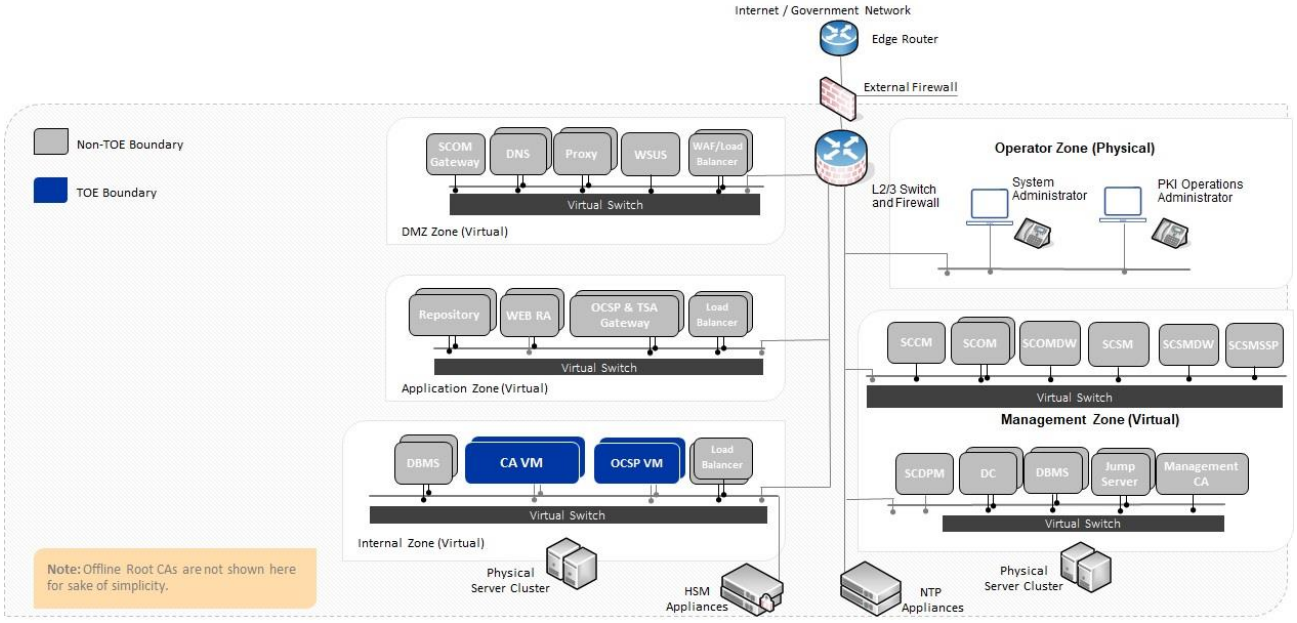
Implements strong operator authentication, detailed and HMAC-protected transaction logging, fine-grain access control, dual control, and many other security features for a high-trust PKI system.

Physical scope of the TOE

The TOE exists as a Java EE 11 application supported on a range of 64-bit operating systems as illustrated below. The high-level trust services are typically deployed on separate machines in an n-tier architecture. The lower-level security services are available to the higher-level trust services where applicable. The TOE interacts with other components to implement its security functions. In the figure above, the TOE is represented with blue color, all other components are outside the scope of TOE. Although the TOE could be deployed on separate machines, the TOE itself is the Java EE 11 application and its security features are not affected by the underlying hardware.

TOE Deployment

The TOE is expected to be deployed using industry best practices for network security by separating into separate security zones as illustrated below:



Note in the above diagram it is assumed the Management Platform is built around Microsoft System Centre, although this is considered outside the scope of the TOE and any other equivalent product can be used.

Delivery of the TOE

Ascertia will provide the Ascertia ADSS PKI Server software installer in form of a ZIP. For Windows Operating System, the ZIP file name will look contain the following format:

- `ADSS-Server-v8.0-Win64-05Jan2023.zip`

Similarly, for Linux, the ZIP name will be like this:

- `ADSS-Server-v8.0-Lin64-05Jan2023.zip`

1. The software is uploaded to Ascertia's community site from where clients can download it and this site requires user authentication. Ascertia will also upload the SHA256 checksum of this installer on the community site.
2. Ascertia will also provide details of how to verify the cryptographic checksum to its ADSS PKI Server

Guidance documents

1. Title: AGD_OPE: Operational User Guidance – ADSS PKI Server
   File name: AGD_OPE_Operational_user_guidance_v5.pdf
2. Title: AGD_PRE: Preparative Procedure - ADSS PKI Server
   File name: AGD_PRE_Preparative_procedures_v4.pdf

The guidance documents are part of the installer zip under `<TOE-Release/docs>` directory.

TOE users are provided with the flaw remediation guide: *Ascertia Support Services Guide* [SSG] document, which is part of the Flaw Remediation Guidance. This document describes the procedures to report suspected or found security flaws/vulnerabilities and it is available on the Community portal. The link is provided to each customer when they buy the TOE.

Logical scope of the TOE

Security Audit

Each part of the TOE provides detailed and HMAC-protected transaction logging. The TOE audits all security related events. All the audit records produced as a result of the TOE operator actions or the API requests from any clients are stored in Ascertia ADSS PKI Server database. The audit records do not include any data which allows the retrieval/decipherment of confidential data. Each Audit record are associated with the relevant user (who performed the action), and with the relevant object (what did the user do).
TOE provides ability to view, search the audit events for specified roles (Auditors). Audit logs are stored in HMAC integrity protected database and can be backed up in digitally signed format on the file system.
In case of the audit trail cannot be written for example audit service is down or run out of space there will be a rollback that means no operation can be committed so there will be no change to any data that are not audited.

Communication, Trusted Path

The TOE implements and enforces the following trusted communication methods and protocols:

- **Operators:** Operators (Administrator, Auditor, Security Officer, CA Operations Staff) access the ADSS PKI Server Admin Console GUI over a mutually authenticated TLS v1.2 channel.
- **TOE Services:** TOE communicates with all its services (eg OCSP Service) mutually authenticated TLS v1.2 channel.
- **External Services:** TOE communicates with all external client services (eg RA and Web RA, Remote Signing Service, Signature Validation Service, TSA Service)

- **CM:** The TOE communicates with CM using vendor specific APIs. User passwords do not travel on this channel. This communication with the CM is enforced by the CM to be secure, authenticated and protected from replay attacks, i.e. the CM meets the requirements of EN 419 221-5 Protection Profile and is certified to Common Criterial EAL4+ level.

The TOE provides certificate-based proof of origin and proof of receipt for issued certificates through CRLs and OCSP responses. The TOE also verifies certificate related messages using signed CMC requests and responses.

Cryptographic Support

The TOE is highly dependent on a certified HSM. Most crypto operations are performed by the HSM.

Due to performance issues the TOE itself generates TLS Client and TLS Server keys and store it in database encrypted by DEK. HTTPS using TLS is used to connect to database external services and also to connect to internal TOE components.

DEK is generated by the TOE but encrypted with KEK that is generated by and stored in HSM.

There are some keys (LogSigning, CA key etc..) that can be configured the generate by TOE or the HSM. Whenever a key is generated by the TSF, the random is also generated by the TSF. If the key is generated in the HSM, the random is also generated by the HSM.

RSA and ECC schemes are supported. Key establishment is also supported.

All keys are destructed when are not more needed. For example, when a client or user is removed from the system all their keys and passwords are deleted from database, device and memory and not used anymore for any purpose.

Public keys are protected and the keyed hash of them are verified each time they are used.

Crypto functions used:

- AES encryption
  - KEK: It uses AES to encrypt/decrypt DEK
  - DEK: It uses ASES encryption to encrypt/decrypt passwords/keys in ADSS database.
  - TLS/HTTPS session keys: During TLS communication an AES key is generated to encrypt/decrypt the traffic.
- Signing
  - Certificates signed by CA keys
  - TLS Session keys
  - Log signing key
  - If the key is generated by the TSF, signature is performed by the TSF. If the keys are generated in the HSM, signature is also performed by HSM
- Hashing
  - Hashing is performed before each signing operation (certificate sign, CRL sign, TLS, log sign etc…)
  - If the key is generated by TSF, hashing is also done by the TSF. If the key is generated by the HSM, hashing is also performed by HSM
- Keyed-Hash Message Authentication

- If HMAC key is generated in the TOE, TOE performs the HMAC operation, if HMAC key is generated in the HSM, HSM performs the HMAC.

User Data Protection

All the user data is stored in an external database. The sensitive data is always encrypted and the database records are HMAC integrity protected.
The user certificates are always created based on certificate templates that are pre-configured by Administrators. The certificate requests are always linked to the issued certificates. The certificate requests approvals can be done by CA Operations Staff. The TOE supports both CRL and OCPS standards for providing certificate status information. In case of any user deallocation, all its passwords and keys are made unavailable and are not re-used anymore.
All communication between any of the TOE components and its environment are TLS protected.
All the keys that are used for authentication or channel encryption are protected by integrity mechanisms just like all records in the database.

Identification and authentication

Upon accessing the TOE, the user has to logon to the ADSS PKI Server using TLS client certificates before being allowed to perform any activity. TLS client certificates and associated private keys should be stored on a secure smart card/USB token thereby providing an extra layer of security for the private key plus two-factor authentication. The revocation status of the TLS certificates can also be checked at the time of logon by configuring this in ADSS PKI Server. However, it is recommended that the accounts are also immediately updated on ADSS PKI Server at the time a certificate is revoked. The Ascertia ADSS PKI Server ensures that access to system objects is strictly controlled. Users are first identified and authenticated as explained above, and once this process is complete and the user has successfully logged in, then access to system objects is controlled according to the user's role. Each role has a definition of which system objects it can access, and the type of access, e.g., read only, or edit/create/delete.

Security Management

The following entities have access to the TOE:

- **Operator:** The operator has access to ADSS PKI Server Admin Console where operator's permissions are controlled using Role Based Access Control (RBAC) mechanism. In RBAC different roles with different permissions are created on ADSS PKI Server Admin Console and then these roles are assigned to operators. One operator is mapped to only one role at a time. These operator roles are Administrator, Auditor, Security Officer and CA Operations Staff. The first three roles come pre-installed whereas the CA Operations Staff role is created after installation to perform CA specific operations only.
- **Business Application:** It has access to web APIs of the ADSS PKI Server Service to perform different operations. A business application is first registered in ADSS PKI Server using ADSS PKI Server Admin Console where TLS client certificate of the business application is also configured. During interaction of the business application with ADSS PKI Server Service it is authenticated using its registered TLS client certificate.

Protection of the TSF

The TSF is protected via multiple security mechanisms. All sensitive data are stored encrypted via DEK (data encryption key) in the database. All the data that are stored in database are HMAC integrity protected. KEK (Key Encryption Key – that the DEK is encrypted with) is generated and stored in a certified HSM module and can never leave the HSM in plain text. Neither any other keys from the database.

The TOE monitors its services and any time if any of its or its environments services become unavailable it stops working. The TOE keeps polling the relevant services and whenever they are up and running it can resume working without any user interaction. The TOE uses external database, HSM and NTP server and whenever any of these are down, it will stop working.

TOE also supports a trusted update in case of any bugfix is needed.

TOE Access

The TOE is accessible via Admin Console and its service interfaces.

The service interfaces are protected, and each client must authenticate with their own TLS certificate before invoking any functions.

The Admin Console is available after TLS certificate authentication. The Operators are able to logout and terminate their sessions.

2. Conformance Claims

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

This ST and Target of Evaluation (TOE) are Part 2 extended.

This ST and Target of Evaluation (TOE) are Part 3 conformant.

This ST does not claim exact conformance, but based on Protection Profile for Certification Authorities, version 2.1 [PP_CA_V2.1].

This ST claims conformance to: EAL4 assurance package defined in the [CC3] augmented by ALC_FLR.3.

Note that this evaluation also includes evaluation assurance activities that are defined in the [PP_CA_V2.1] that has augmented the CEM and are not considered to be alterations to Part 3.

2.1. Applicable Technical Decisions

The table below contains the technical decisions related to the [PP].

0599 – Corrections to SAR Section in CAPP	Not Applied as the TOE was evaluated on EAL4.
0522 – Updates to Certificate Revocation (FIA_X509_EXT.1)	Applied
0500 – Cryptographic selections and updates for CAPP	Applied
0415 – Trusted Update Test 4 Conditional	Only affects testing activity
0375 – FMT_MOF.1(4) selection	Applied
0353 – Guidance for Certificate Profiles	Only affects testing activity

0348 – FCS_TLSS_EXT.2.4 for TLS 1.2 or higher	Only affects testing activity
0328 – Split Knowledge Procedures distinction	Applied
0294 – Correction of TLS SFRs in CA PP ver 2.1	Applied
0287 – FAU_STG.4 Testing	Applied
0286 – Audit Events for FPT_RCV.1	Applied
0278 – Clarification of Role for Managing Manual Certificate Requests	Applied
0276 – X.509 Code Signing on TOE Updates	Applied

3. Security Problem Description

The security problem is described in terms of the threats that the TOE is expected to address, assumptions about its operational environment, and any organizational security policies that the TOE is expected to enforce.

3.1. Threats

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 nonrepudiation.

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.

3.2. Assumptions

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 assumed to follow and apply all administrator guidance in a trusted manner.

3.3. Organizational Security Policies

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.

4. Security Objectives

4.1. Security Objectives for 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.
Addressed by: 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.
Addressed by: FAU_ADP_EXT.1

O.CERTIFICATES

The TSF must ensure that certificates, certificate revocation lists, and certificate status information are valid.
Addressed by: 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.
Addressed by: 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.DISPLAY_BANNER The TOE will display an advisory warning regarding use of the TOE.
Addressed by: FTA_TAB.1

O.INTEGRITY_PROTECTION

The TOE will provide appropriate integrity protection for TSF data and software and any user data stored by the TOE.
Addressed by: FCS_CDP_EXT.1, FCS_CKM_EXT.5, FDP_ITT.1, FPT_ITT.1, 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.
Addressed by: FCO_NRO_EXT.2, FCO_NRR_EXT.2

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.

Addressed by: 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.7, FCS_CKM_EXT.8, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_HTTPS_EXT.1, FCS_RBG_EXT.1, FCS_STG_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, FDP_ITT.1, FPT_ITT.1, FPT_KST_EXT.1, FPT_KST_EXT.2, FPT_SKP_EXT.1, FTP_ITC.1, FTP_TRP.1

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).
Addressed by: FCS_CDP_EXT.1, FPT_FLS.1, FPT_RCV.1

O.RESIDUAL_INFORMATION_CLEARING

The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
Addressed by: FDP_RIP.1

O.SYSTEM_MONITORING

The TOE will provide the capability to generate audit data. The TOE will record in audit records: date and time of action and the entity responsible for the action.
Addressed by: FAU_ADP_EXT.1, FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.3, FAU_SCR_EXT.1
(selection-based), FAU_GCR_EXT.1, FAU_STG_EXT.1, FIA_UIA_EXT.1, FPT_STM.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.
Addressed by: 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.4

O.TSF_SELF_TEST

The TOE will provide integrity protection to detect modifications to firmware, software, and archived data.
Addressed by: FPT_TST_EXT.2
Application Note: If this SFR is not claimed by the TOE, this functionality is expected to be satisfied by the environmental objective OE.TRUSTED_PLATFORM.

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.
Addressed by: FCS_CDP_EXT.1, FCS_COP.1(2), FIA_X509_EXT.2, FPT_TUD_EXT.1

4.2. Security Objectives for the Operational Environment

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_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.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.SESSION_PROTECTION_LOCAL The Operational Environment provides the ability to lock or terminate local administrative sessions.

OE.SESSION_PROTECTION_REMOTE The Operational Environment provides the ability to lock or terminate remote 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 on which the TOE has been installed is securely configured, regularly patched, and not subject to unauthorized access.

4.3. Security Objectives Rationale

The following table illustrates the correspondence between the threats, assumptions, and organizational security policies described in the security problem definition and the TOE/environmental objectives that are satisfied in order to ensure that the threats are sufficiently mitigated by the TSF and the Operational Environment.

Table 3 - Security Objective Mapping

SPD Element	Objective	Requirements
A.NO_GENERAL_PURPOSE It is assumed that there are no generalpurpose 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 services, ineffective security mechanisms, or unintended circumvention of security mechanisms.	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.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.4
	OE.AUDIT_STORAGE The Operational Environment provides a mechanism for the storage 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.	
	OE.SESSION_PROTECTION_REMOTE The Operational Environment provides the ability to lock or terminate remote administrative sessions.	N/A
	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	N/A

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
	O.RECOVERY The TOE will have the capability to store and recover to a previous state at the direction of the administrator	FCS_CDP_EXT.1, FPT_FLS.1, FPT_RCV.1
	OE.TRUSTED_PLATFORM The operating system on which the TOE has been installed is securely configured, regularly patched, and not subject to unauthorized access.	FPT_TST_EXT.2
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,	FDP_CER_EXT.1, FDP_CRL_EXT.1, FDP_OCSPG_EXT.1,

	and firmware), and components (software, hardware, and firmware), auditing of configuration data, and controlling changes to configuration items.	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, FDP_ITT.1, FPT_ITT.1, FPT_TST_EXT.2, FCS_CDP_EXT.1(1), FCS_CDP_EXT.1(2), FCS_CKM_EXT.5
	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, FCO_NRR_EXT.2

	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	[Remove if all administrative actions from O.CONFIGURATION_MANAGEMENT requirements are performed directly by the TOE]

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.7, FCS_CKM_EXT.8, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_HTTPS_EXT.1, FCS_RBG_EXT.1(1), FCS_RBG_EXT.1(2), FCS_STG_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, FDP_ITT.1, FPT_ITT.1, 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.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.SESSION_PROTECTION_LOCAL The Operational Environment provides the ability to lock or terminate local administrative sessions.	N/A

	OE.SESSION_PROTECTION_REMOTE The Operational Environment provides the ability to lock or terminate remote administrative sessions.	N/A

	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	N/A
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(2), 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_PROTECTION The TOE will protect audit records against unauthorized access, modification, or deletion to ensure accountability of user actions.	N/A
	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_SCR_EXT.1, FAU_GCR_EXT.1, FAU_STG_EXT.1, FIA_UIA_EXT.1, FPT_STM.1
	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	O.RESIDUAL_INFORMATION_CLEARING	FDP_RIP.1

cleared when resources are reallocated.	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.	
T.WEAK_CRYPTO A weak hash or signature scheme may be compromised by an attacker and used to	O.PROTECTED_COMMUNICATIONS The TOE will provide protected communication channels for administrators,	FCS_CDP_EXT.1, FCS_CKM.1, FCS_CKM.2
apply integrity checks to malicious content so that it appears legitimate.	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_CKM_EXT.1(1), FCS_CKM_EXT.1(2), FCS_CKM_EXT.4, FCS_CKM_EXT.7, FCS_CKM_EXT.8, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_HTTPS_EXT.1, FCS_RBG_EXT.1, FCS_STG_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, FDP_ITT.1, FPT_ITT.1, 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(2), 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

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.	O.DISPLAY_BANNER The TOE will display an advisory warning regarding use of the TOE.	FTA_TAB.1

5. Security Requirements

The Security Functional Requirements (SFRs) included in this section are derived from Part 2 of the *Common Criteria for Information Technology Security Evaluation*, *Version 3.1*, *Revision 5*, with additional extended functional components.

The CC defines operations on Security Functional Requirements: assignments, selections, refinements and iterations. This document uses the following font conventions to identify the operations defined by the CC:

- **Refinement** Operation (denoted by **bold text**) is used to add details to a requirement, and thus further restricts a requirement.
- **Selection** (denoted by *italicized text):* is used to select one or more options provided by the [CC] in stating a requirement.
- **Assignment** operation (denoted by *italicized text*) is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: are identified with a number inside parentheses (e.g. "(1)").

Operations that are already done by [PP_CA_V2.1] are not highlighted but marked in the same way as the operations performed by the ST Author in this document.

5.1. TOE Security Functional Requirements Summary

The following table lists the SFRs claimed by the TOE:

Class	Component
Security Audit (FAU)	FAU_ADP_EXT.1 FAU_SAR.1 FAU_SAR.3 FAU_STG_EXT.1 FAU_GCR_EXT.1 FAU_GEN.1 FAU_GEN.2 FAU_STG.4 FAU_SCR_EXT.1
Communication (FCO)	FCO_NRO_EXT.2 FCO_NRR_EXT.2

Cryptographic Support (FCS)	FCS_CDP_EXT.1(1) FCS_CDP_EXT.1(2) FCS_CKM.1 FCS_CKM.2 FCS_CKM_EXT.1(1) FCS_CKM_EXT.1(2) FCS_CKM_EXT.4 FCS_CKM_EXT.5 FCS_CKM_EXT.7 FCS_CKM_EXT.8 FCS_COP.1(1) FCS_COP.1(2) FCS_COP.1(3) FCS_COP.1(4) FCS_RBG_EXT.1(1) FCS_RBG_EXT.1(2) FCS_STG_EXT.1 FCS_HTTPS_EXT.1 FCS_TLSS_EXT.1 FCS_TLSS_EXT.2 FCS_TLSC_EXT.2
User Data Protection (FDP)	FDP_CER_EXT.1 FDP_CER_EXT.2 FDP_CER_EXT.3 FDP_CSI_EXT.1 FDP_RIP.1 FDP_CRL_EXT.1 FDP_OCSPG_EXT.1 FDP_ITT.1 FDP_STG_EXT.1
Identification and Authentication (FIA)	FIA_X509_EXT.1 FIA_UAU_EXT.1 FIA_UIA_EXT.1 FIA_ESTS_EXT.1
Security Management (FMT)	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
Protection of the TSF (FPT)	FPT_FLS.1 FPT_KST_EXT.1 FPT_KST_EXT.2 FPT_RCV.1

	FPT_SKP_EXT.1 FPT_STM.1 FPT_TUD_EXT.1 FPT_TST_EXT.2 FPT_ITT.1
TOE Access (FTA)	FTA_SSL.4 FTA_TAB.1
Trusted Path/Channels (FTP)	FTP_TRP.1 FTP_ITC.1

Extended components definition

All the extended requirements in this ST have been drawn from CA PP [PP_CA_V2.1]. The PP defines the
following extended requirements and since they are not refined in this ST the PP should be consulted for more information in regard to those CC extensions.
The extended components are as follows:

- FAU_ADP_EXT.1: Audit Dependencies
- FAU_STG_EXT.1: External Audit Trail Storage
- FAU_GCR_EXT.1: Generation of Certificate Repository
- FAU_SCR_EXT.1: Certificate Repository Review
- FCO_NRO_EXT.2: Certificate-Based Proof of Origin
- FCO_NRR_EXT.2: Certificate-Based Proof of Receipt
- FCS_CDP_EXT.1: Cryptographic Dependencies
- FCS_CKM_EXT.1: Cryptographic Key Generation
- FCS_CKM_EXT.4: Cryptographic Key Destruction
- FCS_CKM_EXT.5: Public Key Integrity
- FCS_CKM_EXT.7: Key Generation for KEKs
- FCS_CKM_EXT.8: Key Hierarchy Entropy
- FCS_RBG_EXT.1: Cryptographic Random Bit Generation
- FCS_STG_EXT.1: Cryptographic Key Storage
- FCS_HTTPS_EXT.1: Extended: HTTPS Protocol
- FCS_TLSC_EXT.2: TLS Client Protocol with Mutual Authentication
- FCS_TLSS_EXT.1: TLS Server Protocol
- FCS_TLSS_EXT.2: TLS Server Protocol with Mutual Authentication
- 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_CRL_EXT.1: Certificate Revocation List Validation
- FDP_OCSPG_EXT.1: OCSP Basic Response Generation
- FDP_STG_EXT.1: Public Key Protection

- FIA_UAU_EXT.1: Authentication Mechanism
- FIA_UIA_EXT.1: User Identification and Authentication
- FIA_X509_EXT.1: Certificate Validation
- FIA_X509_EXT.2: Certificate-Based Authentication
- FIA_ESTS_EXT.1: Enrollment over Secure Transport (EST) Server
- 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

5.2. TOE Security Functional Requirements

The Security Functional Requirements (SFRs) included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, with additional extended functional components. The following table lists the SFRs that are defined in this section as well as any auditable events associated with their enforcement. The following table presents the baseline (mandatory) requirements for compliant TOEs, and also used to specify whether the TSF or OE is responsible for actions pertaining to a particular audit event associated with the SFRs (this is done in FAU_ADP_EXT.1 below). If the TOE relies on the Operational Environment to provide some of the TOE's auditing functionality, it is identified whether each of the auditable events for the claimed SFRs are implemented by the TSF or by the Operational Environment, along with the specific environmental component that provides the auditing functionality if applicable.

Table 4 - Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents	Retention Normal/Extended	Responsible TSF or OE Component
FAU_ADP_EXT.1	None.	None.	N/A	
FAU_SAR.1	None.	None.	N/A	
FAU_SAR.3	None.	None.	N/A	
FAU_STG_EXT.1	None.	None.	N/A	
FAU_GCR_EXT.1	None.	None.	N/A	
FAU_GEN.1	None.	None.	N/A	
FAU_GEN.2	None.	None.	N/A	
FAU_STG.4	None.	None.	N/A	
FAU_SCR_EXT.1	None.	None.	N/A	
FCO_NRO_EXT.2	None.	None.	N/A	
FCS_CDP_EXT.1	None.	None.	N/A	
FCS_CKM.1	All occurrences of non-ephemeral and	Success: public key generated	Normal	TSF

	[*no other*] key generation for TOE related functions.			
FCS_CKM.2	All occurrences of nonephemeral and [*ephemeral*] key establishment for TOE related functions.	Success: key established	Normal	OE
FCS_CKM_EXT.1(1)	None.	None.	N/A	
FCS_CKM_EXT.1(2)	None.	None.	N/A	
FCS_CKM_EXT.4	Failure of the key destruction process for TOE related keys.	Identity of object or entity being cleared.	Normal	OE
FCS_CKM_EXT.5	Detection of integrity violation for stored TSF data.	None.	Normal	TSF
FCS_CKM_EXT.7	None.	None.	N/A	
FCS_CKM_EXT.8	None.	None.	N/A	
FCS_COP.1(1)	None.	None.	N/A	
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 Normal	TSF TSF
FCS_COP.1(3)	None.	None.	N/A	
FCS_COP.1(4)	None.	None.	N/A	
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	OE
FCS_RBG_EXT.1	None.	None.	N/A	
FCS_TLSC_EXT.2	Failure to establish a TLS session. Establishment/ Termination of a TLS session.	Reason for failure. None.	Normal	OE
FCS_TLSS_EXT.1	Failure to establish a TLS session.	Reason for failure.	Normal	OE

	Establishment/ Termination of a TLS session.	None.		
FCS_TLSS_EXT.2	Failure to establish a TLS session. Establishment/ Termination of a TLS session.	Reason for failure. None.	Normal	OE
FDP_CER_EXT.1	Certificate generation.	Success: [certificate object identifier].	Extended	TSF

FDP_CER_EXT.2	Linking of certificate to certificate request	Success: [certificate object identifier], [Certificate request]. Failure: Reason for failure, [Certificate request].	Extended	TSF
FDP_CER_EXT.3	Failed certificate approvals.	Reason for failure. [Certificate request,].	Normal	TSF
FDP_CSI_EXT.1	None.	None.	N/A	
FDP_RIP.1	None.	None.	N/A	
FDP_CRL_EXT.1	Failure to generate CRL.	None.	Normal	TSF
FDP_ITT.1	None.	None.	N/A	
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	
FDP_OCSPG_EXT.1	Failure to generate certificate status information.	None.	Extended	TSF
FIA_ESTS_EXT.1	EST requests (generated or received) containing certificate requests or revocation	Identifiers for all entities authenticating the request, including the entity providing client authentication	Extended	

	requests. EST responses issued.	for the EST transport (if any). The submitted request. Any signed response.		
FIA_X509_EXT.1	Failed certificate validations.	None.	Normal	TSF
FIA_X509_EXT.2	Failed authentication	None	Normal	OE
FIA_UAU_EXT.1	All uses of the authentication mechanism used for access to TOE related functions.	Origin of the attempt (e.g., IP address).	Normal	TSF
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	TSF
FMT_MOF.1(1)	None.	None.	N/A	
FMT_MOF.1(2)	None.	None.	N/A	
FMT_MOF.1(3)	None.	None.	N/A	
FMT_MOF.1(4)	None.	None.	N/A	
FMT_MOF.1(5)	None.	None.	N/A	
FMT_MTD.1	None.	None.	N/A	
FMT_SMF.1	None.	None.	N/A	
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	TSF
FPT_FLS.1	Invocation of failures under This requirement.	Indication that the TSF has failed with the type of failure that occurred.	Normal	TSF
FPT_KST_EXT.1	None.	None.	N/A	
FPT_KST_EXT.2	All unauthorized attempts to use TOE secret and private keys.	Identifier of user or process that attempted access.	Normal	TSF

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	TSF
FPT_SKP_EXT.1	None.	None.	N/A	
FPT_STM.1	Changes to the time.	The old and new values for the time.	Normal	OE
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	
FPT_TUD_EXT.1	Initiation of update.	Version number.	Extended	OE
FPT_ITT.1	None.	None.	N/A	
FTA_SSL.4	The termination of an interactive session.	None.	Normal	TSF
FTA_TAB.1	None.	None.	N/A	
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	OE
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	OE

Security Audit (FAU)

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 in Table 4 that require persistent storage*].

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.

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 alias*] associated with the event.

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, on an external IT entity using a trusted channel protocol defined in FTP_ITC.1*].

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.

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) [*Specifically defined auditable events listed in Table 4]*].

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 4*].

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.

FAU_STG.4 Prevention of Audit Data Loss

FAU_STG.4.1 **Refinement**: The TSF shall [*prevent audited events, except those taken by the **Auditor***] and [*display a relevant error message*] if the audit trail **cannot be written to**.

FAU_SCR_EXT.1 Certificate Repository Review

FAU_SCR_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 [***valid from, valid to***].

Communications (FCO)

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 a 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)*, [***OCSP(RFC 8954)***]] 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 via* [*EST using optional "full CMC" functionality in accordance with FIA_ESTS_EXT.1*], [*support manual processes for revocation requests and responses*]].

FCO_NRR_EXT.2 Certificate-Based Proof of Receipt

FCO_NRR_EXT.2.1 The TSF shall provide proof of receipt for [*EST*] by providing signed responses using mechanisms in accordance with [*FIA_ESTS_EXT.1*].

Cryptographic Support (FCS)

FCS_CDP_EXT.1(1) Cryptographic Dependencies(TSF)

FCS_CDP_EXT.1.1(1) The TSF shall [*implement cryptographic functionality*] in order to perform [*FCS_CKM.1, FCS_CKM_EXT.1(1), 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(TSF), FCS_STG_EXT.1] cryptographic operations.

FCS_CDP_EXT.1(2) Cryptographic Dependencies(OE)

FCS_CDP_EXT.1.1(2) The TSF shall [*invoke interfaces provided by the Operational Environment*] in order to perform [*FCS_CKM.1, FCS_CKM.2, FCS_CKM_EXT.1(2), FCS_CKM_EXT.4, FCS_CKM_EXT.5, FCS_CKM_EXT.7, FCS_CKM_EXT.8, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1(OE), FCS_STG_EXT.1*] cryptographic operations.

FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1 **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;*
- *ECC schemes using "NIST curves" [selection: 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 [***equivalent to or greater than a symmetric key strength of 112 bits***] ~~that meet the following: [assignment: list of standards].~~

FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1 **Refinement**: The TSF shall **[*invoke interfaces provided by the Operational Environment to perform*****] key establishment** in accordance with a specified cryptographic key **establishment** algorithm **[**

- ***RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2;***
- ***Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair- Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";*** **]**

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

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(TSF))*

]

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*] [[*256-bit*] *symmetric KEKs using* [
• *an RBG that meets this profile (as specified in FCS_RBG_EXT.1(OE)), a key generation capability of the Operational Environment,*

]]].

FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1 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 The TSF shall [*destroy*] all plaintext keying material cryptographic security parameters when no longer needed.

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 [*keyed hashes (in accordance with FCS_COP.1(4))*].

FCS_CKM_EXT.5.2 The [*keyed hash*] used to protect a public key shall be verified upon each access to the key.

FCS_CKM_EXT.7 Key Generation for KEKs

FCS_CKM_EXT.7.1 The [*Operational environment*] shall support a hardware protected REK generated in accordance with FCS_CKM_EXT.1.1(2).

FCS_CKM_EXT.7.2 A REK shall not be able to be read from or exported from the hardware.

FCS_CKM_EXT.7.3 The TSF shall be able only to request encryption/decryption by the key and shall
not be able to read, import, or export a REK.

FCS_CKM_EXT.7.4 A REK shall be generated [by a RBG in accordance with FCS_RBG_EXT.1(OE)].

FCS_CKM_EXT.8 Key Hierarchy Entropy

FCS_CKM_EXT.8.1 The TSF shall provide a traceable hierarchy of keys (DEKs or KEKs) formed from
combinations or by encrypting one key with another 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.

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

FCS_COP.1.1(1) **Refinement**: The TSF shall **[*perform, invoke interfaces in the operational environment to perform*]** [*encryption and decryption*] in accordance with a specified cryptographic algorithm: **[**

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

]
and cryptographic key size **[*256-bit*]** ~~that meet the following:~~ ~~[assignment: list of standards]~~.

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

FCS_COP.1.1(2) **Refinement:** The TSF shall **[*perform, 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 [2048 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")***

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

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

FCS_COP.1.1(3) **Refinement**: The TSF shall [***perform, invoke interfaces in the operational environment to 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"*].

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

FCS_COP.1.1(4) **Refinement**: The TSF shall [***perform, invoke interfaces in the operational environment to perform***] [*keyed hash message authentication*] in accordance with a specified cryptographic algorithm **HMAC-[*SHA-1, SHA-256, SHA-384, SHA-512*], key size [512, 1024] bits, and message digest sizes [*160, 256, 384, 512*] bits** that meet the following: [*FIPS Pub 198-1,"The Keyed Hash Message Authentication Code"; FIPS Pub 180-4, "Secure Hash Standard"*].

FCS_RBG_EXT.1(1) Cryptographic Random Bit Generation(TSF)

FCS_RBG_EXT.1.1(1) The TSF shall [*perform*] all deterministic random bit generation (RBG) services in accordance with NIST Special Publication 800-90A using [*CTR_DRBG (AES)*].

FCS_RBG_EXT.1.2(1) The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [*Operational Environment-based noise source*] with a minimum of [*128 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.

FCS_RBG_EXT.1(2) Cryptographic Random Bit Generation(OE)

FCS_RBG_EXT.1.1(2) The TSF shall [*invoke interfaces in the operational environment to perform*] all deterministic random bit generation (RBG) services in accordance with NIST Special Publication 800-90A using [*CTR_DRBG (AES)*].

FCS_RBG_EXT.1.2(2) The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [*Operational Environment-based noise source*] with a minimum of [*128 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.

FCS_STG_EXT.1 Cryptographic Key Storage

FCS_STG_EXT.1.1 Persistent private and secret keys shall be stored within the [*TSF, Operational Environment*] [

- *encrypted within a hardware rooted key hierarchy established in accordance with* [*FCS_CKM_EXT.1(2)*], FCS_CKM_EXT.7, and FCS_CKM_EXT.8,
- i*n a hardware cryptographic module*].

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.

FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1 The TSF shall implement [*TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] supporting the following ciphersuites: [

- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*

] and no other ciphersuite.

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 [*perform RSA key establishment with key size [2048 bits, 3072 bits, 4096 bits,*]; *generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie-Hellman parameters of size 2048 bits and [3072 bits]*].

FCS_TLSS_EXT.2 TLS Server Protocol with Mutual Authentication

FCS_TLSS_EXT.2.1 The TSF shall implement [*TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346*)] supporting the following ciphersuites: [

- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*

] and no other ciphersuite.

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 [*perform RSA key establishment with key size [2048 bits, 3072 bits, 4096 bits,*]; *generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie-Hellman parameters of size 2048 bits and [3072 bits]*].

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.2.6 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.

FCS_TLSC_EXT.2 TLS Client Protocol with Mutual Authentication

FCS_TLSC_EXT.2.1 The TSF shall implement [*TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] supporting the following ciphersuites: [

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*

] .

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.2.3 The TSF shall establish a trusted channel only if the peer certificate is valid.

FCS_TLSC_EXT.2.4 The TSF shall [*present the Supported Elliptic Curves Extension with the following NIST curves: [*secp256r1, secp384r1, secp521r1*] and no other curves*] in the Client Hello.

FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

User Data Protection (FDP)

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 IETF 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:
    1. subjectKeyIdentifier
    2. authorityKeyIdentifier
    3. basicConstraints
    4. keyUsage
    5. extendedKeyUsage
    6. certificatePolicy
h. A subject field containing a null Name (e.g., a sequence of zero relative distinguished names) is accompanied by a 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_RBG_EXT.1.

FDP_CER_EXT.2 Certificate Request Matching

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

FDP_CER_EXT.3 Certificate Issuance Approval

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

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, RFC 8954*]].

FDP_CSI_EXT.1.2 The TSF shall support the approval of changes to the status of a certificate by [*CA operations staff*].

FDP_RIP.1 Subset Residual Information Protection

FDP_RIP.1.1 **Refinement**: The TSF **and [*Operational Environment*]** shall ensure that any previous information content of a resource is made unavailable upon the [*deallocation of the resource from*] the following objects: [*passwords, secret keys*].

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.

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 the standards specified in FDP_CSI_EXT.1. At a minimum, the following items shall be enforced:

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.

FDP_ITT.1 Basic Internal Transfer Protection

FDP_ITT.1.1 **Refinement:** The TSF shall ~~enforce the [assignment: *access control SFP(s) and/or information flow control SFP(s)*] to~~ prevent the [*disclosure*, *modification*] of user data when it is transmitted between physically separated parts of the TOE **through the use of [*TLS, TLS/HTTPS*]**

FDP_STG_EXT.1 Public Key Protection

FDP_STG_EXT.1. The TSF shall use [*access controlled storage*, *an integrity mechanism*] to protect the trusted public keys and certificates (trust store elements) used to validate local logon, trusted channel, and external communication to the CA.

Identification and Authentication (FIA)

FIA_X509_EXT.1 Certificate Validation

FIA_X509_EXT.1.1 The TSF shall [*validate*, *interface with the Operational Environment to 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, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.
- The TSF shall validate the revocation status of the certificate using [*OCSP as specified in RFC 6960, CRL as specified in RFC 5280 and refined by RFC 8603*].
- The TSF shall validate the extendedKeyUsage (EKU) 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) in the extendedKeyUsage field.
  - 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 EKU 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 EKU field.
- Delegated OCSP signer's certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
- Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU 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.

FIA_X509_EXT.2 Certificate-Based Authentication

FIA_X509_EXT.2.1 The TSF shall [*interface with the Operational Environment to use*] X.509v3 certificates as defined by RFC 5280 to support [*TLS, HTTPS*], and [*no additional uses*].

FIA_X509_EXT.2.2 When the TSF cannot determine the current revocation status 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.

FIA_UAU_EXT.1 Authentication Mechanism

FIA_UAU_EXT.1.1 The TSF shall [*interface with the OE to provide*] a [*certificate based authentication mechanism*]] to perform privileged user authentication.

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;
- [*no other actions*].

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, [*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.

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 simple 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.1.

FIA_ESTS_EXT.1.3 The TSF shall authenticate EST clients for initial enrollment and for supplemental authentication via [*TLS certificate-based mutual authentication in accordance with RFC 7030 section 3.3.2 and FCS_TLSS_EXT.1*].

FIA_ESTS_EXT.1.4 The TSF shall authorize EST clients based on [***[policy used by the TOE to determine client authorization in accordance with RFC 7030 section 3.7]***].

Security Management (FMT)

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 and remotely; (OE)**
2. **configure the audit mechanism; (TOE)**
3. **configure and manage certificate profiles; (TOE)**
4. **modify revocation configuration; (TOE)**
5. **perform updates to the TOE; (OE)**
6. **perform on-demand integrity tests; (OE)**
7. **import and remove X.509v3 certificates into/from the Trust Anchor Database; (TOE) [**
8. ***import [secret and private keys other than the CA's signing keys];***
9. ***configure certificate revocation list function;***
10. ***configure OCSP function;***
11. ***disable deprecated algorithms;*** **]**

to [*Administrators*].

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

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

1. **approve and execute the issuance of certificates (TOE);**
2. **configure subscriber self-service request constraints (OE); [**
3. ***no other function]***

to **[*CA Operations Staff, RA Staff*].**

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

FMT_MOF.1.1(3) **Refinement**: The [***TSF, Operational Environment***] shall restrict the

ability to

1. **approve certificate revocation (TOE); [**
2. ***no other function*** **]**

to **[*CA Operations Staff*].**

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

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

1. **perform destruction of sensitive data when no longer needed (OE); [**
2. ***participate as a second party for archival and recovery (OE);***
3. ***import a key share to support recovery of a CA signing key (OE);***
4. ***perform encrypted export of private or secret key or critical data (OE)***

] to **[*Administrators*].**

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

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

- **Delete entries from the audit trail (OE) [**
- ***Search the audit trail (TOE)***
- ***Set or change the retention period parameter for audit records requiring extended retention (OE)***

] to **[***auditors***].**

FMT_MTD.1 Management of TSF Data

FMT_MTD.1.1 The TSF shall restrict the ability to manage the TSF data to [*privileged users*].

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 and remotely (OE);*
2. *Ability to perform updates to the TOE (OE);*
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 (OE);*
10. *Ability to perform on-demand integrity tests (OE);*

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 modify the CRL configuration, Ability to modify the OCSP configuration] (TOE);*
14. *Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1 (TOE);*
15. *Ability to configure the cryptographic functionality (TOE);*
16. *Ability to import private keys (TOE);*
17. *Ability to export TOE private keys (not for archival) (TOE);*
18. *Ability to disable deprecated algorithms (TOE);*
19. *]*

].

FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 **Refinement**: The TSF **and [*no other component***] shall maintain the roles: [

- *Administrator,*
- *Auditor,*
- *CA Operations Staff,*
- ***Security Officer***[1]

]

FMT_SMR.2.2 **Refinement**: The TSF and [***no other component***] shall be able to associate users with roles.

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.

Protection of the TSF (FPT)

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: [*integrity test failure, database or HSM failures*]].

[1] PP] selection was refined as the TOE manages Security Officer role and none of the ones that are listed in the selection.

FPT_KST_EXT.1 No Plaintext Key Export

FPT_KST_EXT.1.1 The TSF and [*Operational Environment*] shall prevent the plaintext export of [all keys used by the TSF].

FPT_KST_EXT.2 TSF Key Protection

FPT_KST_EXT.2.1 The TSF and [*Operational Environment*] shall prevent unauthorized use of all TSF private and secret keys.

FPT_RCV.1 Manual Trusted Recovery

FPT_RCV.1.1 After [*External module unavailable (database, HSM, NTP)*] the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_SKP_EXT.1 Protection of Keys

FPT_SKP_EXT.1.1 The TSF shall [*implement, 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).

FPT_STM.1 Reliable Time Stamps

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

FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The TSF shall [*interface with the Operational Environment to implement*] the ability to check for updates and patches to the TOE.

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

FPT_TUD_EXT.1.3 The TSF shall [*interface with the Operational Environment to 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 [*interface with the Operational Environment to 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 [*inform the Administrator*].

Application Note: The digital signature mechanism referenced in the third element is the one specified in FCS_COP.1(2).

FPT_TST_EXT.2 Integrity Test

FPT_TST_EXT.2.1 The TSF shall apply a [*keyed hash according to FCS_COP.1(4)*] to the [*Trust Anchor Database element(s), TSF keys used to manage certificates, certificate database*].

FPT_TST_EXT.2.2 Integrity shall be verified at [*power-up*].

FPT_ITT.1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1 **Refinement:** The TSF shall protect TSF data from [*disclosure, modification*] when it is transmitted between separate parts of the TOE **through the use of [*TLS*]**.

TOE Access (FTA)

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.

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.

Trusted Path/Channels (FTP)

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 [*modification, 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*].

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 network based IT entities supporting the following capabilities: [*audit server, external cryptographic module, RA, [OCSP module]*]** 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 TSF, 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 [*store and query audit records, use of crypto functionality provided by the HSM, respond to certificate requests, check certificate status*].

5.3. Security Functional Requirements dependencies

Some SFR dependencies are satisfied with extended components instead of CCP2 SFRs.

SFR	Dependency	Extended Component
FAU_GEN.2, FMT_SMR.2	FIA_UID.1	FIA_UIA_EXT.1
FAU_STG.4	FAU_STG.1	FAU_STGEXT.1
FCS_CKM.1, FCS_CKM.2, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4)	FCS_CKM.4	FCS_CKM_EXT.4
FDP_ITT.1.1	FDP_IFC.1	FCS_HTTPS_EXT.1, FCS_TLSS_EXT.2, FCS_TLSC_EXT.2

5.4. Security Assurance Requirements

The security assurance requirement level is EAL 4 augmented by ALC_FLR.3.
EAL 4 was chosen because the developer wanted to have higher assurance than [CA_PP] requires. EAL4 was chosen since it is best suited to address the stated security objectives of the TOE. EAL4 challenges vendors to use best (rather than average) commercial practices, and at the same time it allows the vendor to evaluate their product at a detailed level while benefitting from the Common Criteria Recognition Agreement, which is a recognised agreement in many countries of the world. The chosen assurance level appropriately challenges the threats defined in the TOE environment. At EAL4, penetration testing is performed by the evaluator assuming an attack potential of Enhanced-Basic.
ALC_FLR.3 augmentation was added to introduce the flaw remediation capabilities offered by the Developer.

6. TOE Summary Specification

6.1. Security Audit

TOE generates different types of logs:

- Application Logs: generated by the TOE for troubleshooting, monitoring etc. These logs are not protected cryptographically. These logs are generated on file system and allow rotations based on the size and date. These logs can be backed up by IT team on some external storage.

- Audit Logs: generated by the TOE related to security related events i.e. Certificate generation etc. These logs are cryptographically protected with HMAC key. Each log entry contains the complete details of the audit data e.g. user who requested the certificate, operator who updated system configurations etc.
- System Logs: generated by the environment supporting TOE e.g. Operating system, HSM, RDBMS etc. These logs are also not protected and lie outside the scope of TOE.

Audit logs gives the evidence of the relevant operation performed. Each log entry contains the following information:

- Requestor who initiated the event
- Type of operation performed (Certificate generation, certificate revocation etc.)
- Result status (Success/Failed/Pending etc.)
- Request/Response Time (PKI Server time is synched with trusted time source i.e. NTP Server)
- User information etc.

As stated above all the audit logs are stored in ADSS PKI Server database and each record is protected with HMAC calculated on each row to ensure integrity of that record. The HMAC symmetric key is securely held in the CM.
Audit logs can be exported to be viewed later by the Auditor. These logs can be signed with cryptographic keys for integrity and confidentiality purposes. During import operation, signature is verified and then imported in PKI Server for viewing.
In case of Audit trail is full or the database is not available the TOE receives an error and rolls the previous transaction back that means no operation can be finished without being audited. In case of the database is not available the TOE stops operating and starts polling it. As soon as the database become available the TOE will be up again. In case the audit trail is full, the Auditor can backup the Audit data on the local drive digitally signed and can remove the already backed up records from the database freeing up some space.
Each audit record contains date and time of the event (using reliable timestamp), type of event, subject identity (the identity of the user that caused the event if applicable, i.e. an identified user initiated the event), and the outcome (success or failure) of the event. The audit trail does not include any data which allows the retrieval of sensitive data.
All audit data is searchable on Admin Console for Auditors based on many different attributes for example “certificate alias”.
The data that are audited are listed in Table 4.
There are no ephemeral keypairs generated in the system.
FAU_ADP_EXT.1, FAU_SAR.1, FAU_SAR.3, FAU_STG_EXT.1, FAU_GCR_EXT.1.1, FAU_GEN.1, FAU_GEN.2, FAU_STG.4
The certificates are stored in the database and protected with HMAC. TSF uses JDBC interface to communicate with database.
FAU_GCR_EXT.1.1, FAU_SCR_EXT.1

6.2. Communication

TOE or HSM digitally signs the certificates, CRLs and OCSP responses to provide proof of origin using RSA or ECDSA keys. TOE verifies proof of origin for certificate requests it receives using mechanism in accordance with FIA_ESTS_EXT.1.

The TSF shall provide proof of receipt by providing signed responses using mechanism in accordance with FIA_ESTS_EXT.1.
FCO_NRO_EXT.2, FCO_NRR_EXT.2 (A certificate can be revoked either via API call on its EST interface of TOE or by a privileged operator on TOE GUI web interface).

6.3. Cryptographic Support

For cryptographic operations the TOE can use one of the following Cryptography Modules: Utimaco, Thales Luna, Entrust nCipher nSheild or even cloud HSMs like Azure Key Vault or AWS CloudHSM.
For Crypto operations that are implemented by the TOE and CM is not invoked the TOE uses IAIK-JCE Cryptographic Service Provider. IAIK provides its own implementation of java.security.SecureRandom and uses entropy source of the underlying operating system.
FCS_CDP_EXT.1, FCS_RBG_EXT.1

The list of ADSS PKI Server Keys are as follows:

- KEK (Key Encrypting Key):
  - should be generated in HSM (Algorithm = AES, Size = 256).
  - it is used to encrypt/decrypt DEK.
- DEK (Data Encrypting Key):
  - always generated by TSF and stored in DB.
  - encrypted with KEK. (Algorithm = AES, size = 256).
  - used to encrypt/decrypt passwords and keys that are stored in DB.
- HMAC
  - can be generated by both HSM and TSF.
  - When generated by TSF its stored in DB
  - Algorithm: HMACSHA1, HMACSHA256, HMACSHA384, HMACSHA512
  - size = 512
  - used to compute HMAC on database records to ensure their integrity.
- TLS Server Authentication Key
  - generated by TSF and stored in database.
  - Algorithm: RSA (1024, 2048, 3072, 4096, 8192), ECDSA (160, 192, 224, 256, 384, 521))
  - It is the ADSS TLS Server Key
- TLS Client Authentication Key
  - can be generated by both HSM and TSF
  - if generated by TSF then stored in database
  - Algorithm: RSA (1024, 2048, 3072, 4096, 8192), ECDSA (160, 192, 224, 256, 384, 521).
  - It can be used when ADSS communicates with an external ADSS Server or system as a client.
- Log Signing Key
  - can be generated by both HSM and TSF
  - if generated by TSF then stored in database.
  - Algorithm: RSA (1024, 2048, 3072, 4096, 8192), ECDSA (160, 192, 224, 256, 384, 521)).
  - it is used to sign the logs that we archived from database on file system.
- Other asymmetric key pairs: These include
  - a CA key that is generated in ADSS and then this CA is used to issue certificates.

    - these keys can also be generated by HSM or TSF. If generated by HSM then its also stored inside the HSM. If generated by TSF, then its stored in the TSF database encrypted with DEK.
    - Algorithm: RSA (1024, 2048, 3072, 4096, 8192), ECDSA (160, 192, 224, 256, 384, 521))
- Master Key:
    - This can be generated by HSM or TSF. If generated by TSF then there are two possible options available for storage 1) TSF splits the master key into three parts and stores at secret locations of the disc 2) the TSF splits the key using an M of N scheme where key is divided into N parts and TSF asks N number of operators to take backup of each part and each part is encrypted using the PKBDF2 algorithm where each operator provides its own password. Nothing is stored on the disc in this case but during each startup of TSF M number of operators have to provide their backup part of the key to TSF and TSF combines these parts to create a key.
    - Algorithm = AES, Size = 256.
    - used to encrypt database password, adss.keystore password.

FCS_CKM.1, FCS_CKM.2, FCS_CKM_EXT.1(1), FCS_CKM_EXT.1(2), FCS_CKM_EXT.7, FCS_CKM_EXT.8, FCS_STG_EXT.1
TOE clears secret keys (generated by TOE itself) from memory when no longer needed by destruction of the reference of the key directly followed by a request for garbage collection.
FCS_CKM_EXT.4
Public keys are all stored in the database and protected by HMAC.
FCS_CKM_EXT.5
The list of ADSS PKI Server Crypto functions are as follows:

- AES encryption
    - KEK: It uses AES to encrypt/decrypt DEK (Algorithm = AES, Size = 256, mode=AES-CBC)
    - DEK: It uses AES encryption to encrypt/decrypt passwords/keys in ADSS database (Algorithm=AES, Size=256, mode=AES-CBC)
    - TLS/HTTPS session keys: During TLS communication an AES key is generated to encrypt/decrypt the traffic. (Algorithm=AES, Size=128/256, mode=AES-CBS/AES-GCM)
- Signing
    - Certificates signed by CA keys
    - TLS Session keys: RSA-based key establishment schemes and Elliptic curve-based key establishment schemes are used for cipher suites used for TLS/HTTPS communication both as sender and recipient. RSA-based key establishment schemes, elliptic curve-based key establishment schemes are used for digital signature generation and verification. The RSA and ECDSA keys are generated according to FCS_CKM.1.
    - Log signing key
    - If the key is generated by the TSF, signature is performed by the TSF. If the keys are generated in the HSM, signature is also performed by HSM
    - Supported Algorithms:
        - RSA Algorithm with key sizes (2048, 3072, 4096 and 8192)
        - ECDSA Algorithm with key sizes (256, 384, 512)
- Hashing

    - Hashing is performed before each signing operation (certificate sign, CRL sign, TLS, log sign etc…)
    - If the key is generated by TSF, hashing is also done by the TSF. If the key is generated by the HSM, hashing is also performed by HSM
    - The hashing algorithms SHA-1, SHA-256, SHA-384, SHA-512 and message digest sizes 160, 256, 384, and 512 bits that meet the following: FIPS Pub 180-4, “Secure Hash Standard” are supported
- Keyed-Hash Message Authentication
    - If HMAC key is generated in the TOE, TOE performs the HMAC operation, if HMAC key is generated in the HSM, HSM performs the HMAC.
    - The HMAC algorithms HMAC-SHA-1, HMAC-SHA-256, SHA-384, and SHA-512, and key sizes 512, 1024 and message digest sizes 160, 256, 384, and 512 bits that meet the following: FIPS Pub 198-1,“The Keyed Hash Message Authentication Code”; FIPS Pub 180-4, “Secure Hash Standard” are supported

FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4)
The TOE provides all its users with an HTTPS interface to configure the TOE. The user authenticates over the HTTPS interface using an x509 certificate. The user’s browser sends the user’s X.509 TLS client certificate to the TOE during the TLS negotiation.
If the TOE finds the user’s certificate trustworthy (i.e., valid and chaining to a trusted root), then the TOE establishes the TLS session and TOE permits further attempted user actions depending upon that user's authorization.
FCS_HTTPS_EXT.1
FCS_TLSS_EXT.1, FCS_TLSS_EXT.2 - The TOE acts as a TLS Server when servicing requests from administrators on its Console and when servicing requests from subscribers on its Service instance. TOE denies connections attempting to use TLS1.0 , SSL2.0 and SSL 3.0. The TOE supports *RSA key establishment with key size [2048 bits, 3072 bits, 4096 bits,*]; *generates EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1, secp521r1] and generates Diffie-Hellman parameters of size 2048 bits and [3072 bits]*]. TOE supports mutual authentication of TLS clients using X.509 certificates. The TOE compares the expected reference identifier (Subject DN) to the value found in the presented certificate. The supported cipher suites are listed in the SFR.
FCS_TLSC_EXT.2 - The TOE acts as a TLS client while communicating with external CAs (when current TOE is an intermediate CA and an external TOE hosted as some upper CA or Root CA). The TOE supports reference identifiers of Common name, DNS name, and IP addresses, but does not support wildcards in the reference identifiers. The TOE does not support or utilize certificate pinning. The supported cipher suites are listed in the SFR. The TOE presents the Supported Elliptic Curves Extension with the following NIST curves: [secp256r1, secp384r1, secp521r1] in the Client Hello by default and no configurations are required for it.

6.4. User Data Protection

The TOE provides administrators a customizable framework to apply policies for incoming certificate requests and to control the input request types and output certificate types; these are called certificate profiles.

Certificate profiles set the required information for certificate enrollment. Every certificate request received by the TOE must specify an applicable certificate profile to be used to issue that certificate.

The TOE ensures that a certificate-requesting subject possesses the applicable private key by requiring the certificate request be signed using the applicable private key.

Additionally, the TOE ensures that when issuing certificates, the serialNumber field contains at least 20 random bits drawn from the TOE using IAIK-JCE library.
The TOE supports the approval of certificates issued according to a configured certificate profile through the web interface, the "CA Operations staff" can approve certificates via the web interface (which is the only interface through which manual issuance occurs).

The TOE provides certificate status information through published CLRs and by responding to OCSP requests.

Published CRLs comply with ITU-T Recommendation X.509v2 CRL, while OCSP support complies with RFC 6960.

The TOE allows revocation of a certificate through an EST request and through the HTTPS interface (Admin Console).
Each certificate request is identified by a unique ID and certificate serial number is also stored against this request to create a linkage between the request and issued certificate.
FDP_CER_EXT.1, FDP_CER_EXT.2, FDP_CER_EXT.3, FDP_CSI_EXT.1

The TOE supports CRL generation on demand or on schedule depending on its configuration. Issued CRLs contain values in accordance with ITU-T Recommendation X.509. The TOE supports v2 CRLs and supports all of the PP-reference values.

The TOE responds to an OCSP query with a response that includes fields defined by RFC 6960 and 8954 and which include at least: version, signature Algorithm, this Update, produced At, and cert Status.

TOE clears any data buffers that contains sensitive information (passwords, secret keys) after their use.

FDP_RIP.1, FDP_CRL_EXT.1, FDP_OCSPG_EXT.1
The TSF prevents the disclosure and modification of user data when it is transmitted between physically separated parts of the TOE through the use of TLS and TLS/HTTPS.
FDP_ITT.1
TOE stores trusted public keys and certificates in its database and integrity is protected using HMAC. The certificates are loaded using web GUI interface available to privileged operators and public key integrity is checked during each access of the key.
TOE stores audit data in database and uses TLS to communicate with database. The audit data integrity is protected using HMAC.
FDP_STG_EXT.1, FAU_STG_EXT.1
The TOE protects TSF data in transit between separate parts by TLS and HTTPS/TLS.
FPT_ITT.1.1

6.5. Identification and Authentication

Every external application or operator (Administrator, Auditor, Security Officer, or CA Operations Staff) is authenticated over TLS client authentication which uses PKI signature mechanism and doesn’t require any password.
Authentication is not required for following:

- responding to OCSP request when open access is granted

Identification is not required for following:

- responding to OCSP request when open access is granted

User security attributes are associated to the TOE user identity and authentication credentials, allowing a unique match.

- For Administrator/Auditor/Security Officer/CA Operations Staff: first name, last name, Operator ID, Email Address, Mobile Number, Role and its TLS Client Authentication Certificate.
- For External Applications (External RAs & Business applications): Client ID (Unique for each client), Friendly Name, Phone No, Email Address, Address.

In case of a failed authentication attempt the TOE counts the failed attempts for each Business Application and if it reaches a configured number the TOE blocks the Business Application access for a configured period.

The TOE validates the certificates itself and it will validate the following:

- the certificate is not revoked, the certificate has a valid certificate path from the peer's certificate through any intermediary CAs to a root CA trusted by the TOE,
- the current time is within the certificate's validity period, and the certificate includes extensions that are required or consistent with the current use (e.g., CA flag value, KU and EKU).

Certificate revocation status is performed by getting the revocation status of all certificates (starting with the root certificate and working down the chain).

The TOE will reject any certificate for which it cannot determine the revocation status and hence not accept the connection attempt if a revocation status cannot be obtained.

The TOE uses X.509v3 certificates, as defined by RFC 5280, to authenticate privileged users over HTTPS, and to verify the integrity of software updates.

Certificates can also be used for EST access authentication. When the TSF cannot determine the current revocation status of a certificate the TOE rejects the certificate and don't accept the connection request.

FIA_X509_EXT.1, FIA_X509_EXT.2

The TOE uses X.509v3 certificates, as defined by RFC 5280, to authenticate privileged users over HTTPS, and to verify the integrity of software updates.

Certificates can also be used for EST access authentication.

When the TSF cannot determine the current revocation status of a certificate the TOE rejects the certificate and don't accept the connection request.

FIA_UAU_EXT.1, FIA_UIA_EXT.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,

- Enrollment over Secure Transport (EST) in accordance with FIA_ESTS_EXT.1

6.6. Security Management

FMT_MOF.1(1-5), FMT_MTD.1, FMT_SMF.1, FMT_SMR.2 – The Administrative functions of the TOE are available on the ADSS PKI Server Admin Console that is restricted. Operators have access to its web GUI after certificate-based authentication. After identifying the user the user's associated role is identified and only the relevant functions will be available for them. The relevant functions based on their roles are listed in FMT_MOF.1(1-5). The ADSS PKI Server Admin Console is not available for non-administrative users like client services or client applications.

6.7. Protection of the TSF

In case of OE failures including if database or HSM is down, ADSS PKI Server detects it and stops the services and sends alerts to the operators. The alerts are configurable (e-mail, sms etc..). Operators can make DB/HSM/NTP up again fixing the problem. ADSS PKI Server can automatically detect if DB/HSM/NTP become available, so it restarts its services automatically. Operator can also manually restart ADSS but it's not necessary. Until all required OE entities (or even TOE entities) are up and running TSF will not be available so no operations can be done. The sensitive data are all encrypted in the DB, the HSM keys are also not exportable so in case of any failure of DB or HSM the TSF data is safe.

FPT_FLS.1, FPT_RCV.1
ADSS PKI Server has its KEK (Key Encrypting Key) generated and stored in HSM. KEK is not exportable so never leaves the HSM. DEK (Data Encrypting Key) is always generated by the TOE for performance issues and stored in DB encrypted by KEK. DEK is never exported in plain text, only in encrypted format for example during DB backup. Any other keys of the TOE are stored encrypted by DEK.
FPT_KST_EXT.1, FPT_SKP_EXT.1
Unauthorized use of TSF private and secret keys is not possible as each user (Operator or client application) must authenticate themselves using their certificates. The user/client associated roles guarantees that no unauthorized entities can have access to any of the private keys.
FPT_KST_EXT.2
Each TSF related operations are audited so basically each TSF operation uses time. TOE provides reliable timestamps based on externally configured NTP servers. NTP server synchronizes system clock in a specified frequency. If the synchronization fails TSF stop working until NTP is up again just like in case of any other OE entities (DB, HSM).
FPT_STM.1
The TOE obtains the current time from its operational environment.
FPT_TUD_EXT.1
The TOE update package consists of a ZIP file that is digitally signed and verified before applying the update. The verification is done using the public key of the signing certificate according to FIA_X509_EXT.1. If the signature verification fails or the signing certificate does not contain code signing in ExtendedKeyUsage, the Administrator is informed by an error message.
FPT_TST_EXT.2
The TOE computes HMAC of DB records and a parallel thread keeps verifying the checksum. In case of the record is modified the TOE will mark it red on Admin interface, denies any operation with that record and notifies the Administrator. Also, if the verification fails the background thread stops ADSS Services.

6.8. TOE Access

The resources access in TOE is controlled using access control list based on the following:

- access rule – accept or reject the resource access i.e. Client ID registered in TOE (for external applications)
- resource – a resource to which access is controlled e.g. adss/console/certification/cert/manage_issued_certificates.do
- user – a subject that have access rights to a resource e.g. Administrator operator
- role – defines the access rules which are assigned to a user to access the resources

When a controlled resource is accessed, the PKI Server verifies that the client meets the desired access rules for the resource or not. If not, denies access and generates an error. If there are no access rules associated to the resource, access is denied. The TOE access control system maps authentication information to a user entity. The entity is then associated to a role in order to acquire privileges according FMT_SMR.2.
The roles have different access rights on different resources. Additionally, there are many resources organized in a hierarchical way, which allows for higher level access rights to be applied recursively on sub-access rights.
Before logging the user in, TOE warns the user regarding unauthorized use of the TOE.
FTA_SSL.4, FTA_TAB.1.1

6.9. Trusted Path/Channels

The TOE implements and enforces the following trusted communication methods and protocols:

- **Operators:** Operators (Administrator, Auditor, Security Officer, CA Operations Staff) access the ADSS PKI Server Admin Console GUI over a mutually authenticated TLS v1.2 channel.
- **TOE Services:** TOE communicates with all its services (eg OCSP Service) mutually authenticated TLS v1.2 channel.
- **External Services:** TOE communicates with all external client services (eg RA and Web RA, Remote Signing Service, Signature Validation Service, TSA Service) using TLS v1.2 channel.
- **CM:** The TOE communicates with CM using vendor specific APIs. User passwords do not travel on this channel. This communication with the CM is enforced by the CM to be secure, authenticated and protected from replay attacks, i.e. the CM meets the requirements of EN 419 221-5 Protection Profile and is certified to Common Criterial EAL4+ level.

FTP_TRP.1, FTP_ITC.1

7. References

Identifier	Title

[CC]	Common Criteria for Information Technology Security Evaluation – • Part 1: Introduction and General Model, CCMB-2012-09001, Version 3.1 Revision 5, April 2017 [CC1] • Part 2: Security Functional Components, CCMB-2012-09002, Version 3.1 Revision 5, April 2017 [CC2] • Part 3: Security Assurance Components, CCMB-2012-09003, Version 3.1 Revision 5, April 2017 [CC3]
[CEM]	Common Methodology for Information Technology Security Evaluation, Version 3.1 Revision 5, April 2017
[IR7924]	Second Draft NIST IR 7924, Reference Certificate Policy, May 2014
[PP_CA_V2.1]	Protection Profile for Certification Authorities Version 2.1, 2017-12-01
[SSG]	Ascertia Support Services Guide, 8.0, June 2022

8. Acronyms

Acronym	Meaning
AES	Advanced Encryption Standard
AOR	Authorized Organizational Representative
API	Application Programming Interface
CA	Certification Authority
CBC	Cipher Block Chaining
CC	Common Criteria
CCM	Counter with CBC-Message Authentication Code
CCMP	CCM Protocol
CCTL	Common Criteria Testing Laboratory
CMC	Certificate Management over CMS
CMS	Cryptographic Message Syntax
CRL	Certificate Revocation List
CSS	Certificate Status Server
DEK	Data Encryption Key
DES	Data Encryption Standard
DH	Diffie-Hellman
DHE	Diffie Hellman Key Exchange
DKM	Derived Keying Material
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
EDC	Error Detection Code
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESP	Encapsulating Security Payload (IPsec)
FFC	Finite-Field Cryptography
FIPS	Federal Information Processing Standards
GCM	Galois/Counter Mode
HMAC	Keyed Hash Message Authentication Code
HSM	Hardware Security Module
HTTPS	HyperText Transfer Protocol Secure
I&A	Identification and Authentication
IKE	Internet key Exchange
IPsec	Internet Protocol Security
IUT	Implementation Under Test

IV	Initialization Vector
KAT	Known Answer Tests
KDF	Key Derivation Function
KEK	Key Encryption Key
KW	Key Wrap
KWP	Key Wrapping with Padding
MAC	Message Authentication Code
MODP	Modular Exponential
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
NPE	Non-person Entity
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PKV	Public Key Verification
PP	Protection Profile
RA	Registration Authority
RAM	Random Access Memory
RBG	Random Bit Generator
rDSA	RSA Digital Signature Algorithm
REK	Root Encryption Key
RFC	Request for Comment
RNGVS	Random Number Generator Validation System
RSA	Rivest Shamir Adleman
SA	Security Association (IPsec)
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TPM	Trusted Platform Module
TSF	TOE Security Function
TSS	TOE Summary Specification