Protection Profile for Certification Authorities

Version: 2.0

2016-10-28

National Information Assurance Partnership

Revision History

Version	Date	Comment
V1.0	2014-05-16	Initial draft
V1.1	2016-07-07	Formatting updates and changes based on TC feedback
V1.2	2016-10-26	Updates based on additional TC feedback and internal review
V2.0	2016-10-28	Second draft

Table of Contents

1 Introduction

1.1 Overview

Certification Authorities (CAs), and the infrastructure they support, form the basis for one of the primary mechanisms for providing strong assurance of identity in online transactions. The widely placed trust in CAs is at the heart of security mechanisms used to protect business and financial transactions online. Notably, protocols using Transport Layer Security (TLS) rely on certificates issued by CAs to identify and authenticate servers and clients in web transactions. Governments around the world rely on CAs to identify parties involved in transactions with them.

However, historical high-profile security breaches at major CAs trusted by widely used operating systems and browsers have highlighted both the critical role CAs play in securing electronic transactions, as well as the need to strongly protect them from malicious attacks. Analyses have revealed that these security breaches were often the result of insufficient security controls being in place on the computer systems and networks at these CAs, and were sometimes exacerbated by weak record keeping. Third-party auditing programs, whose role it was to verify that proper security controls were in place, were not sufficient to identify these lapses in security.[1]

This Protection Profile (PP) describing security requirements for a Certification Authority is intended to provide a minimal, baseline set of requirements that are targeted at mitigating well defined and described threats. These requirements support CA operations performed in accordance with the National Institute of Standards and Technologies (NIST) Interagency or Internal Report (IR) 7924 (Second Draft), *Reference Certificate Policy*, referred to as the "NIST IR."[2] This PP represents an evolution of "traditional" Protection Profiles and the associated evaluation of the requirements contained within the document. This introduction will describe the features of a compliant TOE, and will also discuss the evolutionary aspects of the PP as a guide to readers of the document.

1.2 Terms

The following sections provide both Common Criteria and technology terms used in this PP.

1.2.1 Common Criteria Terms

Table 1 - Common Criteria Terms

Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Extended Package (EP)	An implementation-independent set of security requirements for a specific subset of products described by a PP.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.

[1] NIST IR 7924 (Second Draft), *Reference Certificate Policy,* May 2014.
[2] Ibid.

Security Assurance Requirement (SAR)	A requirement for how the TOE's proper implementation of the SFRs is verified by an evaluator.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation (TOE)	The product under evaluation. In this case, a certification authority.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in a ST.

1.2.2 Technology Terms

Table 2 - Technology Terms

Administrator	The Administrator is responsible for management activities, including configuration of the CA and its security functions.
Authorized Organizational Representative (AOR)	An optional privileged user role which is delegated authority by the Certification Authority Staff or RA Staff to manage a restricted set of certificates associated to devices belonging to a particular organization.
Certificate Profile	A set of configuration parameters that defines everything associated with a type of certificate, in particular the contents (fields and extensions) of the generated certificate.
Certification authority (CA)	The set of hardware, software, firmware, or some combination thereof, that issues, revokes, and manages public key certificates and certificate status information.
CMC	Certificate Management over CMM. A standard certificate enrollment protocol.
Compromise	The unauthorized disclosure, modification, substitution or use of sensitive data (including plaintext cryptographic keys and other CSPs).
Confidentiality	The property that sensitive information is not disclosed to unauthorized individuals, entities or processes.
Critical security parameter (CSP)	security-related information (e.g., secret and private cryptographic keys, authentication data such as passwords and PINs) appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a CA or the security of the information protected by the CA.
Cryptographic key	A parameter used in conjunction with a cryptographic algorithm that determines: • the transformation of plaintext data into ciphertext data, • the transformation of ciphertext data into plaintext data, • a digital signature computed from data, • a keyed hash computed from data, • the verification of a digital signature computed from data,

	• an authentication code computed from data, or an exchange agreement of a shared secret.
Data Encryption Key (DEK)	A key used to encrypt data-at-rest.
Digital Signature	A non-forgeable transformation of data that allows proof of the source (with nonrepudiation) and verification of the integrity of that data.
Encrypted key	A cryptographic key that has been encrypted with a key encrypting key, a PIN or a password in order to disguise the value of the underlying plaintext key.
Error detection code (EDC)	A code computed from data and comprised of redundant bits of information designed to detect, but not correct, unintentional changes in the data.
Integrity	The property that sensitive data has not been modified or deleted in an unauthorized and undetected manner.
Key encrypting key (KEK)	A key used to encrypt other keys, such as DEKs, or storage that contains keys.
Key sharing	A multi-party computation (MPC) mechanism that allows two or more parties, each with key components, to jointly produce a plaintext key without revealing any of the key components.
Private key	A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and not made public.
Privileged user	An individual with access and login privileges on the CA
Public key	A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and which may be made public. (Public keys are not considered CSPs.)
Public key certificate	A set of data that unambiguously identifies an entity, contains the entity's public key, is digitally signed by a trusted party, and binds the public key to the entity.
Public key (asymmetric) cryptographic algorithm	A cryptographic algorithm that uses two related keys, a public key and a private key. The two keys have the property that, given the public key, it is computationally infeasible to derive the private key.
Registration authority (RA)	The set of hardware, software, firmware, or some combination thereof that is used to validate the identity of a subscriber before instructing the CA to manipulate a certificate on the subscriber’s behalf.
Root encryption key (REK)	A key tied to hardware that is used to encrypt other keys such as KEKs.
Secret key	A cryptographic key used with a secret key cryptographic algorithm, uniquely associated with one or more entities, and which shall not be made public. The use of the term "secret" in this context does not imply a classification level rather the term implies the need to protect the key from disclosure or substitution.
Secret key (symmetric) cryptographic algorithm	A cryptographic algorithm that uses a single, secret key for both encryption and decryption.

Shared secret	A token used by the CMC protocol to help provide identity proofing.
Subscriber	A human or machine entity that is bound to one or more certificates maintained by the CA.
Trust Anchor Database	A list of trusted root Certification Authority certificates.

1.3 Compliant Targets of Evaluation

A CA system is an entity that issues and manages public-key certificates. The CA is the primary component of a public key infrastructure (PKI), which consists of programs, data formats, procedures, communication protocols, security policies, and public key cryptographic mechanisms working together to enable people in various locations to establish trust through secure communications. To achieve this goal, a PKI may provide some or all of the following security management services:

- Key generation/storage
- Certificate generation, modification, re-key, renewal, and distribution
- Certificate revocation list (CRL) generation and distribution
- Key escrow and recovery
- Directory management of certificate related items
- Certificate token initialization/programming/management
- System management functions (e.g., security audit, configuration management, archive)

A CA performs a number of certificate management functions besides certificate issuance:

- **Re-issuance**: A CA handles re-issuance of certificates when they expire, since certificates have a finite validity period. Reissuance may be renewal of the current public key; rekey with a new public key; or modification to other data in the public key certificate.
- **Revocation**: The CA is also responsible for indicating, when notified via a subscriber or privileged user, that a certificate should no longer be used or relied upon; this is referred to as revocation. For example, a certificate needs to be revoked if an individuals' private key is compromised or if the CA issued the certificate to the wrong person. Identifiers of revoked certificates are stored on an electronic list called a certificate revocation list (CRL). The CRL is digitally signed by the CA and published to a repository accessible by the relying parties. The CRL is used to compare against certificates to ensure a certificate is not invalid when used. Alternatively, a CA can provide a Certificate Status Service (CSS) that provides revocation status responses to subscribers and relying parties. The CSS' revocation status information may be based on certificate history information from the CA, a CRL from the CA, or a CRL retrieved from a repository. A CA must be able to provide revocation status, but either approach is acceptable.
- **Distribution**: The CA handles the publishing of certificates and CRLs that it issues to a repository. The repository enables subscribers and relying parties to obtain subscriber certificates and CRLs to perform functions such as encrypting emails and data to recipients or verifying signatures on transactions. Typically, CRL location is advertised in the certificate itself as an HTTP pointer to allow the relying parties to obtain the CRL.

- **Storage**: The CA keeps a history of a subscriber's previously issued and revoked certificates.

There are a number of optional functions that a CA may perform. For example, a CA may issue CRLs or may provide a CSS that responds to certificate status requests from subscribers and relying parties. A CA may generate public/private key pairs for subscribers, usually for encryption; this function may be delegated to a different PKI component. In some cases, a CA will escrow private keys for encryption certificates, a function typically delegated to a key escrow PKI component. If a CA handles subscriber key generation and escrow, it should also keep a history of subscriber keys to support cases where an old encryption key may be required to decrypt data. A CA may also be responsible for verifying subscriber identities who request to interact with the CA. If the CA does not provide this functionality directly, it is expected to interface with a registration authority (RA) that does.

The CA can be internal to an organization or it can be managed by an outside organization dedicated to this type of service. If the CA is internal, the organization controls the CA server, configures how the subscriber identity proofing takes place during registration, maintains the certificates, and revokes certificates when necessary. If the CA is a third party organization specifically designed to serve as a CA, then other individuals and companies pay them to supply this service. Depending on the nature of agreement and service, the organization may be fully or to some extent involved in subscriber registration, certificate management, and revocation.

1.3.1 TOE Boundary

Figure 1 below illustrates an example PKI architecture; this architecture is for illustration only and is not meant to represent requirements for an actual deployment. 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 either directly through the CA itself or optionally through a third-party RA which 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 either at the CA, at the RA or (typically) on the subscriber's system. If not generated by the CA, 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.

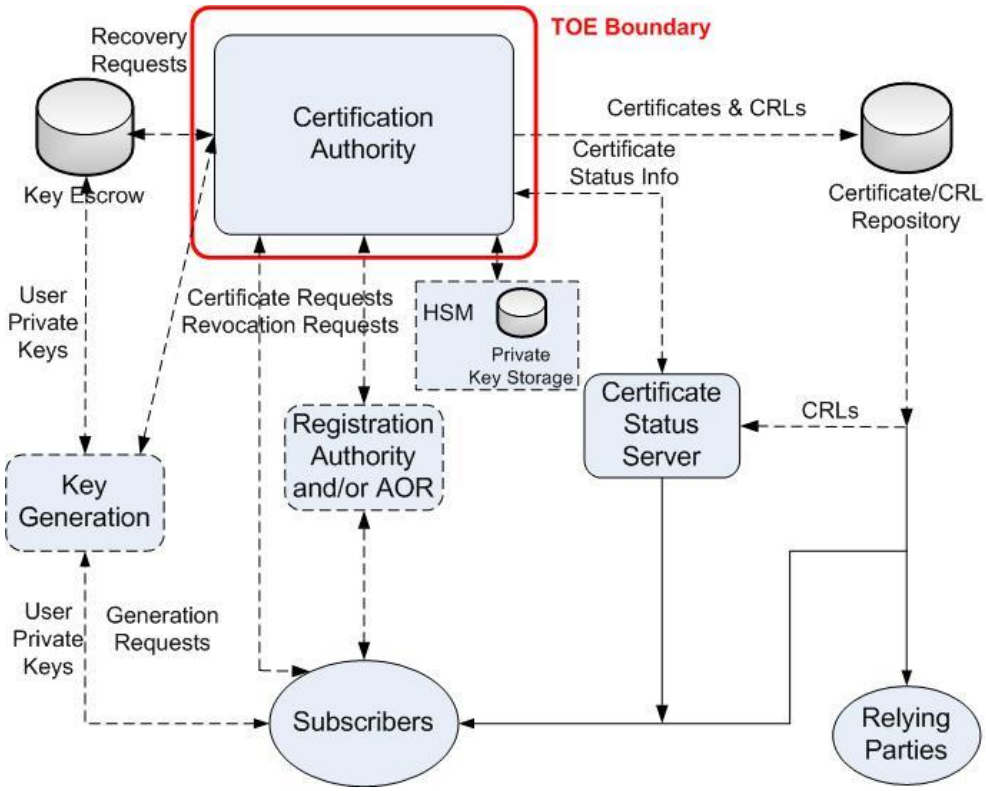

Figure 1 - TOE Boundary in Example PKI Architecture

This PP defines requirements only for CA system component(s) that issue and manage public key certificates and certificate status information, to include interfaces to components not under the control of the ST author that may be required to meet these requirements as shown in Figure 1.

While the functionality that the TOE is obligated to implement (in response to the described threat environment) is discussed in detail in later sections, it is useful to give a brief description here. Compliant TOEs will provide security functionality that addresses threats to the TOE and implements policies that are imposed by law or regulation. Compliant TOEs must authenticate and validate certificate requests and control the use of its private signature key(s) so that only valid, properly authorized certificates are issued; it must validate and authenticate all revocation requests and provide accurate and up-to-date revocation status information; and it must validate any requests for optional services (key generation, key escrow or recovery), authenticate and determine authorization for such services according to applicable security policies and ensure that only authorized services are performed. The TOE must protect itself from common network attacks, limit the damage that could occur by privileged user error, and be able to recover from damage that can occur via either network attacks or human error, to include reconstitution of functionality necessary to maintain any and all certificates issued for the duration of their validity periods in the case of TOE failure. The TOE must also offer auditing of a set of events that are associated with security-relevant activity on the TOE, and these events must be retained for long-term storage even in the event of a failure of the TOE. Audit storage should be reliable and extensible, although this could be on a device that is distinct from the TOE. The TOE must offer some protection for common network denial of service attacks and must also provide the ability to verify the source of updates to components of the TOE.

A CA system which is the Target of Evaluation (TOE) of this PP may be a software package installed on a general computing platform, a set of software packages installed on distributed general computing platforms, or an integrated device including hardware and software. This PP makes no distinction in these cases and imposes requirements on the TOE and/or Operational Environment to ensure that the requirements can be met in any of these cases. Whenever the TOE depends on external components to meet the requirements of this PP, those components are included in the Operational Environment and the AGD_OPE and AGD_PRE sections of this PP describe requirements on the TOE to document these dependencies. For example, the TOE provides cryptographic operations involved in the signing of certificates, which may depend on an external cryptographic module such as a trusted computing module (TPM) on the general computing platform or an external hardware security module (HSM).

The CA manages certificates by providing validity information, either via the issuance of Certificate Revocation Lists (CRLs) or via a Certificate Status Service (CSS) that provides real-time responses to validity queries. Because a CA acts as a trusted third party, and because recommended operations require independent monitoring of its operations, the CA must maintain an audit record that can be reviewed. This audit record may be maintained on the TOE, or on an external audit server.

The threats and security objectives apply generally to a CA system. In order to provide consistent requirements for all TOEs, the requirements in Section 5 include selections to indicate where external components may be used. The TOE platform, external cryptographic modules, external audit servers, and external CSS that are not under the control of the security target (ST) author may be used to meet the respective TOE requirements. In these cases, the ST author must provide evidence that the requirement is met by the selected component. When external components are selected, this evidence is typically via validation against an appropriate PP.

It is intended that the set of requirements in this PP is limited in scope in order to promote quicker, less costly evaluations that provide some value to end users.

1.4 Use Cases

Requirements in this PP are designed to address the security problem for CA systems. The fundamental usage of a CA system will not differ drastically based on the functionality it provides. Different TOEs may vary because of the inclusion or exclusion of the various optional, objective, and selection-based requirements defined in the annexes of this PP but they are all expected to be used in the same general manner for the same general purposes.

2 Conformance Claims

Conformance Statement

To be conformant to this PP, an ST must demonstrate Exact Conformance, a subset of Strict Conformance as defined in [CC] Part 1 (ASE_CCL). The ST must include all components in this PP that are:

- Unconditional (which are always required)
- Selection-based (which are required when certain selections are chosen in the unconditional requirements)

and may include components that are

- Optional
- Objective

Unconditional requirements are found in the main body of the document (Section 5), while appendices contain the selection-based, optional, and objective requirements. The ST may iterate any of these components but it must not introduce any additional component (e.g. from CC Part 2 or 3) that is not defined in this PP.

CC Conformance Claims

This EP is conformant to Parts 2 (extended) and 3 (conformant) of Common Criteria Version 3.1, Revision 4 [CC].

PP Claim

This PP does not claim conformance to any Protection Profile.

Package Claim

This PP does not claim conformance to any packages.

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 non-repudiation.

T.UNAUTHORIZED_ACCESS

A malicious user, process, or external IT entity intentionally circumvents TOE security mechanisms.

T.UNAUTHORIZED_UPDATE

A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.

T.UNDETECTED_ACTIONS

Remote users or external IT entities may take actions that adversely affect the security of the TOE.

T.USER_DATA_REUSE

A malicious user, process, or external IT entity may gain access to user data that is not cleared when resources are reallocated.

T.WEAK_CRYPTO

A weak hash or signature scheme may be compromised by an attacker and used to apply integrity checks to malicious content so that it appears legitimate.

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.

A.TRUSTED_PLATFORM

The operating system on which the TOE has been installed is securely configured, regularly patched, and not subject to unauthorized access.

Note that there are several capabilities where it is at the vendor's discretion as to whether they are implemented by the TSF, the Operational Environment, or a combination of the two. In the cases where the TOE relies on the OE to provide some or all of this functionality, one or more of the following assumptions may apply:

A.ADMINISTRATOR_AUTHENTICATION

The operating system on which the TOE has been installed provides a method of identifying and authenticating administrators prior to granting them access to the TOE.

A.AUDITING

The operating system on which the TOE has been installed provides a mechanism for the generation, storage, and review of audit data.

A.CRYPTOGRAPHY

The Operational Environment provides cryptographic services that can be invoked by the TSF in order to perform security functionality.

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 (selection-based)

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, FAU_STG.1(1) (selection-based), FAU_STG.1(2) (selection-based), FAU_STG_EXT.2 (selection-based)

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_CER_EXT.4 (optional), FDP_CRL_EXT.1 (selection-based), FDP_CSI_EXT.1, FDP_OCSP_EXT.1 (selection-based), FDP_SDP_EXT.1 (selection-based), FDP_STG_EXT.1, FIA_CMC_EXT.1 (selection-based), FIA_EST_EXT.1 (selection-based), FIA_X509_EXT.1, FIA_X509_EXT.2, FPT_NPE_EXT.1 (optional)

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_CER_EXT.4 (optional), FDP_CRL_EXT.1 (selection-based), FDP_OCSP_EXT.1 (selection-based), FMT_MOF.1(1), FMT_MOF.1(2), FMT_MOF.1(3), FMT_MOF.1(4), FMT_MOF.1(5) (optional), FMT_MTD.1, FPT_NPE_EXT.1 (optional)

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 (selection-based), FDP_ITT.1 (selection-based), FPT_ITT.1 (selection-based), FPT_TST_EXT.2 (selection-based)

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 (selection-based)

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(1) (selection-based), FCS_CKM.1(2) (selection-based), FCS_CKM_EXT.1(1) (selection-based), FCS_CKM_EXT.1(2) (selection-based), FCS_CKM_EXT.1(3) (selection-based), FCS_CKM_EXT.1(4) (selection-based), FCS_CKM_EXT.4 (selection-based), FCS_CKM_EXT.7 (selection-based), FCS_CKM_EXT.8 (selection-based), FCS_COP.1(1) (selection-based), FCS_COP.1(2) (selection-based), FCS_COP.1(3) (selection-based), FCS_COP.1(4) (selection-based), FCS_COP.1(5) (optional), FCS_HTTPS_EXT.1 (selection-based), FCS_IPSEC_EXT.1 (selection-based), FCS_RBG_EXT.1 (selection-based), FCS_SSHC_EXT.1 (selection-based), FCS_SSHS_EXT.1 (selection-based), FCS_STG_EXT.1, FCS_TLSC_EXT.1 (selection-based), FCS_TLSS_EXT.1 (selection-based), FDP_ITT.1 (selection-based), FIA_PSK_EXT.1 (selection-based), FPT_ITT.1 (selection-based), FPT_KST_EXT.1, FPT_KST_EXT.2, FPT_SKP_EXT.1, FPT_SKY_EXT.1, FPT_SKY_EXT.2 (selection-based), 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, FCS_CKM_EXT.6 (selection-based), 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.SESSION_LOCK

The TOE will provide mechanisms that mitigate the risk of unattended sessions being hijacked.

Addressed by: FTA_SSL_EXT.1

O.SYSTEM_MONITORING

The TOE will provide the capability to generate audit data and send those data to an external IT entity. The TOE will record in audit records: date and time of action and the entity responsible for the action.

Addressed by: FAU_ADP_EXT.1, FAU_GEN.1 (selection-based), FAU_GEN.2 (selection-based), FAU_SAR.1 (selection-based), FAU_SAR.3 (selection-based), FAU_SAR_EXT.1, FAU_SEL.1 (selection-based), FAU_STG_EXT.1 (selection-based), 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_AFL.1 (selection-based), FIA_PMG_EXT.1 (selection-based), FIA_UAU.7 (selection-based), 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) (optional), FMT_MTD.1, FMT_SMF.1, FMT_SMR.2, FPT_APW_EXT.1 (selection-based), FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4

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.

Addressed by: FPT_TST_EXT.2 (selection-based)

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 (optionally) from a trusted source.

Addressed by: FCS_CDP_EXT.1, FCS_COP.1(3) (selection-based), FIA_X509_EXT.2, FPT_TUD_EXT.1

4.2 Security Objectives for the Operational Environment

OE.NO_GENERAL_PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.TRUSTED_ADMIN

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.

Note that there are several capabilities where it is at the vendor's discretion as to whether they are implemented by the TSF, the Operational Environment, or a combination of the two. In the cases where the TOE relies on the OE to provide some or all of this functionality, one or more of the following objectives may apply:

OE.ADMINISTRATOR_AUTHENTICATION

The operating system on which the TOE has been installed provides a method of identifying and authenticating administrators prior to granting them access to the TOE.

OE.AUDITING

The operating system on which the TOE has been installed provides a mechanism for the generation, storage, and review of audit data.

OE.CRYPTOGRAPHY

The Operational Environment provides cryptographic services that can be invoked by the TSF in order to perform security functionality.

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

Threat, Assumption, or OSP	**Security Objective**
A.NO_GENERAL_PURPOSE	OE.NO_GENERAL_PURPOSE
A.PHYSICAL	OE.PHYSICAL
A.TRUSTED_ADMIN	OE.TRUSTED_ADMIN
P.ACCESS_BANNER	O.DISPLAY_BANNER
T.PRIVILEGED_USER_ERROR	O.AUDIT_LOSS_RESPONSE O.AUDIT_PROTECTION O.SESSION_LOCK O.TOE_ADMINISTRATION
T.TSF_FAILURE	O.TSF_SELF_TEST
T.UNAUTHORIZED_ACCESS	O.PROTECTED_COMMUNICATIONS O.SESSION_LOCK O.TOE_ADMINISTRATION
T.UNAUTHORIZED_UPDATE	O.VERIFIABLE_UPDATES
T.UNAUTHENTICATED_TRANSACTIONS	O.CERTIFICATES O.CONFIGURATION_MANAGEMENT O.INTEGRITY_PROTECTION O.NON_REPUDIATION
T.UNDETECTED_ACTIONS	O.AUDIT_LOSS_RESPONSE O.AUDIT_PROTECTION O.SYSTEM_MONITORING
T.USER_DATA_REUSE	O.RESIDUAL_INFORMATION_CLEARING
T.WEAK_CRYPTO	O.PROTECTED_COMMUNICATIONS O.VERIFIABLE_UPDATES

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 4*, with additional extended functional components.

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. 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)").
- **Extended SFRs**: are identified by having a label "EXT" after the SFR name.

5.1 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 4, 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.

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_EXT.1	None.	None.	N/A	
FCO_NRO_EXT.2	None.	None.	N/A	
FCS_CDP_EXT.1	None.	None.	N/A	
FAU_STG_EXT.1	None.	None.	N/A	
FDP_CER_EXT.1	Certificate generation.	Success: [*selection: Certificate value, certificate object identifier*]	Extended	
FDP_CER_EXT.2	Linking of certificate to certificate request	Success: [*selection: Certificate value, certificate object identifier*], [*selection: Certificate request, link to certificate request object identifier*]. Failure: Reason for failure,	Extended	

		[*selection: Certificate request, link to Certificate request object identifier*].		
FDP_CER_EXT.3	Failed certificate approvals.	Reason for failure. [*selection: Certificate request, link to Certificate request object identifier*].	Normal	
FDP_CSI_EXT.1	None	None	N/A	
FDP_RIP.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	
FIA_X509_EXT.1	Failed certificate validations.	None.	Normal	
FIA_X509_EXT.2	Failed authentications.	None.	Normal	
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	
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	
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_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	
FPT_FLS.1	Invocation of failures under this requirement.	Indication that the TSF has failed with the type of failure that occurred.	Normal	
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	
FPT_RCV.1	The fact that a failure or service discontinuity occurred. Resumption of the regular operation.	TSF failure types that are available on recovery	Extended	
FPT_SKP_EXT.1	None.	None.	N/A	
FPT_SKY_EXT.1	None.	None.	N/A	
FPT_STM.1	Changes to the time.	The old and new values for the time.	Normal	
FPT_TUD_EXT.1	Initiation of update.	Version number	Extended	
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.	Normal	
FTA_SSL.4	The termination of an interactive session.	None.	Normal	
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	None.	Normal	
FTA_TAB.1	None.	None.	N/A	
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	
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	

5.1.1 Security Audit (FAU)

FAU_ADP_EXT.1 Audit Dependencies

FAU_ADP_EXT.1.1 The TSF shall [*selection: implement audit functionality, interface with auditing function(s) in the Operational Environment*] in order to perform audit operations on the following audit data: [*assignment: Auditable events in Table 4 through Table 6 that require persistent storage*].

Application Note: *If the TOE relies on the Operational Environment to provide some or all of the TOE's auditing functionality, the ST author is expected to identify whether each of the auditable events for the claimed SFRs are implemented by the TOE or by the Operational Environment, along with the specific environmental component that provides the auditing functionality if applicable. The ST author should refer to the right-most column of Table 4 through Table 6 and complete these fields accordingly.*

Audit records for the TSF are divided into two sets of events, whose retention periods might be significantly different operationally. Generally, information necessary to maintain an issued certificate or to determine the circumstances of a certificate issuance is required to be available at least as long as the validity of an issued certificate, and perhaps longer according the statutes, laws, or policies applicable to the issuance and intended use of a particular certificate. Other audit data is typically retained only to support normal operations. The 'Retention' column in Table 4 (as well as Tables 5 and 6 for the optional and selection-based SFRs) indicates whether the audit record is intended to be used for 'normal' (shorter-term) or 'extended' (longer-term) purposes.

Assurance Activity

TSS

The evaluator shall examine the TSS and operational guidance in order to verify that they describe each of the relevant auditable events, how audit records of these events are formatted, and what component of the TOE or Operational Environment is responsible for handling these events.

For those auditable events that are generated by the TOE and stored within the TOE boundary, the assurance activities are included for the relevant selection-based audit SFRs.

For any auditable events that are handled by the TOE's Operational Environment, the evaluator shall either demonstrate that these events are generated and handled in a manner consistent with the Security Target or that the relevant audit function has been independently evaluated (e.g. if the auditable events were part of the TSF for an evaluation against the Protection Profile for General Purpose Operating Systems).

If the TSF implements its own cryptography, this SFR is satisfied through the testing of the individual selection-based requirements defined in Appendix B.1 of the PP.

If the TSF implements its own cryptography, this SFR is satisfied through the testing of the individual selection-based requirements defined in Appendix B.1 of the PP.

FAU_SAR_EXT.1 Searchable Certificate Repository

FAU_SAR_EXT.1.1 The TSF shall [*selection: provide a searchable certificate repository, provide the ability to search audit data stored within the [selection: TOE, Operational Environment]*] that can be used to search for certificates containing specified values of the following certificate fields: [*selection:*

- *subject name,*
- *individual components of subject alternative name,*
- *subject ID,*
- *issuer ID,*
- *algorithm ID,*
- *public key,*
- *key usage,*
- *extended key usage,*
- *serial number,*
- *[assignment: list of other certificate fields]],*

returning all matching certificates and [*selection: [assignment: object identifiers necessary to search the audit record for events involving those certificates], audit events involving those certificates*].

Application Note: *The ability to search on certificate fields is useful for conducting forensic analysis. When a searchable list of certificates issued/requested is maintained separately from other audit records, it is sufficient to be able to search for certificates by these fields and search the remainder of audit records by a unique identifier (e.g., the serial number and issuer ID, or thumbprint) for the certificates returned by the search.*

Assurance Activity

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

In conjunction with the tests for FDP_CER_EXT.1, the evaluator shall generate a sufficient number and variety of certificates to populate the repository or audit record with certificates having at least two values for each of the selected search fields selected. Following the instructions within the operational guidance, the evaluator shall search the repository or audit record for certificates containing specific values for each search field selected, and

confirm that all certificates matching the search criteria are returned, and that all returned certificates match the criteria.

5.1.2 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 [*selection: CRLs (RFC 5280), OCSP (RFC 6960), [assignment: other OCSP standards]], no other certificate status information*] and FCS_COP.1(2).

FCO_NRO_EXT.2.3 The TSF shall require and verify proof of origin for certificate requests it receives [*selection: CMC using mechanisms in accordance with FIA_CMC_EXT.1, EST using mechanisms in accordance with FIA_EST_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 *[selection: proof-of-possession mechanisms in CMC using mechanisms in accordance with FIA_CMC_EXT.1, proof-of-possession mechanisms in EST in accordance with FIA_EST_EXT.1].*

FCO_NRO_EXT.2.5 The TSF shall require and verify proof of origin for revocation requests it receives in accordance with [*selection: CMC using mechanisms in accordance with FIA_CMC_EXT.1, EST in accordance with FIA_EST_EXT.1*].

Application Note: *The TOE is responsible for providing proof of origin for information it issues and verifying proof of origin for information it receives. Based on what is chosen in the selection for FCO_NRO_EXT.2.2, the applicable requirements from Annex B (i.e., FDP_CRL_EXT.1, FDP_OCSP_EXT.1) must be included. Based on what is chosen in the selection for FCO_NRO_EXT.2.3-FCO_NRO_EXT.2.5, the applicable requirements from Annex B (i.e., FIA_CMC_EXT.1, FIA_EST_EXT.1) must be included.*

Assurance Activity

TSS

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

Guidance

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

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

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

1. Test 1: For each supported request, the evaluator shall generate and submit a properly authenticated request to the TOE and verify the responses are signed.
2. Test 2: For each supported request, the evaluator shall generate requests that are unsigned, submit to the TOE, and verify that the TOE rejects the request.
3. Test 3: For each supported request, the evaluator shall generate requests that have an invalid signature based on the RFC, submit to the TOE, and verify that the TOE rejects the request.
4. Test 4: For each supported request, the evaluator shall generate requests that are not signed by authorized entities, submit to the TOE, and verify that the TOE rejects the request.
5. Test 5: For each supported request using password based authentication, the evaluator shall use invalid passwords and verify that the TSF rejects the requests.
6. Test 6: For each proof of possession mode supported, the evaluator shall generate an otherwise valid request but modify the proof of possession value. The evaluator shall submit the modified request and verify that the TSF rejects the request.

Transport test:

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

TOE is offline:

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

Test

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

5.1.3 Cryptographic Support (FCS)

FCS_CDP_EXT Cryptographic Dependencies

FCS_CDP_EXT.1.1 The TSF shall [*selection: implement cryptographic functionality, interface with a software cryptographic module utilizing storage that is protected by the Operational Environment, interface with a hardware cryptographic module in the Operational Environment] in order to perform [selection: all, [assignment: subset of SFRs that the specific component implements*]] cryptographic operations.

Application Note: *If the TOE relies on the Operational Environment to provide cryptographic functionality, the ST author is expected to provide justification that this functionality is sufficient to meet the requirements that are defined by the cryptography SFRs defined in this PP. Examples could include the TOE relying on the underlying operating system which has been validated against the Protection Profile for General Purpose Operating Systems or the TOE relying on a hardware-based module (such as a TPM) that was validated against FIPS 140-2 or equivalent national validation program. Additionally, when a software module is used, it is necessary to identify the method by which protected storage is achieved, such as a TPM, a hardware token, or isolated execution environment.*

The interface to OE components could be explicit (the TSF calls the functionality within an OE component), implicit (the OE component meets the requirement without a specific function call from the TSF), or managed (management of the OE component is required to meet the requirement).

If there is more than one component (including the TSF) that is implementing required cryptography for the TOE, iterate this SFR for each such component, and list the specific SFR implemented by the component. In aggregate, all cryptographic SFR required by the TSF should be listed.

Assurance Activity

TSS

If the TSF implements its own cryptography, this SFR is satisfied through the testing of the individual selection-based requirements defined in Appendix B.1 of the PP.

If the TSF interfaces with a cryptographic module in the Operational Environment to provide its cryptographic functionality, the evaluator shall follow national scheme guidance in order to verify that this third-party component provides an appropriate level of cryptography and has been independently validated through a national validation scheme. The evaluator shall also review the evidence to identify the method by which the TSF interfaces with the identified module (e.g. API calls). When the interface is explicit, the evaluator shall work with the product vendor to demonstrate through review of TSF source code or similar evidence that the interface to the OE mechanism is invoked as described. This is done to verify that the claimed cryptographic functionality is what is actually being used by the TSF.

If more than one component (TSF and one or more OE components) is selected, the evaluator shall review the TSS and verify that all SFR required by the TSF are included.

FCS_STG_EXT.1 Cryptographic Key Storage

FCS_STG_EXT.1.1 Persistent private and secret keys shall be stored within the [*selection: TSF, Operational Environment*] [selection: encrypted within a key hierarchy established in accordance with [*selection: FCS_CKM_EXT.1.1(2), FCS_CKM_EXT.1.1(3)] , in an hardware cryptographic module*].

Application Note: *This requirement ensures that persistent secret keys and private keys are stored securely when not in use. If some secrets/keys are manipulated by the TOE and others are manipulated by the environment, then both of the selections can be specified by the ST author and the ST author must identify in the TSS those keys which are manipulated by the TOE and those by the environment.*

If the TOE is an application, and not a dedicated server, then it should store its private keys in the environment-provided key storage.

The ST author is responsible for selecting the manner in which the keys are stored and where they are stored in the selections above.

Assurance Activity

TSS

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

Guidance

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

Test

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

5.1.4 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". At a minimum, the TSF shall ensure that:

a) The version field shall contain the integer 2.
b) The issuerUniqueID or subjectUniqueID fields are not populated.
a) The serialNumber shall be unique with respect to the issuing Certification Authority.
b) 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.
c) The issuer field is not empty.
d) The signature field and the algorithm in the subjectPublicKeyInfo field shall contain the OID for a signature algorithm specified in FCS_COP.1(2).
e) The following extensions are supported:
    a. subjectKeyIdentifier
    b. authorityKeyIdentifier
    c. basicConstraints
    d. keyUsage
    e. extendedKeyUsage
    f. certificatePolicy
f) A subject field containing a null Name (e.g., a sequence of zero relative distinguished names) is accompanied by a populated critical subjectAltName extension.
g) The subjectKeyIdentifier extension is populated with a value unique for each public key contained in a certificate issued by the TSF.

h) 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 TOE's signing certificate.
i) 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 [*selection: serialNumber, notBefore, notAfter*] fields, where the random values are generated in accordance with FCS_RBG_EXT.1.

Application Note: *The requirement applies only to the issuance of X.509 v3 certificates. An optional requirement in Annex A allows for the issuance of X.509 certificates other than V3.*

Consistency is defined in RFC5280 for FDP_CER_EXT.1.2, item I; specifically, for each extendedKeyUsage purpose specified, there must be a consistent keyUsage purpose set.

RFC updates to RFC 5280 are included in this requirement.

The random input to issued certificates in FDP_CER_EXT.1.3 can be spread across multiple of the selectable fields so that the total number of inserted bits is at least 20. Select all that apply.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes the certificate profile function in accordance with FDP_CER_EXT.1.1 The TSS shall describe how certificate profiles are configured and then selected to issue certificates in accordance with FDP_CER_EXT.1.2. The evaluator shall also ensure that the TSS describes how the TSF ensures that a certificate-requesting subject possesses the applicable private key. Finally, the evaluator shall ensure that the TSS describes how 20 bits of random are generated in accordance with FDP_CER_EXT.1.4 and which certificate fields are involved.

Guidance

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

Test

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

- Test 1: The evaluator shall configure a certificate profile using the available guidance, request a certificate using the profile, and then

examine the certificate contents to ensure it matches the configured certificate profile.

- Test 2: The evaluator shall specifically examine the certificate generated in Test 1 to ensure that it satisfies all field constraints in FDP_CER_EXT.1.2.
- Test 3: For each field and extension defined in FDP_CER_EXT.1.2, the evaluator shall attempt to create a certificate request that violates the required conditions of the field. The evaluator shall determine that all such attempts are rejected by the TSF.
- Test 4: The evaluator shall configure a certificate profile with inconsistent keyUsage and extendedKeyUsage fields and shall submit a certificate request using the profile. The evaluator shall determine that the TSF does not issue the certificate.
- Test 5: The evaluator shall configure a certificate profile and create a certificate request that violates the validity period setting in the configured profile (e.g., notBefore precedes the current time, the combination of notBefore and notAfter is beyond the validity period setting). The evaluator shall submit the certificate request using the profile and verify that the TSF rejects the request.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FDP_CER_EXT.2 Certificate Request Matching

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

Application Note: *This requirement ensures that the TOE provides linkage between submitted requests and issued certificates.*

Assurance Activity

TSS

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

Guidance

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

Test

The evaluator shall perform the following test:

- Test 1: The evaluator shall configure a certificate profile using the available guidance and request a certificate using the profile as a subscriber. The evaluator shall then assume the CA Operations role and access the TOE's interface. The evaluator shall inspect the interface and verify that it provides linkage between submitted certificate requests and issued certificates.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FDP_CER_EXT.3 Certificate Issuance Approval

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

Application Note: *Certificate profiles are defined in accordance with FDP_CER_EXT.1. The various iterations of FMT_MOF.1 define the roles that are allowed to approve the issuance of certificates.*

Assurance Activity

TSS

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

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

Test

The evaluator shall perform the following test:

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

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

- Test 2: The evaluator shall construct one or more certificate requests that meet the rules for automatic approval, and shall verify that each requested certificate was issued.
- Test 3: The evaluator shall attempt to construct one or more certificate requests that violate the rules for automatic approval, and shall verify that the requested certificates are not issued.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FDP_CSI_EXT.1 Certificate Status Information

FDP_CSI_EXT.1.1 The TSF shall provide certificate status information whose format complies with [*selection: ITU-T Recommendation X.509v1 CRL, ITU-T Recommendation X.509v2 CRL, the OCSP standard as defined by [selection: RFC 6960, other OCSP standard*]].

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

Application Note: *Based on the selection, the ST author must choose the appropriate requirements from Annex B.*

The ST should specify the format used to supply certificate status information. If other OCSP standard is selected, only current standards shall be selected, the RFC shall be referenced, and any optional features within the RFC shall be specified.

The various iterations of FMT_MOF.1 defines the role or roles authorized to approve changes to a certificate's status.

Assurance Activity

TSS

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

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

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

If the TOE supports the configuration of certificate status information, the evaluator shall examine the operational guidance to ensure that instructions are available to configure the certificate status function to utilize the formats identified in FDP_CSI_EXT.1.1.

Guidance

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

Test

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

- Test 1: For certificate status information, the evaluator shall configure the TSF to provide certificate status information according to each format identified in FDP_CSI_EXT.1.1 in turn and request certificate status for each format. Each certificate status response shall be examined to ensure that it conforms to the format as described in the TSS.
- Test 2: For each selected certificate status format, the evaluator shall issue a valid certificate from the TOE. The evaluator shall then cause the TOE to issue certificate status information. The evaluator shall check the certificate status information to verify that it reflects that the certificate is valid.
- Test 3: For each selected certificate status format, the evaluator shall revoke a valid certificate from the TOE. The evaluator shall then cause the TOE to issue certificate status information. The evaluator shall check the certificate status information to verify that it reflects that the certificate is revoked.
- Test 4: The evaluator shall configure the certificate status change approval function in accordance with the operational guidance. The

evaluator shall create a certificate status change request and submit it to the TOE. The evaluator shall access the TOE using the defined interface and verify that the submitted request is in the appropriate queue. The evaluator shall approve the certificate status change request. The evaluator shall then cause the TOE to issue certificate status information. The evaluator shall check the certificate status information to verify that it reflects the state of the certificate.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FDP_RIP.1 Subset Residual Information Protection

FDP_RIP.1.1 The TSF and [selection: Operational Environment, no other component] shall ensure that any previous information content of a resource is made unavailable upon the [*selection: allocation of the resource to, deallocation of the resource from*] the following objects: [*assignment: list of objects*].

Application Note: *"Resources" in the context of this requirement are any data buffers used to implement certificate authority functions, including network communications with the Certificate Authority. The concern is that a buffer or memory area might be reused in subsequent function or communication channel resulting in inappropriate disclosure of sensitive data. Note that this requirement applies only to resources that the TSF controls. "Objects" refers to any sensitive data objects that are under control of the TSF, such as subscribers' personally identifiable information.*

The first selection should include 'Operational Environment' if the TSF depends on a component of the OE to store and protect TSF data. The ST should specify the component and any interface used to meet this requirement.

Assurance Activity

TSS

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

Guidance

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

Test

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FDP_STG_EXT.1 Certificate Data Storage

FDP_STG_EXT.1.1(1) The TSF shall [*selection: implement, interface with the Operational Environment to implement*] access controlled storage for the trusted public keys and certificates (trust store elements) used to validate local logon, trusted channel, and external communication to the CA.

Assurance Activity

TSS

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

Guidance

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

Test

The evaluator shall perform the following test:

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

5.1.5 Identification and Authentication (FIA)

FIA_X509_EXT.1 Certificate Validation

FIA_X509_EXT.1.1 The TSF shall [*selection: 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 and that the cA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [*selection: the Online Certificate Status Protocol (OCSP) as specified in FDP_CSI_EXT.1, a Certificate Revocation List (CRL) as specified in FDP_CSI_EXT.1].*
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3),
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field,
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.

Application Note: *The TSF may rely on the Operational Environment to perform certificate handling functionality in cases where the TOE relies on an environmental component to provide trusted remote communications.*

FIA_X509_EXT.1 lists the rules for validating certificates. The ST author shall select whether revocation status is verified using OCSP or CRLs. Depending on this selection, the appropriate CRL or OCSP requirements from Annex B must be included.

Certificates may optionally be used for trusted updates of TSF Software (FPT_TUD_EXT.1) and for data/software integrity verification (FPT_TST_EXT.2) and, if implemented, must be validated to contain the Code Signing purpose extendedKeyUsage.

Whenever TLS or HTTPS is used by the TSF to protect communications originating from external IT entities, certificates used to perform authentication must be validated to contain the Client Authentication purpose extendedKeyUsage.

Whenever the TOE originates messaging to external IT services using TLS or HTTPS, certificates must be used to perform the authentication and must be validated to contain the Server Authentication purpose extendedKeyUsage.

It should be noted that in all cases, the validation is expected to end in a trusted root certificate.

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.

Application Note: *This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added to the Trust Anchor Database.*

Assurance Activity

TSS

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

Guidance

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

Test

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

- Test 1: The evaluator shall demonstrate that validating a certificate without a valid certification path results in the function (application validation, trusted channel setup, or trusted software update) failing. The evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails.
- Test 2: The evaluator shall demonstrate that validating an expired certificate anywhere in a certificate path results in the function failing.
- Test 3: The evaluator shall test that the TOE can properly handle revoked certificates –conditional on whether CRL or OCSP is selected; if both are selected, and then a test is performed for each method. The evaluator has to only test one up in the trust chain (future revisions may require to ensure the validation is done up the entire chain). The evaluator shall ensure that a valid certificate is used, and that the validation function succeeds. The evaluator shall then attempt the test with a certificate that will be revoked (for each method chosen in the selection) and verify that the validation function fails.

- Test 4: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate does not contain the basicConstraints extension. The validation of the certificate path fails.
- Test 5: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension not set. The validation of the certificate path fails.
- Test 6: The evaluator shall construct a certificate path, such that the certificate of the CA issuing the TOE's certificate has the cA flag in the basicConstraints extension set to TRUE. The validation of the certificate path succeeds.
- Test 7: The evaluator shall modify a single byte in the certificate and verify that the certificate fails to validate.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FIA_X509_EXT.2 Certificate-Based Authentication

FIA_X509_EXT.2.1 The TSF shall [*selection: use, interface with the Operational Environment to use*] X.509v3 certificates as defined by RFC 5280 to support authentication for code signing for TOE updates, [*selection: IPsec, TLS, HTTPS, SSH], and [selection: integrity verification for TSF protected data, [assignment: other uses], no additional uses*].

Application Note: *The ST author's selection of trusted communication channel shall match the selection in FTP_ITC.1.1. Certificates may optionally be used for integrity verification (FPT_TST_EXT.2) and other uses..*

FIA_X509_EXT.2.2 When the TSF cannot determine the current revocation status of a certificate, the TSF shall [*selection: allow the administrator to choose whether to accept the certificate, accept the certificate, not accept the certificate*].

Application Note: *The TSF may rely on the Operational Environment to perform certificate handling functionality in cases where the TOE relies on an environmental component to provide trusted remote communications. If the ST author selects SSH, the TSF shall be validated against the Extended Package for Secure Shell.*

Often a connection must be established to perform a verification of the revocation status of a certificate - either to download a CRL or to perform OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA_X509_EXT.1, the behavior

indicated in the second selection shall determine the validity. The TOE must not accept the certificate if it fails any of the other validation rules in FIA_X509_EXT.1. If the administrator-configured option is selected by the ST author, the ST author must also select function 22 in FMT_SMF.1.

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

Application Note: *Trusted communication channels include any of IPsec, TLS, or HTTPS, performed by the TSF. Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes the certificate(s) used by the TOE, the different uses for each certificate, and how the TSF chooses which certificates to use. The evaluator shall examine the TSS to confirm that it describes the behavior of the TOE when a connection cannot be established during the validity check of a certificate used in establishing a trusted channel.

Guidance

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

Test

The evaluator shall perform the following tests:

- Test 1: For each function listed in FIA_X509_EXT.2.1 that requires the use of certificates the evaluator shall demonstrate that using a certificate without a valid certification path results in the function failing. Using the operational guidance, the evaluator shall then load a certificate or certificates needed to validate the certificate to be used in the function, and demonstrate that the function succeeds. The evaluator then shall delete one of the certificates, and show that the function fails.
- Test 2: The evaluator shall demonstrate that using a valid certificate that requires certificate validation checking to be performed in at least some part by communicating with a non-TOE IT entity. The evaluator shall then manipulate the environment so that the TOE is unable to verify the validity of the certificate, and observe that the action selected in FIA_X509_EXT.2.2 is performed. If the selected action is

administrator-configurable, then the evaluator shall follow the operational guidance to determine that all supported administrator-configurable options behave in their documented manner.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FIA_UAU_EXT.1 Authentication Mechanism

FIA_UAU_EXT.1.1 The TSF shall [selection: provide, interface with the OE to provide] a [*selection: password-based authentication mechanism, [assignment: other authentication mechanism(s)*] to perform privileged user authentication.

Application Note: *Examples of "other authentication mechanisms" for the selection include one-time password mechanisms such as RSA SecurID, certificates, and biometrics.*

Assurance Activity

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

FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring a non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- Obtain certificate status information;
- Download certificate from repository;
- [*selection: no other actions, [assignment: list of services or actions performed by the TSF in response to non-TOE entity request*]].

Application Note: *A "non-TOE entity" refers to users (privileged user, subscribers, and relying parties) of services available from the TOE directly. While it should be the case that few or no services are available to external entities prior to identification and authentication, if there are some available to non-TOE entities, these should be listed in the assignment statement; otherwise "no other actions" should be selected.*

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, [*selection: no other actions, [assignment: other TSF-mediated 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.

Application Note: *Authentication can be password-based through the local console or through a protocol that supports passwords (such as SSH), or certificates (such as TLS).*

Certificate renewal and certificate revocation requests can be performed by subscribers with valid certificates and are limited to actions on those certificates; subscribers cannot renew or revoke other users' certificates. Privileged user access requires further authentication. If there are other actions available to authenticated users, these should be listed in the assignment; otherwise, "no other actions" should be selected.

Assurance Activity

TSS

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

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

Guidance

The evaluator shall examine the operational guidance to determine that any necessary preparatory steps (e.g., establishing credential material such as pre-shared keys, tunnels, certificates, etc.) to logging in are described. For each supported login method, the evaluator shall ensure the operational guidance provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the operational guidance provides sufficient instruction on limiting all allowed services. The evaluator shall examine the operational guidance to verify that it describes how to configure the constraints on each type of subscriber self-service request.

Test

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

- Test 1: The evaluator shall use the operational guidance to configure the appropriate credential supported for the access

method. For that credential/access method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access.

- Test 2: The evaluator shall configure the non-authenticated services allowed according to the operational guidance, and then determine the services available to an external remote entity (including subscribers and relying parties). The evaluator shall determine that the list of services available is limited to those specified in the requirement. The evaluator shall also verify that non-authenticated remote entities cannot access the services listed in FIA_UIA_EXT.1.2 that require I&A.
- Test 3: For local access, the evaluator shall exercise the services in accordance with FIA_UIA_EXT.1.1 available to a local privileged user prior to I&A, and make sure this list is consistent with the requirement.
- Test 4: The evaluator shall configure the constraints on subscriber self-service requests. The evaluator shall assume a CA Operations Staff or RA Staff role and issue a certificate to at least one unique subscriber. For each configured service, the evaluator shall request authorized activities using the issued certificates and verify that they can be performed.
- Test 5: The evaluator shall configure the constraints on subscriber self-service requests. The evaluator shall assume a CA Operations Staff or RA Staff role and issue a certificate to at least two unique subscribers. For each configured service, the evaluator shall request authorized activities using one issued certificate for the other subscriber's information and shall verify that the request is denied. The evaluator shall request unauthorized activities using one issued certificate and shall verify that the request is denied.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

5.1.6 Security Management (FMT)

Application Note: *FMT_MOF.1 has been broken up into several iterations to define the specific management functions that are available to each of the roles defined by FMT_SMR.2. The ST author should select those security management functions that belong to the roles supported by the TOE. All TSF management functions need to be specified as being able to be performed by at least one of the defined roles.*

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

FMT_MOF.1.1 The **[*selection: TSF, Operational Environment*]** shall restrict the ability to

1. **manage the TOE locally and remotely;**
2. **configure the audit mechanism;**
3. **configure and manage certificate profiles;**
4. **modify revocation configuration;**
5. **configure subscriber self-service constraints;**
6. **perform updates to the TOE;**
7. **perform on-demand integrity tests;**
8. **import and remove X.509v3 certificates into/from the Trust Anchor Database;**

**[*selection:*

9. ***import private keys;***
10. ***configure certificate revocation list function;***
11. ***configure OCSP function;***
12. ***disable deprecated algorithms;***
13. ***accept certificates whose validity cannot be determined;***
14. ***[assignment: other security management functions]*]**

to [*Administrators*].

Application Note: *It is likely that some combination of the TOE and its Operational Environment are collectively responsible for implementing these management functions. In such cases, the ST author should specify, for each function, the component that enforces it.*

Assurance Activity

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

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

FMT_MOF.1.1(2) The TSF shall restrict the ability to

1. **approve and execute the issuance of certificates;**
2. **configure subscriber self-service request constraints;**

**[*selection:*

3. ***configure automated certificate approval management;***
4. ***approve rulesets that govern the authorizations of AORs to manage particular certificates on behalf of an organization;***
5. ***accept, process and export CMC messages;***
6. ***no other function****]* to [***selection: CA Operations Staff, RA Staff***].

Assurance Activity

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

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

FMT_MOF.1.1(3) The TSF shall restrict the ability to

1. approve certificate revocation;
[*selection:*
2. approve rulesets that govern the authorizations of RAs to manage particular certificates on behalf of an organization;
3. ***no other function***] to [*CA Operations Staff*].

Assurance Activity

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

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

FMT_MOF.1.1(4) The TSF shall restrict the ability to

1. perform archival and recovery;
2. perform destruction of sensitive data when no longer needed;
3. perform private or secret key or critical data export
to [***selection: Administrators, Auditor, CA Operations staff***].

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it identifies the restrictions consistent with this requirement. For every function specified across all iterations, the TSS must specify how the restriction is achieved and how (by role or some other specified mechanism).

Guidance

If the role restriction mechanism is configurable, the evaluator shall examine the operational guidance to determine that the necessary instructions to meet each iteration of the FMT_MOF.1 requirement for the TOE in its evaluated configuration are provided.

Test

The evaluator shall, for each management function, assume each role not assigned to that function, attempt to use the function, and verify that the TSF does not permit it.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

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

FMT_MOF.1.1(5) The TSF shall restrict the ability to

- Delete entries from the audit trail

[*selection:*

- *Search the audit trail*
- *Set or change the retention period parameter for audit records requiring extended retention]*

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 [*Administrators*].

Application Note: *The word "manage" includes but is not limited to create, initialize, view, change default, modify, delete, clear, and append. This requirement is intended to be the "default" requirement for management of TSF data; other iterations of FMT_MTD should place different restrictions or operations available on the specifically-identified TSF data. TSF data includes cryptographic information as well; managing these data would include the association of a cryptographic protocol with an interface, for instance.*

Assurance Activity

TSS

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

Guidance

The evaluator shall examine the operational guidance to determine that each of the TSF-data-manipulating functions implemented in response to the

requirements of this PP is identified, and that configuration information is provided to ensure that only administrators have access to the functions.

Test

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions: [

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

[selection:

13. ***Ability to configure the NPE ruleset;***
14. ***Ability to configure automated process used to approve the revocation of a certificate or information about the revocation of a certificate;***
15. ***Ability to approve rulesets that govern the authorizations of RAs or AORs to manage particular certificates on behalf of an organization;***
16. ***[selection: Ability to modify the CRL configuration, Ability to modify the OCSP configuration];***
17. ***Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;***
18. ***Ability to configure the cryptographic functionality;***
19. ***Ability to import private keys;***
20. ***Ability to export TOE private keys (not for archival);***
21. ***Ability to disable deprecated algorithms;***
22. ***Ability to accept certificates whose revocation status cannot be determined;***

23. Ability to accept, process and export CMC messages;
24. No other capabilities]].

Application Note: *Some TOE functions require the use of the Operational Environment. The ST author simply must make clear in the ST what management functions are performed by the TOE itself or which are performed by the TOE in conjunction with its environment.*

Except as indicated below, the security management functions for FMT_SMF.1 are distributed throughout the PP and are included as part of the requirements in FMT_MOF.1, FPT_TST_EXT.1, and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FMT_SMF.1.

Assurance Activity

TSS

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

Guidance

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

Test

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF **and [*selection: Operational Environment, no other component***] shall maintain the roles: [

- *Administrator,*
- *Auditor,*

- *CA Operations Staff,*
- ***[selection: Operator, RA Staff, Authorized Organizational Representative, no other roles]***]

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

FMT_SMR.2.3 The TSF **and [selection: *Operational Environment, 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.

Application Note: *This document specifies six roles: Administrator, Auditor, CA Operations Staff, Operator, Registration Authority, and Authorized Organizational Representative. However, the TOE is not required to maintain all six roles.*

Assurance Activity

TSS

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

Guidance

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

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

Test

The evaluator shall perform the following tests:

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

- Test 2: The evaluator shall attempt to assume the Auditor role in conjunction with any other role as defined in FMT_SMR.2.1 and shall verify it is not possible.
- Test 3: The evaluator shall attempt to assume the CA Operations Staff role in conjunction with any other role as defined in FMT_SMR.2.1 and shall verify it is not possible.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

5.1.7 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: **[*selection: DRBG failure, signature verification failure, integrity test failure, integrity failure on audit, integrity failure on Trust Anchor database, [assignment: other potential TSF failures], [assignment: other potential Operational Environment failures]*]**.

Application Note: *The intent of this requirement is to prevent the use of failed randomization and other events that can compromise the operation of the CA. This means that the TOE must be able to attain a secure/safe state when any of the identified failures occurs. If the TOE should encounter a failure in the middle of a critical operation, the TOE should not just quit operating, leaving key material and user data unprotected.*

The failure of an Operational Environment component can be just as detrimental to security as a failure of the TSF itself. Therefore, in addition to describing the potential TSF failures and how the TOE preserves a secure state in response, the ST author is also expected to use this SFR to express how the TOE is made aware of any environmental failures and how it responds to these.

Assurance Activity

TSS

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

Guidance

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

Test

The evaluator shall perform the following test:

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_KST_EXT.1 No Plaintext Key Export

FPT_KST_EXT.1.1 The TSF and [*selection: Operational Environment, no other component*] shall prevent the plaintext export of [*assignment: list of all keys used by the TSF*].

Application Note: *Keys include all TOE secret and private keys, as well as any user secret and private keys. The intent of this optional requirement is to prevent the keys from being exported during an archive event authorized by the* TOE user or administrator.

If TSF keys are stored in the OE, the TSF requires support of the OE to meet this requirement. The Operational Environment shall be selected and the specific components used shall be described in the TSS.

Assurance Activity

TSS

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

Guidance

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

Test

The evaluator shall perform the following test:

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_KST_EXT.2 TSF Key Protection

FPT_KST_EXT.2.1 The TSF and [*selection: Operational Environment, no other components*] shall prevent unauthorized use of all TSF private and secret keys.

Application Note: *The intent of this requirement is to protect TSF private and secret keys from both unauthorized users and unprivileged processes. Users should not be able to access the keys through "normal" interfaces.*

Assurance Activity

TSS

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

Guidance

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

Test

The evaluator shall perform the following test:

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_RCV.1 Manual Trusted Recovery

FPT_RCV.1.1 After [*assignment: list of failures/service discontinuities*] the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

Application Note: *This requirement ensures that the TSF can determine that the TOE is started up without protection compromise and can recover without protection compromise after discontinuity of operations. Anticipated failures include actions that result in a system crash, media failures, or discontinuity of operations caused by erroneous*

administrative action or lack of erroneous administrative action. The data that needs to be restored includes the TSF keys needed for signature, the Trust Anchor Database, keys needed for management of certificates, all signed certificates, and any certificate status information.

Assurance Activity

TSS

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

Guidance

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

Test

The evaluator shall perform the following test:

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_SKP_EXT.1 Protection of Keys

FPT_SKP_EXT.1.1 The TSF shall [*selection: 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).

Application Note: *The intent of the requirement is that an administrator is unable to read or view the identified keys through "normal" interfaces. While it is understood that the administrator could directly read memory to view these keys, to do so is not a trivial task and may require substantial work on the part of an administrator. Since the administrator is considered a trusted agent, it is assumed they would not endeavor in such an activity.*

Assurance Activity

TSS

Regardless of whether this requirement is met by the TOE or the Operational Environment, the evaluator shall examine the TSS to determine that it details each persistent private and secret key needed to meet the requirements in the ST. For each of these items, the evaluator shall confirm that the TSS details how any secret or private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

Guidance

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

Test

The evaluator shall perform the following test:

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_SKY_EXT.1 Split Knowledge Procedures

FPT_SKY_EXT.1.1 The TSF shall [selection: *support, interface with the operational environment to support] split knowledge procedures to enforce two-party control for the export of CA signing keys, [selection: no other data, user private keys, [assignment: critical data or keys*]] necessary to resume CA functionality after TSF failure using [*selection: key sharing mechanisms in accordance with FCS_CKM_EXT.1(3), FCS_CKM_EXT.1(4), FCS_CKM_EXT.7, and FPT_SKY_EXT.2; [assignment: other mechanism]*].

Application Note: *The intent of this requirement is to limit access to key shares.*

Assurance Activity

If the TSF implements a key sharing mechanism, this SFR is satisfied through the testing of the referenced SFRs in Appendices B.3 and B.8 of the PP.

If the TSF interfaces with a cryptographic module in the Operational Environment to implement a key sharing mechanism, the evaluator shall follow national scheme guidance in order to verify that the mechanism has been

independently validated through appropriate means. The evaluator shall also review the evidence to identify the method by which the TSF interfaces with the identified module (e.g. API calls). The evaluator shall work with the product vendor to demonstrate through review of TSF source code or similar evidence that the OE interface is invoked as described. The evaluator shall also examine the AGD guidance to ensure it contains instructions for configuring the OE to restrict access to the shares.

If the TSF implements another split knowledge procedure, the evaluator shall examine the TSS to ensure the procedure is adequately described, and assess the procedure to ensure that it is effective in restricting access to the CA signing key and all other selected data and keys. The evaluator shall attempt to devise tests to validate that the TSF implements the described mechanism. The evaluator shall review the AGD to ensure it contains clear instructions to privileged users on how to conduct the procedures.

If the TSF interfaces with the OE to implement other split knowledge procedures, the evaluator shall examine the TSS to ensure the procedure is adequately described, and assess the procedure to ensure that it is effective in restricting access to the CA signing key and all other selected data and keys. The evaluator shall ensure that the TSS describes the dependence on the OE and follow national scheme guidance in order to verify that the dependent mechanism(s) have been independently validated through appropriate means. The evaluator shall also review the evidence to identify the method by which the TSF interfaces with the identified module (e.g. API calls). The evaluator shall work with the product vendor to demonstrate through review of source code or similar evidence that the interface is invoked as described. The evaluator shall also examine the AGD guidance to ensure it contains instructions for configuring the OE to restrict access to the CA signing key and all other selected data and keys.

FPT_STM.1 Reliable Time Stamps

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

Application Note: *The TSF is expected to use time data for accuracy in signing and verification activities. Depending on the functionality provided by the TOE, it may also use time data for accurate generation of audit logs and secure communications that have a time component.*

Assurance Activity

TSS

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

Guidance

The evaluator shall examine the operational guidance to ensure it instructs the administrator how to set the time. If the TOE supports the use of a network time protocol (NTP) server, the operational guidance shall describe how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication.

Test

The evaluator shall perform the following tests:

- Test 1: The evaluator shall use the operational guidance to set the time. The evaluator shall then use an available interface to observe that the time was set correctly.
- Test 2: [conditional] If the TOE supports the use of an NTP server; the evaluator shall use the operational guidance to configure the NTP client on the TOE, and set up a communication path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the operational guidance.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The TSF shall [*selection: implement, 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 [*selection: implement, 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 [*selection: implement, 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 [*selection: implement, 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 [*assignment: action to be taken if the verification fails*].

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

Assurance Activity

TSS

The evaluator shall verify that the TSS section of the ST describes all TSF software update mechanisms for updating the system software. The evaluator shall verify that the description includes a digital signature verification of the software before installation and that installation fails if the verification fails. The evaluator shall verify that all software and firmware involved in updating the TSF is described and, if multiple stages and software are indicated, that the software/firmware responsible for each stage is indicated and that the stage(s) which perform signature verification of the update are identified.

The evaluator shall verify that the TSS describes the method by which the digital signature is verified.

The evaluator shall verify that the TSS describes that the public key used to verify the signature is either hardware-protected or is validated to chain to a public key in the Trust Anchor Database. If hardware-protection is selected, the evaluator shall verify that the method of hardware-protection is described and that the ST author has justified why the public key may not be modified by unauthorized parties.

[conditional] If the ST author indicates that the public key for software update digital signature verification, the evaluator shall verify that the update mechanism includes a certificate validation according to FIA_X509_EXT.1 and a check for the Code Signing purpose in the extendedKeyUsage.

Guidance

The evaluator shall examine the operational user to ensure it contains the required information regarding TOE version verification and TOE updates as specified in AGD_OPE.1.

Test

The evaluator shall perform the following tests:

- Test 1: The evaluator shall perform the version verification activity to determine the current version of the product. The evaluator shall obtain a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Then, the evaluator shall perform a subset of other assurance activity tests to demonstrate that the update functions as expected. After the update, the evaluator shall perform the version verification activity again to verify the version correctly corresponds to that of the update.
- Test 2: The evaluator shall obtain or produce an illegitimate update, and shall attempt to install it on the TOE. The evaluator shall verify that the TOE rejects the update.

- Test 3: The evaluator shall obtain or produce an update with an invalid signature, and shall attempt to install it on the TOE. The evaluator shall verify that the TOE rejects the update and performs any other actions specified in the TSS.
- Test 4: The evaluator shall digitally sign the update with a certificate that does not have the Code Signing purpose and verify that application installation fails. The evaluator shall repeat the test using a valid certificate and a certificate that contains the Code Signing purpose and verify that the application installation succeeds.
- Test 5: The tester shall attempt to install an update without the digital signature and shall verify that installation fails. The tester shall attempt to install an update with altered digital signature, and verify that installation fails.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

5.1.8 TOE Access (FTA)

FTA_SSL.3 TSF-Initiated Termination

FTA_SSL.3.1 The TSF shall **[*selection: implement, interface with the Operational Environment to implement*] the ability to** terminate **a remote** interactive session after a [*assignment: **Administrator-configurable** time interval of session inactivity*].

Assurance Activity

TSS

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

Guidance

The evaluator shall examine the operational guidance to ensure it includes instructions for configuring the inactivity time period for remote interactive sessions.

Test

The evaluator shall perform the following test:

- Test 1: The evaluator shall follow the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a remote interactive session with the TOE.

The evaluator shall then observe that the session is terminated after the configured time period.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FTA_SSL.4 User-Initiated Termination

FTA_SSL.4.1 The TSF shall **[*selection: implement, interface with the Operational Environment to implement*] *the ability to*** allow **privileged** user-initiated termination of the **privileged** user's own interactive session.

Assurance Activity

TSS

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

Guidance

The evaluator shall examine the operational guidance to ensure it describes how to terminate interactive sessions.

Test

The evaluator shall perform the following tests:

- Test 1: The evaluator shall initiate an interactive local session with the TOE. The evaluator shall then follow the operational guidance to terminate the session and observe that the session has been terminated.
- Test 2: The evaluator shall initiate an interactive remote session with the TOE. The evaluator shall then follow the operational guidance to terminate the session and observe that the session has been terminated.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FTA_SSL_EXT.1 TSF-Initiated Session Locking

FTA_SSL_EXT.1.1 The TSF shall **[selection: implement, interface with the Operational Environment to implement]**, for local interactive sessions, [*selection:*

- *lock the session—- disable any activity of the user's data access/display devices other than unlocking the session, and requiring that the privileged user re-authenticate to the TSF prior to unlocking the session;*
- *terminate the session]*

after an Administrator-configured time period of inactivity.

Application Note: *This requirement is met by the component of the TOE or TOE platform that performs remote access.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describe the mechanism used for locking local interactive sessions, including the resulting behavior.

Guidance

The evaluator shall examine the operational guidance to ensure it includes instructions for configuring the inactivity time period for local interactive sessions.

Test

The evaluator shall perform the following test:

- Test 1: The evaluator shall follow the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator shall establish a local interactive session with the TOE. The evaluator shall then observe that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator shall ensure that re-authentication is needed when trying to unlock the session.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1 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.

Application Note: *This requirement is intended to apply to interactive sessions between a human user and a TOE. IT entities establishing connections or programmatic connections (e.g., remote procedure calls over a network) are not required to be covered by this requirement.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure that it details each method of access (local and remote) available to the administrator (e.g., serial port, SSH, HTTPS).

Guidance

The evaluator shall examine the operational guidance to ensure it includes instructions for how to configure notices and consent warning messages.

Test

The evaluator shall perform the following test:

- Test 1: The evaluator shall follow the operational guidance to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

5.1.9 Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1 The TSF shall **use [*selection: HTTPS, IPsec, TLS, SSH*]** to provide a **trusted** communication channel between itself and **authorized external network based IT entities supporting the following capabilities: [*selection: audit server, external cryptographic module, directory services, RA, [assignment: other components]*]** 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 [*assignment: list of services for which the TSF is able to initiate communications*].

Application Note: *The intent of the above requirement is to use a cryptographic protocol to protect external communications with authorized IT entities that the TOE interacts with to perform its functions. While there are no requirements on the party initiating the communication, the ST author lists in the assignment for FTP_ITC.1.3 the*

services for which the TOE can initiate the communication with the authorized IT entity. Note that SSH is not included because this protocol is not used by the TSF to connect to other components. If the ST author selects SSH, the TSF shall be validated against the Extended Package for Secure Shell

The requirement implies that not only are communications protected when they are initially established, but also on resumption after an interruption. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an interruption the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.

Assurance Activity

TSS

The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST.

If an external cryptographic module is selected in FTP_ITC.1.1, the evaluator shall examine the TSS to ensure it describes how the external module is used for cryptographic operations versus how any locally provided cryptographic functionality is used.

Guidance

The evaluator shall examine the operational guidance to ensure it contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be interrupted.

Test

The evaluator shall perform the following tests:

- Test 1: The evaluator shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test 2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE.

- Test 3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
- Test 4: The evaluator shall, for each protocol associated with each authorized IT entity tested during test 1, cause an interruption to the connection. The evaluator shall ensure that when connectivity is restored, communications are appropriately protected.

Further assurance activities are associated with the specific protocols.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FTP_TRP.1 Trusted Path

FTP_TRP.1.1 The TSF shall **use [*selection: choose at least one of: HTTPS, IPsec, SSH, 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 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*].

Application Note: This requirement ensures that remote subscribers and privileged users initiate all communication with the TOE via a trusted path, and that all communications with the TOE by remote subscribers and privileged users is performed over this path. The data passed in this trusted communication channel are encrypted as defined the protocol chosen in the first selection. The ST author chooses the mechanism(s) supported by the TOE and ensures the detailed requirements in Annex B corresponding to their selection are copied to the ST if not already present.

Assurance Activity

TSS

The evaluator shall examine the TSS to determine that the methods of remote TOE communication are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE communication are consistent with those specified in the requirement, and are included in the requirements in the ST.

Guidance

The evaluator shall examine the operational guidance to ensure it contains instructions for establishing the remote sessions for each supported method.

Test

The evaluator shall perform the following tests:

- Test 1: The evaluator shall ensure that communications using each specified (in the operational guidance) remote method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test 2: For each method of remote communication supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote session without invoking the trusted path.
- Test 3: The evaluator shall ensure, for each method of remote communication, the channel data is not sent in plaintext.

Further assurance activities are associated with the specific protocols.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

5.2 Security Assurance Requirements

The Security Objectives for the TOE in Section 4 were constructed to address threats identified in Section 3. The Security Functional Requirements (SFRs) in Section 5.1 are a formal instantiation of the Security Objectives. The PP draws from the CC Security Assurance Requirements (SARs) to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

While this section contains the complete set of SARs from the CC, the Assurance Activities to be performed by an evaluator are detailed both in Section 5.1 as well as in this section.

The general model for evaluating TOEs against STs written to conform to this PP is as follows:

After the ST has been approved for evaluation, the Common Criteria Testing Laboratory (CCTL) will obtain the TOE, supporting IT environment, and the administrative guides for the TOE. The Assurance Activities listed in the ST (which will be refined by the CCTL to be TOE-specific, either within the ST or in a separate document) will then be performed by the CCTL. The results of these activities will be documented and presented (along with the administrative guidance used) for validation.

For each assurance family, "Developer Notes" are provided on the developer action elements to clarify what, if any, additional documentation/activity needs to be provided by the developer. For the content/presentation and evaluator activity elements, additional assurance activities are described as a

whole for the family, rather than for each element. Additionally, the assurance activities described in this section are complementary to those specified in Section 5.1.

The TOE security assurance requirements defined in this section identify the management and evaluative activities required to address the threats identified in Section 3.1 of this PP.

5.2.1 Class ADV: Development

The information about the TOE is contained in the guidance documentation available to the end user as well as the TSS portion of the ST. The TOE developer must concur with the description of the product that is contained in the TSS as it relates to the functional requirements. The Assurance Activities contained in Section 5.1 should provide the ST authors with sufficient information to determine the appropriate content for the TSS section.

ADV_FSP.1 Basic Functional Specification

	Developer action elements:
ADV_FSP.1.1D	The developer shall provide a functional specification.
ADV_FSP.1.2D	The developer shall provide a tracing from the functional specification to the SFRs.
Developer Note:	*As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPE and AGD_PRE documentation, coupled with the information provided in the TSS of the ST. The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.*
	Content and presentation elements:
ADV_FSP.1.1C	The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.2C	The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.3C	The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.
ADV_FSP.1.4C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

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

ADV_ FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

Assurance Activity

There are no specific assurance activities associated with these SARs, except ensuring the information is provided. The functional specification documentation is provided to support the evaluation activities described in Section 5.1, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because there is insufficient interface information, then an adequate functional specification has not been provided.

5.2.2 Class AGD: Guidance Documentation

The guidance documents will be provided with the ST. Guidance must include a description of how the IT personnel verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by the IT personnel. Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes instructions to successfully install the TSF in that environment; and Instructions to manage the security of the TSF as a product and as a component of the larger operational environment. Guidance pertaining to particular security functionality is also provided; requirements on such guidance are contained in the assurance activities specified with each requirement.

AGD_OPE.1 Operational User Guidance

Developer action elements:

AGD_OPE.1.1D The developer shall provide operational user guidance.

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

Content and presentation elements:

AGD_OPE.1.1C The operational user guidance shall describe, for each privileged user role, the user-accessible functions and

privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each privileged user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each privileged user role, the available functions and interfaces, in particular all security parameters under the control of the privileged user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each privileged user role, clearly present each type of security-relevant event relative to the privileged user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences, and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each privileged user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements:

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

Assurance Activity

Some of the contents of the operational guidance will be verified by the assurance activities in Section 5.1 and evaluation of the TOE according to the CEM. The following additional information is also required.

The operational guidance shall at a minimum list the processes that comprise the TOE in its evaluated configuration.

The operational guidance shall contain instructions for configuring the Operational Environment to support the functions of the TOE. These instructions shall include configuration of the cryptographic engine associated

with the evaluated configuration of the TOE as well as configuration of the underlying platform. It shall provide a warning to the administrator that use of other cryptographic engines or platforms was not evaluated nor tested during the CC evaluation of the TOE. The documentation must describe the process for installing updates to the TOE. The evaluator shall verify that this process includes the following steps:

Instructions for obtaining the update. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory).

Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful.

The TOE will likely contain security functionality that does not fall in the scope of evaluation under this PP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.

AGD_PRE.1 Preparative Procedures

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE, including its preparative procedures.

Developer Note: *As with the operational guidance, the developer should look to the assurance activities to determine the required content with respect to preparative procedures.*

Content and presentation elements:

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

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

Evaluator action elements:

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

AGD_PRE.1.2E The evaluator *shall apply* the preparative procedures to confirm that the TOE can be prepared securely for operation.

Assurance Activity

As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.

The evaluator shall check to ensure that the following guidance is provided:

- As indicated in the introductory material, administration of the TOE is performed by one or more administrators that are a subset of the group of all users of the TOE. While it must be the case that the overall system (TOE plus Operational Environment [Operational Environment]) provide this capability, the responsibility for the implementation of the functionality can vary from totally the Operational Environment's responsibility to totally the TOE's responsibility. At a high level, the guidance must contain the appropriate instructions so that the Operational Environment is configured so that it provides the portion of the capability for which it is responsible.
- Many of the cryptographic requirements in the PP can be met by the TOE, the Operational Environment, or a combination of the two. The Operational Environment may provide the necessary functionality via use of an external cryptographic module such as a HSM. The guidance must contain the appropriate instructions so that the TOE or Operational Environment is configured to provide the portion of the capability for which it is responsible.

5.2.3 Class ALC: Life-Cycle Support

At the assurance level provided for TOEs conformant to this PP, life-cycle support is limited to end-user-visible aspects of the life-cycle, rather than an examination of the TOE vendor's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it is a reflection on the information to be made available for evaluation at this assurance level.

ALC_CMC.1 Labeling of the TOE

Developer action elements:

ALC_CMC.1.1D The developer shall provide the TOE and a reference for the TOE.

Content and presentation elements:

ALC_CMC.1.1C The TOE shall be labeled with its unique reference.

Evaluator action elements:

ALC_CMC.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

Assurance Activity

The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

ALC_CMS.1 TOE CM Coverage

Developer action elements:

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.2.1C The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

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

Evaluator action elements:

ALC_CMS.2.1E The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.

Assurance Activity

The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.

5.2.4 Class ASE: Security Target Evaluation

As per activities defined in [CEM].

5.2.5 Class ATE: Tests

Testing is specified for functional aspects of the system as well as aspects that take advantage of design or implementation weaknesses. The former is done through the ATE_IND family, while the latter is

through the AVA_VAN family. At the assurance level specified in this PP, testing is based on advertised functionality and interfaces with dependency on the availability of design information. One of the primary outputs of the evaluation process is the test report as specified in the following requirements.

ATE_IND.1 Independent Testing – Conformance

Developer action elements:

ATE_IND.1.1D The developer shall provide the TOE for testing.

Content and presentation elements:

ATE_IND.1.1C The TOE shall be suitable for testing.

Evaluator action elements:

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

ATE_IND.1.2E The evaluator *shall test* a subset of the TSF to confirm that the TSF operates as specified.

Application Note: *If the ST author selects SSH, the TSF shall be validated against Extended Package for Secure Shell*

Assurance Activity

The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of this PP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered.

The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is necessary.

The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the

configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH).

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a "fail" and "pass" result (and the supporting details), and not just the "pass" result.

5.2.6 Class AVA: Vulnerability Analysis

For the current generation of this Protection Profile, the evaluation lab is expected to survey open sources to discover what vulnerabilities have been discovered in these types of products. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. Until penetration tools are created and uniformly distributed to the evaluation labs, the evaluator will not be expected to test for these vulnerabilities in the TOE. The labs will be expected to comment on the likelihood of these vulnerabilities given the documentation provided by the vendor. This information will be used in the development of penetration testing tools and for the development of future Protection Profiles.

AVA_VAN.1 Vulnerability Survey

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

Assurance Activity

As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part

of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in certification authority products, the communications and enrollment protocols used, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, a test would be suitable at the assurance level of this PP. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.

A. Optional Requirements

As indicated in the introduction to this PP, the baseline requirements (those that must be performed by the TOE or its underlying platform) are contained in the body of this PP. Additionally, there are three other types of requirements specified in Annexes A, B, and C.

The first type (in this Annex) is requirements that can be included in the ST, but do not have to be in order for a TOE to claim conformance to this PP. The second type (in Annex B) is requirements based on selections in the body of the PP: if certain selections are made, then additional requirements in that annex will need to be included. The third type (in Annex C) is components that are not required in order to conform to this PP, but will be included in the baseline requirements in future versions of this PP, so adoption by Certification Authority vendors is encouraged. Note that the ST author is responsible for ensuring that requirements that may be associated with those in Annex A, Annex B, and/or Annex C but are not listed (e.g., FMT-type requirements) are also included in the ST.

A.1 Optional CA functionality

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

FCS_COP.1.1(5) The TSF shall perform [*Password-based Key Derivation Functions*] in accordance with a specified cryptographic algorithm **HMAC-[*selection: SHA-1, SHA-256, SHA-384, SHA-512*]** and output cryptographic key sizes [*selection: 128-bit, 256-bit*] that meet the following: [*NIST SP 800-132*].

Application Note: *This requirement is optional because it is only applicable in cases where the TSF securely stores cryptography data with a software-based key hierarchy that uses a PBKDF to perform key derivation. This is not the only method of key derivation that can be used which is why this SFR is not categorized as selection-based.*

The key cryptographic key sizes in the second selection should be made to correspond to the KEK key sizes selected in FCS_CKM_EXT.1(2). A future requirement will require a PBKDF iteration count of at least 1000.

This password must be conditioned into a string of bits that forms the submask to be used as input into the KEK. Conditioning can be performed using one of the identified hash functions or the process described in NIST SP 800-132; the method used is selected by the ST author. NIST SP 800-132 requires the use of a pseudo-random function (PRF) consisting of HMAC with an approved hash function. The ST author selects the hash function used, also includes the appropriate requirements for HMAC and the hash function.

Assurance Activity

TSS

The evaluator shall check that the TSS describes the method by which the password is first encoded and then fed to the SHA algorithm. The settings for the algorithm (padding, blocking, etc.) shall be described, and the evaluator

shall verify that these are supported by the selections in this component as well as the selections concerning the hash function itself. The evaluator shall verify that the TSS contains a description of how the output of the hash function is used to form the submask that will be input into the function and is the same length as the DEK as specified in FCS_CKM_EXT.1.

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

Guidance

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

Test

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FDP_CER_EXT.4 Non-X.509v3 Certificate Generation

FDP_CER_EXT.4.1 For X.509 certificate formats other than v3, the TSF shall ensure that these certificate formats contain the following general characteristics:

- Version (0 or 1);
- Unique identifier of the issuer;
- keyUsage;
- Unique identifier of the certificate
- Validity period
- Signature field in accordance with FCS_COP.1(2)

Application Note: *This optional requirement can be included if X.509 certificate formats other than the mandated v3 are supported.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes the X.509 certificate generation function and the supported and non- features of the ITU-T Recommendation X.509, in accordance with FDP_CER_EXT.2.1, that can be

used to issue certificates. The evaluator shall ensure that the TSS identifies which of the values identified in FDP_CER_EXT.2.1 can be included in generated certificates.

Guidance

If the TOE supports configurable certificate profiles, the evaluator shall examine the operational guidance to ensure that instructions are available to configure certificate profiles used for the generation of X.509 certificates.

Test

The evaluator shall perform the following test:

- Test 1: For each field defined in FDP_CER_EXT.2.1, the evaluator shall attempt to create a certificate request that violates the required conditions of the field. The evaluator shall determine that all such attempts are rejected by the TSF.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_NPE_EXT.1 NPE Constraints

FPT_NPE_EXT.1.1 The TSF shall enforce an Administrator-configurable rule set that specifies authorizations to submit NPE certificate requests.

Application Note: *The rule sets specify when approval by a CA, RA, and/or AOR is required and limits the authorizations of RAs or specific Authorized Organizational Representatives (AORs), to approve NPE certificates associated to a particular organization.*

FPT_NPE_EXT.1.2 The TSF shall require the CA Operations Staff to register any RA, and shall require a CA Operations Staff or authorized RA to register any AORs, and associate each AOR with an organization or set of devices prior to that AOR making requests on behalf of an assigned organization or devices.

Application Note: *Registration authorities may be restricted in the types of certificates they are authorized to request, or the subjects asserted in those requests, but typically have wide authority to request certificates. AORs, on the other hand, are restricted to NPE certificate types, and are further restricted to request certificates for a small number of devices owned by their affiliated organization. Similar to subscriber self-service requests, an AOR's request authority is provided only for those certificates associated to devices the particular AOR is authorized to manage.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes the AOR constraint mechanism, including the ruleset and its enforcement.

Guidance

The evaluator shall examine the operational guidance to verify that it describes how to configure the ruleset. The evaluator shall ensure that the operational guidance includes instructions on how the RAs and CA Operations staff register the AORs and associate the AORs with particular organizations. The evaluator shall also examine the operational guidance to ensure it also describes how AORs, RAs or CA Operations Staff perform certificate management on behalf of the organization for which they are registered.

Test

The evaluator shall perform the following tests:

- Test 1: The evaluator shall assume the Administrator role and configure the ruleset. The evaluator shall then assume other roles and verify that no other roles can modify the ruleset.
- Test 2: The evaluator shall configure the ruleset that restricts an AOR to a particular organization. The evaluator shall assume a CA Operations Staff or RA role and register an AOR with an organization, authorizing the AOR to perform specific operations on that organization's behalf. The evaluator shall then verify that the AOR can perform each authorized operation on behalf of the organization.
- Test 3: The evaluator shall configure the ruleset that restricts an AOR to a particular organization. The evaluator shall assume a CA Operations Staff or RA role and register an AOR with an organization, authorizing the AOR to perform specific operations on that organization's behalf. The evaluator shall verify that the AOR cannot perform any operations on behalf of organizations for which it is not registered. The evaluator shall also verify that the AOR cannot perform unauthorized operations on behalf of its assigned organization.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

A.2 Auditable Events

For each of the optional requirements claimed by the TOE, the ST author shall include the associated auditable events to the claims made in FAU_GEN.1 and ensure that they are correctly generated as part of testing.

Table 5 - Auditable Events for Optional Requirements

Requirement	Auditable Events	Additional Audit Record Contents	Retention Normal/Extended	Responsible TSF or OE Component
FCS_COP.1(5)	None.	None.	N/A	
FDP_CER_EXT.4	Certificate generation.	Name/identifier of certificate, value of certificate generated.	Extended	
FPT_NPE_EXT.1	All changes to NPE rule sets and NPE associations.	The changes made to the NPE rule sets and associations.	Extended	

B. Selection-Based Requirements

As indicated in the body of this PP, there are several methods by which conformant TOEs can mitigate threats against compromise of the communication channel between administrators, other portions of the (distributed) TOE, or external IT entities. One of the secure communication protocols (IPsec, SSH, TLS, TLS/HTTPS) must be implemented in order to provide protected connectivity for (at a minimum) the audit server and remote administrators. Since there are requirements associated with each of the protocol suites, specification of the protocols in the PP becomes confusing and problematic, since specification of optional requirements is not readily supported by the CC. In order to address this situation as cleanly as possible, the following requirements should be included in the ST depending on the selections for the FTP_ITC.1 and FTP_TRP.1 components.

Additionally, distributed TOEs are allowed to claim conformance to this PP. In these cases, the communications between the disparate parts of the TOE need to be protected, and so the ST author includes FPT_ITT.1 in the main body of the ST.

Note that minor adjustments to the narrative information in the beginning of the ST may be required depending on the selections performed. Additionally, depending on the requirements selected, the appropriate information from Section B.2 *Auditable Events* will need to be added to the auditable events table in the ST.

B.1 Management of Subscriber Data

This PP does not mandate the presence or absence of a Registration Authority (RA) in the TOE's Operational Environment. Regardless of whether or not an RA is present, it is necessary for subscriber data to be protected from unauthorized disclosure. Therefore, if the TOE does not rely on an RA, or if the CA provides a centralized repository for subscriber information, the following SFRs must be claimed in order for the TOE to provide assurance that subscriber data is protected.

FCS_CKM_EXT.1(1) Symmetric Key Generation for DEKs

FCS_CKM_EXT.1.1(1) The TSF shall be able to generate data encryption keys (DEKs) of size [*selection: 128-bit, 256-bit*] from

[*selection:*

- *an RBG that meets this profile (as specified in FCS_RBG_EXT.1),*
- *combined from KEKs in a way that preserves the effective entropy of each factor by [selection:*
  - *using an XOR operation,*
  - *concatenating the keys and using a key derivation function (KDF) in accordance with SP 800-108,*
  - *encrypting one key with another in accordance with FCS_COP.1(1) and using modes [selection: AES-CCM, AES-GCM, AES Key Wrap, AES Key Wrap with Padding]]*].

Application Note: *There are two major types of keys: data encryption keys (DEKs) and key encryption keys (KEKs). DEKs are used to protect data at rest (e.g., subscriber PII) that needs to be encrypted. KEKs are used to protect other keys – DEKs, other KEKs, and other types of keys stored by the user or applications.*

For the third selection, if any option but the RBG option is selected, FCS_CKM_EXT.7 in Annex C must be included.

Assurance Activity

TSS

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

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

Guidance

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

Test

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FDP_SDP_EXT.1 User Sensitive Data Protection

FDP_SDP_EXT.1.1 The TSF shall protect [*selection:*

- *subscriber identity information,*
- *subscriber contact information,*
- *photograph from official ID such as an organization ID badge, passport or driver's license,*
- *background check information,*
- *copies of legal documents,*
- *captured biometrics,*

- *[assignment: other personally identifiable information]*]

stored within the TOE via encryption in accordance with FCS_COP.1(1) using a DEK.

FDP_SDP_EXT.1.2 The TSF shall destroy all protected data when no longer required in accordance with the specified cryptographic data destruction method:

[*selection:*

- *by clearing the DEK encrypting the protected data,*
- *in accordance with the following rules:*
  - *For volatile EEPROM the destruction shall be executed by a single direct overwrite [selection:*
    - *consisting of a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1),*
    - *consisting of zeros],*

    followed by a read-verify.
  - *For volatile flash memory the destruction shall be executed by [selection:*
    - *a single direct overwrite consisting of zeros followed by a read-verify,*
    - *a block erase followed by a read-verify]*].

Application Note: *In the selection in FDP_SDP_EXT.1.1, the ST author should indicate all protected data (e.g., subscriber PII) that is protected by the TOE or Operational Environment and specify how that protection is accomplished. Information included only in issued certificates is not included in this requirement.*

For FDP_SDP_EXT.1.2, destroying data refers to rendering it inaccessible to any authorized or unauthorized user or process.

Assurance Activity

TSS

The evaluator shall examine the TSS to verify that it describes (for each supported platform) how the data destruction functionality is invoked.

The evaluator shall examine the TSS to ensure it describes each user data type as indicated in FDP_SDP_EXT.1.1, including where it is stored, how it is protected, when it is destroyed (for example, immediately after use, on system shutdown, etc.); and the type of destruction procedure that is performed.

Guidance

If the protection and destruction of user data is configurable, the evaluator shall examine the operational guidance to ensure it instructs the administrator how to ensure that user data is protected and destroyed in accordance with this requirement.

Test

The evaluator shall perform the following tests for each platform listed in the ST:

- Test 1: The evaluator shall, for each user data type listed in the TSS, locate where the data is stored and verify that it is encrypted.
- Test 2: The evaluator shall, for each user data type listed in the TSS, initiate the supported data destruction mechanism according to the documented times that it should be initiated for that user data type (e.g., immediately after use, on system shutdown, etc.) and verify that the protected data has been destroyed.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

B.2 Internal Audit Requirements

FAU_ADP_EXT.1 allows the ST author to specify whether the TSF implements audit functionality itself or invokes the Operational Environment to perform audit-related services. If the ST author specifies that the TSF is responsible for implementing auditing functions, the corresponding requirements in this section shall be claimed as part of the TOE. If all of the functions below are not claimed, the ST author shall select both options in FAU_ADP_EXT.1.1 and describe how the omitted functions are performed by external services.

FAU_GEN.1 Audit Data Generation

FAU_GEN.1.1 The TSF shall generate a local audit record of the following auditable events:

a) Start-up of the audit functions;
b) All auditable events for the [*not specified*] level of audit; and [
c) *All administrative actions*;
d) *[Specifically defined auditable events listed in Table 4 through Table 6]*].

Application Note: *The ST author can include other auditable events directly in the table; they are not limited to the list presented.*

Requirements in the PP that are supported in whole or in part by the Operational Environment, i.e., where the selection, "interface with …" is made in an SFR, may either have audit events forwarded to the TSF, or maintained by the component invoked. The requirement shall be replicated for each component invoked, and the list in Table 4 decomposed by the component storing the data. The ST author shall also add the relevant auditable events from the optional and selection-based SFRs based on the auditable events listed in Table 5 and Table 6. While the TSF is required to establish an audit record to indicate at least the invocation of the function (i.e. an administrative action took place), the required event data itself (i.e. the result of performing the action) may be included in the audit record of the

specific component of the Operational Environment. In aggregate, all required items must be accounted for. Each of the following SFR will be replicated, consistent with the replication of FAU_GEN.1.1, to reflect the different audit records.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*information specified in column three of Table 4 through Table 6*].

Application Note: *As with the previous component, the ST author should update Table 4 above with any additional information generated. "Subject identity" in the context of this requirement could either be the administrator's user ID or the affected network interface, for example.*

When the Operational Environment is selected (e.g., because the function being audited is performed in whole or in part by the Operational Environment) the ST should identify whether the required information is within the audit record of the TSF or the audit record of another component of the Operational Environment.

Assurance Activity

TSS

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

Guidance

The evaluator shall examine the operational guidance to ensure that it describes the audit mechanism, lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall ensure that the TSS describes every audit event type mandated by the PP and that the description of the fields contains the information required in FAU_GEN.1.2, and the additional information specified in Tables 4 through 6, depending on the characterization of the SFR associated with the particular event as mandatory, optional, or selection-based.

The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP. The evaluator shall examine the operational guidance and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP. The evaluator shall document the methodology or approach taken while determining which actions in the

operational guidance are security relevant with respect to this PP. The evaluator may perform this activity as part of the activities associated with ensuring the operational guidance satisfies the requirements in accordance with AGD_OPE.

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

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

Test

The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in Table 4, any events in Table 5 and Table 6 that correspond with the optional and selection-based SFRs claimed in the Security Target, startup of the audit functions (or startup of the TOE if audit functionality is not enabled or disabled independently of the TOE), and administrative actions. This should include all instances of an event. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable.

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

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FAU_GEN.2 User Identity Association

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

Assurance Activity

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

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

Assurance Activity

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

FAU_SAR.3 Selectable Audit Review

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

Application Note: *FAU_SAR_EXT.1 defines the ability of the TOE to search a certificate repository and/or audit trail to find certificates based on certain values of individual fields. The TSF will likely identify certificates using some form of unique identifier that is not immediately identifiable to an auditor. The intent of this SFR along with FAU_SAR_EXT.1 is that the auditor has the ability to obtain a certificate's unique identifier by searching for other known fields and then using that unique identifier as an input to searching audit data for all activities involving that certificate.*

Assurance Activity

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

FAU_SEL.1 Selective Audit

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

a) [*selection: object identity, user identity, subject identity, host identity, event type*]
b) [*assignment: list of additional attributes that audit selectivity is based upon*].

Application Note: *This SFR is not intended to be used to give the TSF a way to suppress the generation of audit records. Instead, the intended use is to specify one or both of:*

- *specific audit events for inclusion in an audit trail that provided an extended retention period*
- *specific audit events for inclusion in a specific audit repository (if more than one is available)*

Assurance Activity

TSS

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

Guidance

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

Test

The evaluator shall perform the following tests:

- Test 1: For each attribute listed in the requirement, the evaluator shall devise a test to show that selecting the attribute causes only audit events with that attribute (or those that are always recorded, as identified in the administrative guidance) to be recorded.
- Test 2: [conditional] If the TSF supports specification of more complex audit pre-selection criteria (e.g., multiple attributes, logical expressions using attributes) then the evaluator shall devise tests showing that this capability is correctly implemented. The evaluator shall also, in the test plan, provide a short narrative justifying the set of tests as representative and sufficient to exercise the capability.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FAU_STG.1(1) Protected Audit Trail Storage (Normal)

FAU_STG.1.1(1) The TSF shall protect the **locally** stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2(1) The TSF shall be able to [*prevent*] modifications to the **locally** stored audit records **with normal retention requirements** in the audit trail.

Application Note: *This requirement applies to any audit records that exist within the TOE boundary, including temporarily buffered data that is bound for an audit repository in the Operational Environment. Prevention of unauthorized modification is sufficient if detection is not a possibility.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure that it lists each type of audit log generated by the TOE. For each audit log, the TSS shall describe how it is stored, where it is located, and how it is protected. The evaluator shall verify that the TSS' description of the protection includes prevention of unauthorized deletion. The TSS description shall also include detection and/or prevention of unauthorized modification. If roles other than the Auditor are not provided with an interface for accessing the stored audit records, the TSS shall provide a justification for why the role cannot delete or modify the audit records

Guidance

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

Test

The evaluator shall perform the following tests for each role other than the Auditor role:

- Test 1: The evaluator shall assume a role and attempt to delete the stored audit records, then verify that the attempted deletion failed. Test 2 (Conditional): The evaluator shall assume a role and attempt to modify the stored audit records and verify that the attempted modification was allowed but detected in accordance with the TSS activity.

- Test 2: (Conditional): The evaluator shall assume a role and attempt to modify the stored audit records and verify that the attempted modification was prevented in accordance with the TSS activity.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FAU_STG.1(2) Protected Audit Trail Storage (Archive Data)

FAU_STG.1.1(2) The TSF shall protect the **locally** stored audit records **with extended retention requirements** in the audit trail from deletion [selection: prior to their retention period, prior to transfer to an external audit server] by an auditor.

FAU_STG.1.2(2) The TSF shall be able to [*prevent*] modifications to the **locally** stored audit records **with extended retention requirements** in the audit trail.

Application Note: *This requirement applies to audit data that is expected to persist intact beyond the validity of certificates issued by the CA, even in the event of unexpected TSF failure. Refer to Table 4 through Table 6 for the auditable events marked as requiring extended retention that are relevant to this SFR.*

This requirement applies to any audit records that exist within the TOE boundary, including temporarily buffered data that is bound for an audit repository in the Operational Environment. Audit events requiring extended retention stored in the OE are expected to meet this requirement via integrity and redundancy mechanisms typically provided in archive servers.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure that it lists each type of audit log generated by the TOE. For each audit log, the TSS shall describe how it is stored, where it is located, and how it is protected. The evaluator shall verify that the TSS' description of the protection includes prevention of deletion prior to the retention period or transfer to an external archive. The TSS description shall also include prevention of modification of events after they are written to the audit record. If the TSF requires the actions of an Auditor to meet these requirements, the TSS shall describe the restrictions on Auditor activity. If roles other than the Auditor are provided with an interface for accessing the stored audit records, the TSS shall provide a justification for why the role cannot delete or modify the audit records.

Guidance

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

Test

The evaluator shall perform the following tests for each role other than the Auditor role:

- Test 1: The evaluator shall assume a role and attempt to delete the stored audit records, then verify that the attempted deletion failed.
- Test 2: (Conditional): The evaluator shall assume a role and attempt to modify the stored audit records and verify that the attempted modification was prevented in accordance with the TSS activity.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FAU_STG.4 Prevention of Audit Data Loss

FAU_STG.4.1 The TSF shall [*prevent audited events, except those taken by the **Auditor***] and [*assignment: other actions to be taken in case of audit storage failure*] if the audit trail **cannot be written to**.

Application Note: *This requirement applies to the TOE regardless of whether the audit trail is stored within the TOE boundary (e.g. the audit trail is full) or within the Operational Environment (e.g. the connection to a remote audit repository is broken). In either case, the ST author is expected to describe how the TSF is made aware of any such failures and how it behaves in response.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes the behavior of the TSF and what actions can be performed by the Auditor, if any, when the audit trail is full.

Guidance

The evaluator shall examine the operational guidance to ensure it describes what having a full audit trail means and how an Auditor recognizes that this has occurred. The evaluator shall also examine the operational guidance to ensure it includes remedial steps for correcting the issue.

Test

The evaluator shall perform the following test:

- Test 1: The evaluator shall cause the audit trail to become full, verify that the TSF behaves as documented in the TSS, and verify that the Auditor can perform the documented remedial steps.

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

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 [*selection: locally on the TOE, on an external IT entity using a trusted channel protocol defined in FTP_ITC.1*].

Application Note: *The TOE may rely on a non-TOE audit server for storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment. In the selection, the ST author chooses the means by which this connection is protected. The ST author also ensures that the supporting protocol requirement matching the selection is included in the ST.*

The external IT entity is part of the Operational Environment.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes the local audit storage mechanism. The TSS must also describe the means by which the audit data are transferred to the external IT entity and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism.

Guidance

The evaluator shall examine the operational guidance to ensure it describes the configuration of the local audit storage mechanism, including its location and size.

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

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

Test

The evaluator shall perform the following test:

- Test 1: The evaluator shall establish a connection between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall verify that the connection has been successfully established, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.
- Test 2: The evaluator shall examine the audit data transferred to the external audit server in Test 1 and compare it to the locally stored audit data. The evaluator shall verify that the audit records match. If there and any differences, the evaluator shall examine the operational guidance to verify that it explains any discrepancies between locally stored and transmitted audit data.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FAU_STG_EXT.2 Audit Data Retention

FAU_STG_EXT.2.1 The TSF shall apply the following rules for retention of audit data:

- Audit records required to have extended retention shall be retained at least until an auditor configured extension beyond the validity of all certificates impacted by the event.
- [*assignment: list of rules*].

Application Note: *The TSF may apply different policies for different types of audit data (e.g. one type of record may be stored indefinitely while another type is automatically purged after a set period of time). If the TOE interfaces with an environmental audit store, the ST author is expected to identify any interaction that the TSF may have with the audit store in order to satisfy this policy (e.g. the TOE may automatically issue a command to purge certain types of data from the environmental audit store when some condition is satisfied).*

Assurance Activity

The evaluator shall review the TSS to ensure the rules specified are adequate for the retention of audit records as indicated in Tables 4 through 6.

The evaluator shall assume the role of an auditor and establish an extension period for the retention of certificate-related audit records. The evaluator shall cause the TSF to issue a certificate of short validity period. Prior to the retention period (not-after-date+extension), and prior to transferring the audit record to an external archive, the evaluator shall attempt to delete he audit record of an event marked 'extended', and observe that the audit record was not deleted. Also during this time, the evaluator shall attempt to modify the audit record of an event marked 'extended', and observe that the audit record was not modified.

B.3 Internal Cryptographic Requirements

FCS_CDP_EXT.1 allows the ST author to specify whether the TSF implements cryptographic functionality itself or invokes the Operational Environment to perform cryptographic services. If the ST author specifies that the TSF is responsible for implementing its own cryptography, the requirements in this section shall be claimed as part of the TOE.

FCS_CKM.1(1) Cryptographic Key Generation (for Key Establishment)

FCS_CKM.1.1(1) The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with

[*selection:*

- ***NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;***
- ***NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [selection: P-521, no other curves] (as defined in FIPS PUB 186-4, "Digital Signature Standard")***
- ***NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes***

]

and specified cryptographic key sizes [*assignment*: *equivalent to or greater than a symmetric key strength of 112 bits*].

Application Note: *The ST authors should specify whether the TOE generates these keys or whether the Operational Environment is used.*

This component requires that the TSF or Operational Environment be able to generate the public/private key pairs that are used for key establishment purposes for the various cryptographic protocols (HTTPS, TLS, IPsec, SSH) used by the TOE. If multiple schemes are supported, then the ST author should iterate this

requirement to capture this capability. The scheme used will be chosen by the ST author from the selection.

Since the domain parameters to be used are specified by the requirements of the protocol in this PP, it is not expected that the TOE will generate domain parameters, and therefore there is no additional domain parameter validation needed when the TOE complies with the protocols specified in this PP.

The generated key strength of 2048-bit DSA and RSA keys need to be equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it specifies which protocols use these mechanisms. The evaluator shall also examine the entropy section to ensure the strength of mechanism is covered.

Guidance

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

Test

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

Key Generation:

The evaluator shall verify the implementation of the key generation routines of the supported schemes using the applicable tests below.

Key Generation tests for RSA-Based Key Establishment Schemes:

This test verifies the ability of the TSF to correctly produce values for the key components including the public verification exponent e, the private prime factors p and q, the public modulus n and the calculation of the private signature exponent d.

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

o Random Primes:

- o Provable primes
- o Probable primes

o Primes with Conditions:

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

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

Key Generation for Finite-Field Cryptography (FFC) – Based 56A Schemes

FFC Domain Parameter and Key Generation Tests

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

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

o Cryptographic and Field Primes:

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

and two ways to generate the cryptographic group generator g:

o Cryptographic Group Generator:

- o Generator g constructed through a verifiable process
- o Generator g constructed through an unverifiable process.

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

o Private Key:

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

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

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

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

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

- o g != 0,1
- o q divides p-1
- o g^q mod p = 1
- o g^x mod p = y

for each FFC parameter set and key pair.

Key Generation for Elliptic Curve Cryptography (ECC)- Based 56A Schemes

ECC Key Generation Test

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

ECC Public Key Verification (PKV) Test

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

Key Establishment Schemes

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

SP800-56A Key Establishment Schemes

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

Function Test

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

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

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

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

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

Validity Test

The Validity test verifies the ability of the TOE to recognize another party's valid and invalid key agreement results with or without key confirmation. To conduct this test, the evaluator shall obtain a list of the supporting cryptographic functions included in the SP800-56A key agreement implementation to determine which errors the TOE should

be able to recognize. The evaluator generates a set of 24 (FFC) or 30 (ECC) test vectors consisting of data sets including domain parameter values or NIST approved curves, the evaluator's public keys, the TOE's public/private key pairs, MACTag, and any inputs used in the KDF, such as the other info and TOE id fields.

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

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FCS_CKM.1(2) Cryptographic Key Generation (for Authentication)

FCS_CKM.1.1(2) The TSF shall generate **asymmetric** cryptographic keys **used for authentication** in accordance with a specified cryptographic key generation algorithm

**[*selection:*
- ***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 and for RSA schemes;***
- ***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and [selection: P-521, no other curves];*** **]**

and specified cryptographic key sizes [*assignment: equivalent to or greater than a symmetric key strength of 112 bits*].

Application Note: *The generated key strength of 2048-bit DSA and RSA keys need to be equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it specifies which SFRs use these mechanisms. The evaluator shall also examine the entropy section to ensure the strength of mechanism is covered.

Guidance

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

Test

Tests for key generation are found in the FCS_COP.1(2) assurance activities.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FCS_CKM_EXT.1(3) Extended: Key Generation for Key Encryption Keys (TOE Key Archival)

FCS_CKM_EXT.1.1(3) The TSF shall be able to generate [selection: asymmetric KEKs of [assignment: security strength greater than or equal to 112 bits] security strength in accordance with FCS_CKM.1.1(1), symmetric KEKs of [selection: 128-bit, 256-bit] key size] for the archival of TOE keys from two or more shares according to a key sharing mechanism.

Application Note: *This requirement ensures that persistent secret keys and private keys are stored securely when not in use and are available in the event of failure of the TSF or the OE component. At a minimum, the CA signing key is required to be archived since it is required to issue certificate status information until all issued certificates are expired.*

This SFR should be included when the second selection in FPT_SKY_EXT.1 indicates a key sharing mechanism is used for archive of critical TSF keys.

Both asymmetric and symmetric key methods are acceptable.

Assurance Activity

The evaluator will check the TSS to ensure that it lists each persistent secret and private key needed to meet the requirements in the ST. For each of these items, the evaluator will confirm that the TSS lists for what purpose it is used, and how it is stored.

FCS_CKM_EXT.1(4) Symmetric Key Generation for Key Shares

FCS_CKM_EXT.1.1(4) The TSF shall be able to generate key shares of size [***greater than or equal to the security strength of the KEK defined in FCS_CKM_EXT.1(3)***] for the key sharing mechanism indicated in FCS_CKM_EXT.1(3).

Assurance Activity

The evaluator shall review the TSS to confirm that the key share mechanism described preserves key strength of the encrypted keys.

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

[*selection:*

- *by clearing the KEK encrypting the target key,*
- *for volatile memory, the destruction shall be executed by a [selection:*
  - *single direct overwrite consisting of [selection:*
    - *a pseudo-random pattern using the TSF's RBG,*
    - *zeroes,*
    - *ones,*
    - *a new value of a key,*
    - *[assignment: some value that does not contain any CSP]],*
  - *removal of power to the memory,*
  - *destruction of reference to the key directly followed by a request for garbage collection],*
- *for non-volatile memory that consists of the invocation of an interface provided by the underlying platform that [selection:*
  - *logically addresses the storage location of the key and performs a [selection: single, [assignment: ST author-defined multi-pass]] direct overwrite consisting of [selection:*
    - *a pseudo-random pattern using the TSF's RBG,*
    - *zeroes,*
    - *ones,*
    - *a new value of a key,*
    - *[assignment: some value that does not contain any CSP]],*
  - *instructs the underlying platform to destroy the abstraction that represents the key]*.

FCS_CKM_EXT.4.2 The TSF shall destroy all plaintext keying material cryptographic security parameters when no longer needed.

Application Note: *The interface referenced in the requirement could take different forms, the most likely of which is an application programming interface to an OS kernel. There may be various levels of abstraction visible. For instance, in a given implementation the application may have access to the file system details and may be able to logically address specific memory locations. In another implementation the application may simply have a handle to a resource and can only ask the platform to delete the resource. The level of detail to which the TOE has access will be reflected in the TSS section of the ST.*

Several selections allow assignment of a 'value that does not contain any CSP'. This means that the TOE uses some other specified data not drawn from an RBG meeting FCS_RBG_EXT requirements, and not being any of the particular values listed as other selection options. The point of the phrase 'does not contain any CSP' is to ensure that the overwritten data is carefully selected, and not taken from a general 'pool' that might contain current or residual data that itself requires confidentiality protection.

Key destruction does not apply to the public component of asymmetric key pairs.

Assurance Activity

TSS

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

Guidance

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

Test

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

- Test 1: The evaluator shall utilize appropriate combinations of specialized operational environment and development tools (debuggers, simulators, etc.) for the TOE and instrumented TOE builds to test that keys are cleared correctly, including all intermediate copies

of the key that may have been created internally by the TOE during normal cryptographic processing with that key.

Cryptographic TOE implementations in software shall be loaded and exercised under a debugger to perform such tests. The evaluator shall perform the following test for each key subject to clearing, including intermediate plaintext copies of keys that are subsequently encrypted for storage by the TOE:

1. Load the instrumented TOE build in a debugger.
2. Record the value of the key in the TOE subject to clearing.
3. Cause the TOE to perform a normal cryptographic processing with the key from #1.
4. Cause the TOE to clear the key.
5. Cause the TOE to stop the execution but not exit.
6. Cause the TOE to dump the entire memory footprint of the TOE into a binary file.
7. Search the content of the binary file created in #4 for instances of the known key value from #1.

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

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

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FCS_CKM_EXT.5 Public Key Integrity

FCS_CKM_EXT.5.1 Public keys stored within the TSF shall be protected against undetected modification through the use of [*selection: digital signatures (in accordance with FCS_COP.1(2), keyed hashes (in accordance with FCS_COP.1(4))*].

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

FCS_CKM_EXT.5.3 The TSF shall perform actions in accordance with FPT_FLS.1 when key integrity fails.

Application Note: *A TOE may store public keys received in certificate requests, contained in to-be-issued certificates, for trust anchors, or for certain communications protocols, (e.g., SSH). Trust anchor database entries used to validate certificate (even when stored in the form of certificates), and public keys submitted for inclusion in a certificate issued by the TSF require integrity protection as specified by this component.*

Assurance Activity

TSS

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

Guidance

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

Test

NOTE: It might not be possible to access public keys via the TOE interface. If that is the case, then the evaluator must describe the interface and indicate why the interface does not allow access to the public keys.

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

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FCS_CKM_EXT.6 TOE Key Archival

FCS_CKM_EXT.6.1 TOE secret and private keys required for continuity of operations and [*selection: user private keys, no other keys*] shall be exported from the TOE for the purpose

of archival in encrypted form in accordance with FCS_COP.1(1) using KEKs generated in accordance with FCS_CKM_EXT.1(3).

Application Note: *This requirement is required when 'key sharing mechanisms…' is selected in FPT_SKY_EXT.1., and ensures that the archival of any keys required for continuity of operations (e.g., signature keys used to sign CRLs) from the TOE involves encryption of those keys using KEKs that were derived using key sharing mechanisms as specified in FCS_CKM_EXT.1.(3).*

Assurance Activity

TSS

The evaluator shall examine the TSS to verify that it lists the keys that are archived and that the method of archival is described.

Guidance

If the archival function is configurable, the evaluator shall verify that the operational guidance provides instructions on how to perform this function in a manner that is consistent with its description in the ST.

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

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

[*selection:*

- ***AES-CBC (as defined in NIST SP 800-38A) mode***
- ***AES-CCM (as defined in NIST SP 800-38C) mode,***
- ***AES-GCM (as defined in NIST SP 800-38D) mode,***
- ***AES-XTS (as defined in NIST SP 800-38E) mode,***
- ***AES Key Wrap (KW) (as defined in NIST SP 800-38F) mode***
- ***AES Key Wrap with Padding (KWP) (as defined in NIST SP 800-38F) mode*** **]**

and cryptographic key size **[*selection: 128-bit, 256-bit*]**.

Application Note: *For the second selection of FCS_COP.1.1(1), the ST author should choose the mode or modes in which AES operates. For the third selection, the ST author should choose the key sizes besides 128-bit that are supported by this functionality.*

This SFR is in support of multiple TOE encryption requirements. AES-CBC is used for encryption only, AES-CCM and AES-GCM for encryption and authentication, AES-XTS for encryption only, and AES Key Wrap and AES Key Wrap with Padding for key wrapping. It is necessary for the ST author to ensure that the selected AES modes and key sizes are consistent with the claims made in any of the selection-based cryptographic protocols (e.g. if FCS_IPSEC_EXT.1 is selected, CBC and/or GCM must be selected depending on the selections made in FCS_IPSEC_EXT.1.4).

Assurance Activity

TSS

Requirement met with the Operational Environment: For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the encryption/decryption function(s) claimed in that platform's ST contains the encryption/decryption function(s) in the TOE's ST. The evaluator shall also examine the TSS to verify that it describes (for each supported platform) how the encryption/decryption functionality is invoked for each algorithm, mode and key size selected in TOE's ST (it should be noted that this may be through a mechanism that is not implemented by the TOE; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).

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

Guidance

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

Test

AES-CBC Tests

AES-CBC Known Answer Tests

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

> **KAT-1.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 5 plaintext values for each key size selected and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with an all-zeros key of length equal to the selected key size, for each key size selected..
>
> To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using the ciphertext values as input and AES-CBC decryption.
>
> **KAT-2.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 5 key values for each key size selected and obtain the ciphertext value that results from AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the

keys shall be of length equal to the selected key size, for each key size selected.

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

KAT-3. To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of key values described below for each key size selected and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The keys in each set shall have the same length as the selected key size, for each key size, *N*. Key *I* in each set shall have the leftmost *I* bits be ones and the rightmost *N*-*I* bits be zeros, for *I* in [1,N].

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

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

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

AES-CBC Multi-Block Message Test

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

The evaluator shall also test the decrypt functionality for each mode by decrypting an *i*-block message where 1 < *I* <=10. The evaluator shall choose a key, an IV and a ciphertext message of length *I* blocks and decrypt the message,

using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.

AES-CBC Monte Carlo Tests

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

```
# Input: PT, IV, Key
for I = 1 to 1000:
    if I == 1:
        CT[1] = AES-CBC-Encrypt(Key, IV, PT)
        PT = IV
    else:
        CT[i] = AES-CBC-Encrypt(Key, PT)
        PT = CT[i-1]
```

The ciphertext computed in the 100^{0t}h iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

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

AES-CCM Tests

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

Each selected key length

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Each selected key length

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

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

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

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

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

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

XTS-AES Monte Carlo Test

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

Each selected key length

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

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

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

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

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

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

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

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

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

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

> One plaintext length shall be one octet. One plaintext length shall be 20 octets (160 bits).
>
> One plaintext length shall be the longest supported plaintext length less than or equal to 512 octets (4096 bits).

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

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

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

- ***RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of [assignment: 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 [selection: P-521, no other curves] (as defined in FIPS PUB 186-4, "Digital Signature Standard"),***
- ***Digital Signature Algorithm (DSA) with a key size (modulus) of 2048 bits or greater, that meets FIPS-PUB 186-4, "Digital Signature Standard"***].

Application Note: *The ST should specify whether the TOE performs the algorithms, or whether the TOE in combination with the Operational Environment is used. For each supported TOE platform, evidence is required that the claimed platform is able to meet the requirements on behalf of the TOE.*

The ST author should choose the algorithm implemented to perform digital signatures; if more than one algorithm is available, this requirement (and the corresponding FCS_CKM.1 requirement) should be iterated to specify the functionality. For the algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.

For elliptic curve-based schemes, the key size refers to the log_2 of the order of the base point. As the preferred approach for digital signatures, ECDSA will be required in future publications of this PP.:

Assurance Activity

TSS

Requirement met by the Operational Environment: For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the signature generation and verification functionality claimed in that platform's ST contains the signature generation and verification function(s) in the TOE's ST. The evaluator shall also examine the TSS to verify that it describes (for each supported platform) how the signature generation and verification functionality is invoked for each algorithm, mode and key size selected in TOE's ST (it should be noted that this may be through a mechanism that is not implemented by the TOE; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).

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

Guidance

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

Test

Key Generation:

Key Generation for RSA Signature Schemes

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

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

- Random Primes:
    - Provable primes
    - Probable primes
- Primes with Conditions:
    - Primes p1, p2, q1,q2, p and q shall all be provable primes
    - Primes p1, p2, q1, and q2 shall be provable primes and p and q shall be probable primes
    - Primes p1, p2, q1,q2, p and q shall all be probable primes

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

ECDSA Key Generation Tests

FIPS 186-4 ECDSA Key Generation Test

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

pairs to the public key verification (PKV) function of a known good implementation.

FIPS 186-4 Public Key Verification (PKV) Test

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

ECDSA Algorithm Tests

ECDSA FIPS 186-4 Signature Generation Test

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

ECDSA FIPS 186-4 Signature Verification Test

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

RSA Signature Algorithm Tests

Signature Generation Test

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

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

Signature Verification Test

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

The TOE attempts to verify the signatures and returns success or failure.

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

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

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

Application Note: *In future versions of this document, SHA-1 may be removed as an option. SHA-1 for generating digital signatures was disallowed after December 2013, and SHA-1 for verification of digital signatures is strongly discouraged as there may be risk in accepting these signatures.*

The selection of the hashing algorithm must correspond to the selection of the message digest size; for example, if SHA-1 is chosen, then the only valid message digest size selection would be 160 bits.

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

Assurance Activity

TSS

Requirement met by the Operational Environment: For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the signature generation and verification functionality claimed in that platform's ST contains the hash function(s) in the TOE's ST. The evaluator shall also examine the TSS to verify that it describes (for each supported platform) how the hash functionality is invoked for each algorithm, mode and key size selected in TOE's ST (it should be noted that this may be through a mechanism that is not implemented by the TOE; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).

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

Guidance

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

Test

If this requirement is met by the TOE, the evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP.

Short Messages Test–- Bit-oriented Mode

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

Short Messages Test–- Byte-oriented Mode

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

Selected Long Messages Test–- Bit-oriented Mode

The evaluator shall devise an input set consisting of m messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 99*i, where $1 \leq i \leq m$. The message text shall be pseudorandomly generated. The evaluator shall compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Selected Long Messages Test–- Byte-oriented Mode

The evaluator shall devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. The length of the ith message is 512 + 8*99*i, where $1 \leq i \leq m/8$. The message text shall be pseudorandomly generated. The evaluator shall compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

Pseudorandomly Generated Messages Test

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

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

FCS_COP.1.1(4) The TSF shall perform [*keyed hash message authentication*] in accordance with a specified cryptographic algorithm **HMAC-[*selection: SHA-1, SHA-256, SHA-384, SHA-512*], key size [*assignment: key size (in bits) used in HMAC*], and message digest sizes [*selection: 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"*].

Application Note: *The intent of this requirement is to specify the keyed hash message authentication function used when used for key establishment purposes for the various cryptographic protocols used by the TOE (e.g., trusted channel). The hash selection must support the message digest size selection. The hash selection should be consistent with the overall strength of the algorithm used for FCS_COP.1(1).*

In future versions of this document, SHA-1 may be removed as an option. SHA-1 for generating digital signatures was disallowed after December 2013, and SHA-1 for verification of digital signatures is strongly discouraged as there may be risk in accepting these signatures.

Assurance Activity

TSS

Requirement met by the Operational Environment: For each platform listed in the ST, the evaluator shall examine the ST of the platform to ensure that the keyed hash functionality claimed in that platform's ST contains the signature generation and verification function(s) in the TOE's ST. The evaluator shall also examine the TSS to verify that it describes (for each supported platform) how the keyed hash functionality is invoked for each algorithm, mode and key size selected in TOE's ST (it should be noted that this may be through a mechanism

that is not implemented by the TOE; nonetheless, that mechanism will be identified in the TSS as part of this assurance activity).

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

Guidance

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

Test

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

- Test 1: For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be compared to the result of generating HMAC tags with the same key and IV using a known good implementation.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FCS_RBG_EXT.1 Cryptographic Random Bit Generation

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation (RBG) services in accordance with NIST Special Publication 800-90A using [*selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)*] seeded by an entropy source that accumulated entropy from [*selection, a software-based noise source, a hardware-based noise source*].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [*selection: a software-based noise source, TSF hardware-based noise source*] with a minimum of [*selection: 128 bits, 256 bits*] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and authorization factors that it will generate.

Application Note: *NIST Special Pub 800-90, Appendix C describes the minimum entropy measurement that will probably be required future versions of FIPS-140. If possible this should be used immediately and will be required in future versions of this PP.*

For the second selection in FCS_RBG_EXT.1.1, the ST author should select the standard to which the RBG services comply (either 800-90 or 140-2 Annex C).

SP 800-90 contains four different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used (if 800-90 is selected), and include the specific underlying cryptographic primitives used in the requirement or in the TSS. While any of the identified hash functions (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed.

The ST author also ensures that any underlying functions are included in the baseline requirements for the TOE.

For the selection in FCS_RBG_EXT.1.2, the ST author selects the appropriate number of bits of entropy that corresponds to the greatest security strength of the algorithms included in the ST. Security strength is defined in Tables 2 and 3 of NIST SP 800-57A. For example, if the implementation includes 2048-bit RSA (security strength of 112 bits), AES 128 (security strength 128 bits), and HMAC-512 (security strength 256 bits), then the ST author would select 256 bits.

In the future, this profile will require at least one hardware-based noise source; the ST author may select additional noise source. A hardware noise source is a component that produces data that cannot be explained by a deterministic rule, due to its physical nature. In other words, a hardware based noise source generates sequences of random numbers from a physical process that cannot be predicted. For example, a sampled ring oscillator consists of an odd number of inverter gates chained into a loop, with an electrical pulse traveling from inverter to inverter around the loop. The inverters are not clocked, so the precise time required for a complete circuit around the loop varies slightly as various physical effects modify the small delay time at each inverter on the line to the next inverter. This variance results in an approximate natural frequency that contains drift and jitter over time. The output of the ring oscillator consists of the oscillating binary value sampled at a constant rate from one of the inverters – a rate that is significantly slower than the oscillator's natural frequency.

Any hardware component behaving in similarly variable ways that cannot be explained by a precise and predictable rule can serve as a hardware-based noise source. It is also possible to use multiple independent noise sources to increase entropy production and reduce attack potential (by requiring attackers to exploit multiple random bit streams) as long as at least one of the sources is hardware based. It should be noted that timing of interrupts caused by mechanical I/O devices and system counters are not considered hardware-based noise sources for the purposes of this requirement.

Assurance Activity

TSS

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

Guidance

The evaluator shall examine the AGD guidance to ensure it provides clear instructions on how to configure the Operational Environment. If any part of the deterministic RBG services is configurable, the evaluator shall ensure that the operational guidance provides clear instructions for how to configure them.

Test

Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Annex E, Entropy Documentation and Assessment.

The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms.

Implementations Conforming to NIST Special Publication 800-90A

The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RBG functionality.

If the RBG has prediction resistance enabled, each trial consists of

(1) instantiate DRBG,

(2) generate the first block of random bits,

(3) generate a second block of random bits,

(4) uninstantiate.

The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90A).

If the RBG does not have prediction resistance, each trial consists of

(1) instantiate DRBG,

(2) generate the first block of random bits,

(3) reseed,

(4) generate a second block of random bits,

(5) uninstantiate.

The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

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

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

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

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

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_TST_EXT.2 Integrity Test

FPT_TST_EXT.2.1 A [*selection: error detection code (EDC) of at least 32 bits, keyed hash according to FCS_COP.1(4), digital signature algorithm according to FCS_COP.1(2)*] shall be applied to the Trust Anchor Database element(s) stored by the TSF, TSF keys used to manage certificates, certificate database, [*assignment: other data relevant to TSF security*] software and firmware residing within the TSF.

FPT_TST_EXT.2.2 Integrity shall be verified at [*selection: power-up, initialization, on-demand by a privileged user*]. If verification fails, the TSF shall perform actions in accordance with FPT_FLS.1.

Application Note: *This SFR is applicable when 'integrity verification for TSF protected data' is selected in FIA_X509_EXT.2.1. A 64-bit EDC is preferred.*

Assurance Activity

TSS

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

Guidance

The evaluator shall examine the operational guidance to ensure that it includes instructions to verify the integrity of the stored TSF data and code.

Test

The evaluator shall perform the following tests:

- Test 1: The evaluator shall use the operational guidance instructions to verify the integrity of each protected element specified in the TSS.
- Test 2: The evaluator shall modify a TOE binary to verify the integrity test fails and the action defined in FPT_TST_EXT.2.2 occurs. If this test cannot be performed, the evaluator shall provide a justification.
- Test 3: The evaluator shall modify a portion of TSF data to verify the integrity test fails and the action defined in FPT_TST_EXT.2.2 occurs. If this test cannot be performed, the evaluator shall provide a justification.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

B.4 Password Handling Requirements

The following requirements are only applicable when the TOE provides its own password-based authentication mechanism. Therefore they should be included in the TOE boundary if and only if the following conditions are met:

- The selection "local password-based authentication mechanism" in FIA_UAU_EXT.1.1 has been chosen

FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 The TSF shall [*selection: implement, interface with the Operational Environment to implement*] the ability to detect when [*an administrator configurable positive*

*integer within [assignment: range of acceptable value*s]] unsuccessful authentication attempts occur related to [*remote login by a privileged user*].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been [*met*], the TSF shall **[*selection: choose one of: prevent the remote privileged user from successfully authenticating until [assignment: action] is taken by an Administrator, prevent the privileged user from successfully authenticating until an Administrator defined time period has elapsed*]**.

Application Note: *This requirement does not apply to a privileged user at the local console, since it does not make sense to lock a local privileged user's account in this fashion. This could be addressed by (for example) requiring a separate account for local privileged users or having the authentication mechanism implementation distinguish local and remote login attempts. The "action" taken by an administrator is implementation specific and would be defined in the administrator guidance (for example, lockout reset or password reset). The ST author chooses one of the selections for handling of authentication failures depending on how the TOE has implemented this handler.*

Assurance Activity

TSS

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

If the Operational Environment is responsible for this function, the evaluator shall verify that the TSS describes that function.

Guidance

The evaluator shall examine the operational guidance to ensure that instructions for configuring the number of successive unsuccessful authentication attempts (1.1) and time period (1.2, if implemented) are provided, and that the process of allowing the remote privileged user to once again successfully log on is described for each "action" specified (if that option is chosen). If different actions or mechanisms are implemented depending on the authentication method (e.g., TLS vs. SSH), all must be described.

If the Operational Environment is responsible for this function, the evaluator shall verify that the operational guidance instructs the reader to rely on this capability.

Test

The evaluator shall perform the following tests for each method by which remote privileged users access the TOE, either directly or by authenticating to

the Operational Environment from which the TOE inherits user information (e.g., TLS, SSH):

- Test 1 [conditional on first selection item]: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE. The evaluator shall test that once the limit is reached, attempts with valid credentials are not successful. For each action specified by the requirement, the evaluator shall show that following the operational guidance and performing each action to allow the remote privileged user access are successful.
- Test 2 [conditional on second selection item]: The evaluator shall use the operational guidance to configure the number of successive unsuccessful authentication attempts allowed by the TOE and a time period after which valid logins will be allowed for a remote privileged user. After exceeding the specified number of invalid login attempts and showing that valid login is not possible, the evaluator shall show that waiting for the interval defined by the time period before another access attempt will result in the ability for the remote privileged user to successfully log on using valid credentials.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for privileged passwords:

- Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [*selection: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: other characters]*];
- Minimum password length shall be settable by the Administrator, and support passwords of 14 characters or greater.

Application Note: *The ST author selects the special characters that are supported by TOE; they may optionally list additional special characters supported using the assignment. "Privileged passwords" refers to passwords used by privileged users at the local console or over protocols that support passwords, such as SSH and HTTPS.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes how the minimum password is established and the range of values that can be assigned.

Guidance

The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length.

Test

The evaluator shall perform the following test:

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1 The TSF shall provide only [***obscured feedback and [assignment: list of other feedback]***] to the **privileged** user while the authentication is in progress.

Application Note: *"Obscured feedback" implies the TSF does not produce a visible display of the exact authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user that may provide any indication of the authentication data. The assignment can include unobscured feedback such as "the number of characters typed" or "the authentication mechanism that failed the authentication."*

Assurance Activity

TSS

For each authentication mechanism selected in FIA_UAU_EXT.2.1, the evaluator shall examine the TSS to ensure it describes how obscured feedback is provided to the authenticating user. If no obscured feedback is provided, the TSS must provide justification for why it is not provided.

Guidance

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

Test

The evaluator shall perform the following test:

- Test 1: The evaluator shall locally authenticate to the TOE and verify that at most obscured feedback is provided while entering the authentication information.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_APW_EXT.1 Protection of Privileged User Passwords

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

Application Note: *The intent of the requirement is that raw password authentication data are not stored in the clear, and that no user or administrator is able to read the plaintext password through "normal" interfaces. An all-powerful administrator of course could directly read memory to capture a password but is trusted not to do so.*

In this version of the PP there are no requirements on the method used to store the passwords in non-plaintext form, but cryptographic methods based on the requirements in FCS_COP are preferred. In future versions of this PP, FCS_COP-based cryptographic methods that conform to the Level 2 Credential Storage requirements from NIST SP 800-63 will be required.

Assurance Activity

TSS

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

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

Test

The evaluator shall perform the following test:

- Test 1: The evaluator shall use forensic tools to search storage media to verify that passwords cannot be found in an unobscured (e.g., plaintext) form.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

B.5 Certificate Request Protocol

FCO_NRR_EXT.2 Certificate-Based Proof of Receipt

FCO_NRR_EXT.2.1 The TSF shall provide proof of receipt for [*selection: CMC, EST*] by providing signed responses using mechanisms in accordance with [*selection: FIA_CMC_EXT.1, FIA_EST_EXT.1*].

Application Note: *Based on what is chosen in the selections, the applicable requirements from Annex B (i.e., FIA_CMC_EXT.1, FIA_EST_EXT.1) must be included.*

This SFR is claimed if "CMC full responses" is selected in FIA_CMC_EXT.1.2.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes the mechanisms used for generating proof of origin for certificate request response.

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

Test

The evaluator shall perform the following test for each selection:

- Test 1: For each supported request message, the evaluator shall generate and submit a properly authenticated request to the TOE and verify the response is signed. The evaluator shall verify the signature on the responses and show that they are signed by the TOE that generated the response.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FIA_CMC_EXT.1 Certificate Management over CMS (CMC)

FIA_CMC_EXT.1.1 (General Server support) The TSF shall be able to accept and process CMC full requests and [*selection: simple requests, no other requests*].

FIA_CMC_EXT.1.2 (General Server Support) The TSF shall be able to generate CMC simple responses and [*selection: CMC full responses, no other*] that are consistent with the selected certificate profile and which are in accordance with RFC 5272 as updated by RFC 6402, meeting the compliance requirements for CMS server and certification authorities in accordance with RFC 5474 as updated by RFC 6402.

FIA_CMC_EXT.1.3 (Subordinate CAs as CMC clients/end-entities) The TSF shall be able to generate CMC full requests and [*selection: simple requests, no other requests*] and to accept and process CMC simple and CMC full responses in accordance with RFC 5272 as updated by RFC 6402, meeting the compliance requirements for a client and end-entity in accordance with RFC 5474 as updated by RFC 6402.

FIA_CMC_EXT.1.4 (online CMC transport) The TSF shall require CMC transport over HTTPS for online CMC messages in accordance with RFC 5273 as updated by RFC 6402, where the HTTPS is established in accordance with FCS_HTTPS_EXT.1. For CMC requests containing certificate requests other than initial certificate requests authenticated using shared secrets in AuthenticatedData requests or in the Identity Proof Version 2 Control of SignedData requests, the TSF shall require HTTPS with client authentication, shall ensure the authenticating entity is the same as the entity signing the CMC request and any subject indicated in the requested certificate(s) are the same as the authenticating entity, or the authenticating entity is [selection: an authorized RA for the requested subject, an AOR registered for the requested subject, no other entity].

FIA_CMC_EXT.1.5 (cryptographic support for CMC) The TSF shall require CMC simple and full messages use cryptographic support in accordance with this profile. At a minimum the TSF shall ensure:

- Signature generation and verification for SignedData are performed in accordance with FCS_COP.1(2)
- Encryption for EnvelopedData is performed in accordance with FCS_COP.1(1)
- PasswordRecipientInfo for EnvelopedData or AuthenticatedData is derived in accordance with FCS_COP.1(5)
- hashAlgId in Identity Proof Version 2 control, keyGenAlgorithm in Pop Link Witness Version 2 control, witnessAlgID in Encrypted POP and Decrypted POP controls, hashAlgorithm in Publish Trust Anchors control are in accordance with FCS_COP.1(3)
- macAlgId in Identity Proof Version 2 control, macAlgorithm in POP Link Witness Version 2 Control, and the POPAlgID in Encrypted POP and Decrypted POP controls, are in accordance with FCS_COP.1(4)
- DHPOP mechanisms shall be as specified in RFC 6955 with cryptographic support in accordance with this Protection Profile

FIA_CMC_EXT.1.6 (offline) The TSF shall accept, process and export CMC messages under the control of local privileged user sessions for privileged users with CA Operations Staff, [*selection: RA Staff, no other*] role.

Application Note: *Testing requires the establishment of a CMC client with capabilities to inspect and manipulate requests.*

In subsequent versions of the PP, the TSF will be required to meet the Suite B profile for CMC as described in RFC 6403.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure that it describes how CMC support is provided.

The evaluator shall examine the TSS to determine how initial requests are authenticated when no certificates are available.

Guidance

The evaluator shall examine the operational guidance to ensure it contains instructions on how to configure CMC processing to support the TOE's certificate profiles.

If the TSS indicates that neither AuthenticatedData or Identity Proof Version 2 Control mechanisms using shared secretes are supported, the evaluator shall also examine the operational guidance to ensure that it describes how to authenticate requests for subordinate CA certificates, initial subscriber certificates and, if supported, initial certificates for Registration Authority Officers, when no other certificates are available.

Test

The evaluator shall perform the following tests:

Test Group A. Offline CA Operations:

- Test 1:
  - The evaluator shall establish the TSF in an offline mode as an operational root CA, according to AGD-PRE.
  - The evaluator shall use the CMC client to generate a CMC full request to obtain the CA's certificate. The evaluator shall log into the TSF with CA Operations Staff role to submit the request, and observe that the CA's certificate is returned in the response.
  - [Conditional on CMC support for shared secrets:] While still logged into the TSF, the evaluator shall establish a username and shared secret to be used to authenticate a subordinate CA (for Test 2).
  - The evaluator shall install the CA's certificate into the client's trust store for use in subsequent tests.
- Test 2:

- The evaluator shall establish a second instance of the TSF to be a subordinate to the root CA established in Test 1.
- The evaluator shall log into this TSF in the CA Operations Staff role and load the self-signed certificate obtained in Test 1 into the subordinate CA's trust store.
- The evaluator shall generate certificate request(s):
  - [Conditional on CMC support shared secrets:] The evaluator shall request the subordinate CA TSF to generate three CMC requests for the root CA to sign its certificate. One of the requests shall use the established username and shared secret from Test 1 to authenticate and provide proof of possession using CMC AuthenticatedData or Identity Proof Version 2 Control mechanisms, the second request shall use the same username, but modify the shared secret, and the third will modify the username, but use the established shared secret.
  - [Conditional, if the TSF does not provide CMC support for shared secrets:] The evaluator shall follow operational guidance to authenticate the request to the root CA.
- The evaluator shall observe that the requests are formulated as expected and that the root CA certificate repository and audit trail indicate the successful generation.

- Test 3:
  - The evaluator shall sign into the root CA in the CA Operations Staff role and submit in turn the requests generated in Test 2.
  - The evaluator shall observe that the root CA returns a CMC response containing the subordinated CA's signed certificate for the correctly authenticated request, and that the root CA certificate repository and audit trail indicates successful generation.
  - [Conditional on CMC support for shared secrets:] For each request including modified authentication data, the evaluator shall observe that the root CA either returns a full CMC request indicating errors or does not return a request, and that the CA audit trail indicates the errors.
- Test 4: The evaluator shall sign into the subordinate CA in the CA Operations Staff role, import the simple CMC response and complete the initialization of the subordinate CA.

Test Group B. Online subordinate CA (uses root and subordinate CA established in offline tests):

- Test 1:
  - [Conditional on CMC support for shared secrets:] The evaluator shall log onto the subordinate CA in the CA Operations Staff role and establish a username and shared secret for entities represented by the CMC client established above. A different username and shared secret should be used for at least as many entities are there are request types and POP controls (but at least two).

- For each request type indicated in the selection for FIA_CMC_EXT.1.1 and for each POP control supported, the evaluator shall use the client to establish a CMC request, using a different identifier (subject name) for each request.
- [Conditional on CMC support for shared secrets:] The evaluator shall log onto the subordinate CA in the CA Operations Staff role and establish a username and shared secret for entities represented by the CMC client established above. A different username and shared secret should be used for at least as many entities are there are request types and POP controls (but at least two).
- For each request type indicated in the selection for FIA_CMC_EXT.1.1 and for each POP control supported, the evaluator shall use the client to establish a CMC request, using a different identifier (subject name) for each request.
  - [Conditional on CMC support for shared secrets:] The evaluator shall authenticate the requests using the established username/shared secret combinations.
  - [Conditional on TSF not providing CMC support for shared secrets:] The evaluator shall follow operational guidance to authenticate the requests.
- The evaluator shall copy each request, and create new requests with modified POPs.
- [Conditional on CMC support for shared secrets:] The evaluator shall establish an HTTPS session without client authentication between the CMC client and the Subordinate CA, and submit in turn, each of the modified requests, observing that the Subordinate CA's returns full CMC responses indicating POP errors, or does not return responses and the subordinate CA logs indicate the errors.
- The evaluator shall then submit in turn, each of the unmodified requests under the HTTPS session, provide any required approvals, and observe that the Subordinate CA returns CMC responses containing signed end-entity certificates, each of which properly chain to the root CA and that the subordinate CA repository and audit trail indicate successful issuance.

- Test 2:
  - The evaluator shall select one of the client's certificates and use the CMC client to generate a CMC request for a certificate update, authenticated with the selected certificate.
  - The evaluator shall submit the request under the existing, non-authenticated HTTPS session, and observe that either the subordinate CA response with a full CMC response indicating that the transport is invalid or no response is provided and the subordinate CA audit trail indicates the error.
- Test 3: The evaluator shall establish a new HTTPS session with the subordinate CA using client authentication with the selected client certificate (and associated private key) and resubmit the request

selected in Test 2, observing that the subordinate CA returns a simple CMC response containing a valid certificate for the client. The HTTPS session is retained for Test 4.

- Test 4: The evaluator shall select a second client certificate, with a different subject name from that used to establish the HTTPS session, and shall generate a CMC request to update that certificate. The evaluator shall observe that the subordinate CA returns a full CMC response indicating CMC transport failure or does not respond, and that the subordinate CA audit trail indicates the error.

Test Group C. Support for Certificate Profiles

- Test 1: The evaluator shall configure the subordinate CA to use a certificate profile requiring extensions not used in Test Groups A or B.
- Test 2: The evaluator shall select a valid certificate and use the CMC client to generate a CMC request to update the certificate that is otherwise valid, but not populating the required extension, establish an HTTPS session between the client and the subordinate CA with client authentication using the selected client certificate and associated private key, and submit the CMC request. The evaluator shall observe that the subordinate CA returns a full CMC response rejecting the update indicating a profile error or does not return a response, and that the subordinate CA audit trail indicates the error
- Test 3: The evaluator shall generate another otherwise valid CMC request for the selected certificate, this time populating the extension, but with an invalid value. The evaluator shall submit the request via the proper HTTPS transport and observe that the subordinate CA returns a full CMC response indicating the profile error or does not respond, and that the subordinate CA audit trail indicates the error.
- Test 4: Finally, the evaluator shall generate and submit a valid CMC request including the extension and observe that the subordinate CA returns a simple CMC response with the updated certificate and that the subordinate CA certificate repository and audit trail indicate the successful issuance.

Test Group D. Additional Testing of Controls

- Test 1: For each required control, the evaluator shall generate and submit an otherwise valid CMC request including a certificate update where the control is missing, or submitted with an invalid value, and observe that the subordinate CA returns a full CMC with the error indicated or does not respond, and that the subordinate CA audit trail indicates the error.

Test Group E. Additional Cryptographic Testing

- Test 1: For each item in FIA_CMC_EXT.1.5, the evaluator shall generate and submit an otherwise valid CMC request including a certificate update where the item uses an invalid cryptographic mechanism, and observe that the subordinate CA returns a full CMC indicating the failure or does not respond, and that the subordinate CA audit trail indicates the error.

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FIA_EST_EXT.1 Enrollment over Secure Transport

FIA_EST_EXT.1.1 The TSF shall use the Enrollment over Secure Transport (EST) protocol as specified in RFC 7030 to receive and act upon certificate enrollment requests using the simple enrollment method described in RFC 7030 Section 4.2.

FIA_EST_EXT.1.2 Certificate enrollment requests shall be authenticated using an existing certificate and corresponding private key as specified by RFC 7030 Section 3.3.2 or [*selection: authenticated using a username and password as specified by RFC 7030 Section 3.2.3, no other authentication methods*].

Application Note: *Enrollment over Secure Transport (EST) uses the Certificate Request Message as specified in FIA_CMC_EXT.1. EST also uses HTTPS as specified in FCS_HTTPS_EXT.1 to establish a secure connection with an EST client.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes the implementation of this protocol and specifies the supported ciphersuites. The evaluator shall examine the TSS to ensure that the ciphersuites specified are identical to those listed for this component.

Guidance

The evaluator shall examine the operational guidance to ensure it contains instructions on configuring the TOE so that EST conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).

Test

The evaluator shall perform the following tests:

- Test 1: The evaluator shall use an EST client to request certificate enrollment from the TOE using the simple131 enrolment method described in RFC 7030 Section 4.2, authenticating the request using an existing certificate and corresponding private key as described by RFC 7030 Section 3.3.2. The evaluator shall confirm that the TOE issues a certificate and returns it to the client.
- Test 2: If username and password authentication is selected, the evaluator shall use an EST client to request certificate enrollment from the TOE using the simple131 enrolment method described in RFC 7030 Section 4.2, authenticating the request using a username and password as described by RFC 7030 Section 3.2.3. The

evaluator shall confirm that the TOE issues a certificate and returns it to the client.

- Test 3: The evaluator shall modify the EST client or setup a man-in-the-middle tool between the EST client and TOE to perform the following modifications to the certificate request:
    - Modify at least one byte in the certificationRequestInfo field of the certificate request message and verify that the TOE rejects the request.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

B.6 Certificate Status Information

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.

Assurance Activity

TSS

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

Test

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

The evaluator shall perform the following tests:

- Test 1: If CRL can be issued, the evaluator shall configure the CRL function using available user guidance and request a CRL in order to ensure that the resulting CRL satisfies all field constraints in FDP_CRL_EXT.1.1.
- Test 2: For each field defined in FDP_CRL_EXT.1.1, the evaluator shall attempt to create a CRL that violates the required conditions of the field. The evaluator shall determine that all such attempts are rejected by the TSF.
- Test 3: The evaluator shall make a selection of fields from a configured CRL function and shall attempt to create a CRL that violates the required conditions of the field. The evaluator shall determine that all such attempts are rejected by the TSF.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FDP_OCSP_EXT.1 OCSP Basic Response Validation

FDP_OCSP_EXT.1.1 When the TSF is configured to allow OCSP responses of the basic response type, the TSF shall verify that all mandatory fields in the OCSP basic response contain values in accordance with FDP_CSI_EXT.1. For formats conforming to IETF RFC 6960, at a minimum, the following items shall be validated:

a) The version field shall contain a 0.
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_OCSP_EXT.1.2 For formats other than those specified by IETF RFC 6960, the following elements shall be present:

a) Version
b) Signature algorithm field
c) Time at which status is known to be correct
d) Time at which response was signed

e) Time at which next response will be available

Assurance Activity

TSS

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

Test

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

The evaluator shall perform the following tests:

- Test 1: If OCSP is supported, the evaluator shall configure the OCSP response function and request certificate status via OCSP in order to ensure that the response satisfies all constraints in FDP_OCSP_EXT.1.1.
- Test 2: For each OCSP response format defined in FDP_OCSP_EXT.1.1, for each item in the format, the evaluator shall attempt to create an OCSP response that violates the required conditions. The evaluator shall determine that all such attempts are rejected by the TSF.
- Test 3: The evaluator shall make a selection of items from a configured OCSP response function and shall attempt to create an OCSP response that violates the required conditions. The evaluator shall determine that all such attempts are rejected by the TSF.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

B.7 Trusted Channel Options

FCS_HTTPS_EXT.1 HTTPS Protocol

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

Application Note: *The ST author must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done either by adding elements to this component, or by additional detail in the TSS.*

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

Assurance Activity

The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish protected communications with remote IT entities, focusing on when client authentication is required Testing for this activity is done as part of the TLS testing.

FCS_IPSEC_EXT.1 IPsec Protocol

FCS_IPSEC_EXT.1.1 The TSF shall implement IPsec to protect communication among TSF components and between TSF and OE components as specified in RFC 4301 and discard unauthorized communication.

Application Note: *RFC 4301 calls for an IPsec implementation to protect IP traffic through the use of a Security Policy Database (SPD). The SPD is used to define how IP packets are to be handled: PROTECT the packet (e.g., encrypt the packet), BYPASS the IPsec services (e.g., no encryption), or DISCARD the packet (e.g., drop the packet). The SPD can be implemented in various ways, including router access control lists, firewall rulesets, a "traditional" SPD, etc. Regardless of the implementation details, there is a notion of a "rule" that a packet is "matched" against and a resulting action that takes place.*

While there must be a means to order the rules, a general approach to ordering is not mandated, as long as the SPD can distinguish the IP packets and apply the rules accordingly. There may be multiple SPDs (one for each network interface), but this is not required.

Assurance Activity

TSS

The evaluator shall examine the TSS and determine that it describes what takes place when a packet is processed by the TOE, e.g., the algorithm used to process the packet. The TSS describes how the SPD is implemented and the rules for processing both inbound and outbound packets in terms of the IPsec policy. The TSS describes the rules that are available and the resulting actions available after matching a rule. The TSS describes how those rules and actions form the SPD in terms of the DISCARD (e.g., drop the packet), and PROTECT (e.g., encrypt the packet) actions defined in RFC 4301 (BYPASS should not be included).

As noted in section 4.4.1 of RFC 4301, the processing of entries in the SPD is non-trivial and the evaluator shall determine that the description in the TSS is sufficient to determine which rules will be applied to protect communication between TOE components and authorized external IT entities, and discard all other communications. This description shall cover both the initial packets (that is, no SA is established on the interface or for that particular packet) as well as packets that are part of an established SA.

Guidance

The evaluator shall examine the operational guidance to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for processing a packet. The description includes both cases – a rule that ensures packets between authorized components are protected and a rule that all other packets are dropped. The evaluator shall determine that the description in the operational guidance is consistent with the description in the TSS, and that the level of detail in the operational guidance is sufficient to allow the administrator to set up the SPD in an unambiguous fashion. This includes a discussion of how ordering of rules impacts the processing of an IP packet.

Test

The evaluator uses the operational guidance to configure the TOE to carry out the following tests:

- Test 1: The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet. The selectors used in the construction of the rule shall be different such that the evaluator can generate a packet and send packets to the gateway with the appropriate fields (fields that are used by the rule - e.g., the IP addresses, TCP/UDP ports) in the packet header. The evaluator performs both positive and negative test cases for each type of rule (e.g., a packet that matches the rule and another that does not match the rule). The evaluator observes via the audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packets were dropped encrypted by the IPsec implementation.
- Test 2: The evaluator shall devise several tests that cover a variety of scenarios for packet processing. As with Test 1, the evaluator ensures both positive and negative test cases are constructed. These scenarios shall exercise the range of possibilities for SPD entries and processing modes as outlined in the TSS and operational guidance. Potential areas to cover include rules with overlapping ranges and conflicting entries, inbound and outbound packets, and packets that establish SAs as well as packets that belong to established SAs. The evaluator shall verify, via the audit trail and packet captures, for each scenario that the expected behavior is exhibited, and is consistent with both the TSS and the operational guidance.

FCS_IPSEC_EXT.1.2 The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

Assurance Activity

The assurance activity for this element is performed in conjunction with the activities for FCS_IPSEC_EXT.1.1.

Test

The evaluator uses the operational guidance to configure the TOE to carry out the following tests:

- Test 1: The evaluator shall configure the SPD such that there is a rule for dropping a packet, encrypting a packet, and allowing a packet to flow in plaintext. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches the rule to allow the packet to flow in plaintext and send that packet. The evaluator should observe that the network packet is passed to the proper destination interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator-created entries (there may be a "TOE/platform created" final entry that discards packets that do not match any previous entries). The evaluator sends the packet, and observes that the packet was dropped.

FCS_IPSEC_EXT.1.3 The TSF shall implement transport mode and [*selection: tunnel mode, no other mode*].

Assurance Activity

The evaluator checks the TSS to ensure it states that the VPN can be established to operate in tunnel mode and/or transport mode (as identified in FCS_IPSEC_EXT.1.3).

Guidance

The evaluator shall confirm that the operational guidance contains instructions on how to configure the connection in each mode selected.

Test

The evaluator shall perform the following test(s) based on the selections chosen:

a) Test 1 (conditional): If tunnel mode is selected, the evaluator uses the operational guidance to configure the TOE/platform to operate in tunnel mode and also configures a VPN peer to operate in tunnel mode. The evaluator configures the TOE/platform and the VPN peer to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a connection from the TOE/Platform to the VPN peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.

b) Test 2: The evaluator uses the operational guidance to configure the TOE/platform to operate in transport mode and also configures a VPN peer to operate in transport mode. The evaluator configures the TOE/platform and the VPN peer to use any of the allowed cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator then initiates a

connection from the TOE/platform to connect to the VPN peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) and [*selection: AES-GCM-128 (specified in RFC 4106), AES-GCM-256 (specified in RFC 4106), no other algorithms*] together with a Secure Hash Algorithm (SHA)-based HMAC.

Assurance Activity

The evaluator shall examine the TSS to verify that the algorithms AES-CBC-128 and AES-CBC-256 are implemented. If the ST author has selected either AES-GCM-128 or AES-GCM-256 in the requirement, then the evaluator verifies the TSS describes these as well. In addition, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in FCS_COP.1(4) Cryptographic Operations (for keyed-hash message authentication).

Guidance

The evaluator checks the operational guidance to ensure it provides instructions on how to configure the TOE/platform to use the algorithms, and if either AES-GCM-128 or AES-GCM-256 have been selected the guidance instructs how to use these as well.

Tests

The evaluator shall configure the TOE/platform as indicated in the operational guidance configuring the TOE/platform to use each of the supported algorithms, attempt to establish a connection using ESP, and verify that the attempt succeeds.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [*selection:*

- *IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [selection: no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions];*
- *IKEv2 as defined in RFC 5996 and [selection: with no support for NAT traversal, with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions*].

Application Note: *If the TOE implements SHA-2 hash algorithms for IKEv1 or IKEv2, the ST author shall select RFC 4868. If the ST author selects IKEv1, FCS_IPSEC_EXT.1.15 must also be included in the ST.*

Assurance Activity

TSS

The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented. If IKEv1 is claimed, the evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. It may be that this is a configurable option.

Guidance

The evaluator shall check the operational guidance to ensure it instructs the administrator how to configure the TOE/platform to use IKEv1 and/or IKEv2 (as selected), and uses the guidance to configure the TOE/platform to perform NAT traversal for the following test (if selected). If IKEv1 is claimed and the use of main mode requires configuration of the TOE/platform prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance.

Test

Tests are performed in conjunction with the other IPsec evaluation activities with the exception of the activities below:

- (Conditional): If the TOE claims IKEv1, the evaluator shall configure the TOE/platform as indicated in the operational guidance (if applicable) and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported.
- (Conditional): The evaluator shall configure the TOE/platform so that it will perform NAT traversal processing as described in the TSS and RFC 5996, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [*selection: IKEv1, IKEv2*] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 3602 and [*selection: AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm*].

Application Note: *AES-GCM-128 and AES-GCM-256 may only be selected if IKEv2 is also selected, as there is no RFC defining AES-GCM for IKEv1.*

Assurance Activity

TSS

The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/or IKEv2 payload, and that the algorithms AES-CBC-128, AES-CBC-256 are specified, and if others are chosen in the selection of the requirement, those are included in the TSS discussion.

Guidance

The evaluator ensures that the operational guidance describes the configuration of the mandated algorithms, as well as any additional algorithms selected in the requirement. The guidance is then used to configure the TOE/platform to perform the following test for each ciphersuite selected.

Test

The evaluator shall configure the TOE/platform to use the ciphersuite under test to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using the indicated ciphersuite. The evaluator will confirm the algorithm was that used in the negotiation.

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [*selection*:

- *IKEv1 Phase 1 SA lifetimes can be configured by an Administrator based on [selection:*
  - *number of packets/bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 24] hours];*
- *IKEv2 SA lifetimes can be configured by an Administrator based on [selection:*
  - *number of packets/bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 24] hours]*].

Application Note: *The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either packet/volume-based lifetimes or time-based lifetimes. This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits are not acceptable. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the operational guidance generated for AGD_OPE.*

Assurance Activity

Guidance

The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the operational guidance. If time-based limits are supported, the evaluator ensures that the Administrator is able to configure Phase 1 SA values for 24 hours. Currently there are no values mandated for the number of packets or number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

Test

When testing this functionality, the evaluator needs to ensure that both sides are configured appropriately. From the RFC “A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA

is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

a) Test 1 (Conditional): The evaluator shall configure a maximum lifetime in terms of the number of packets (or bytes) allowed following the operational guidance. The evaluator shall configure a test peer with a packet/byte lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, and determine that once the allowed number of packets (or bytes) through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 1 negotiation.

b) Test 2 (Conditional): The evaluator shall configure a maximum lifetime of 24 hours for the Phase 1 SA following the operational guidance. The evaluator shall configure a test peer with a lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, maintain the Phase 1 SA for 24 hours, and determine that once 24 hours has elapsed, a new Phase 1 SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 1 negotiation.

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [*selection:*

- *IKEv1 Phase 2 SA lifetimes can be configured by an Administrator based on* [*selection:*
  - *number of packets/bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 8] hours*];
- *IKEv2 Child SA lifetimes can be configured by an Administrator based on [selection:*
  - *number of packets/bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 8] hours]*].

Application Note: *The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either packet/volume-based lifetimes or time-based lifetimes. This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits are not acceptable. In general, instructions for setting the parameters of the*

implementation, including lifetime of the SAs, should be included in the operational guidance generated for AGD_OPE.

Assurance Activity

Guidance

The evaluator shall verify that the values for SA lifetimes can be configured and that the instructions for doing so are located in the operational guidance. If time-based limits are supported, the evaluator ensures that the Administrator is able to configure Phase 2 SA values for 8 hours. Currently there are no values mandated for the number of packets or number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

Test

When testing this functionality, the evaluator needs to ensure that both sides are configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

a) Test 1 (Conditional): The evaluator shall configure a maximum lifetime in terms of the number of packets (or bytes) allowed following the operational guidance. The evaluator shall configure a test peer with a packet/byte lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, and determine that once the allowed number of packets (or bytes) through this SA is exceeded, a new SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 2 negotiation.

b) Test 2 (Conditional): The evaluator shall configure a maximum lifetime of 8 hours for the Phase 2 SA following the operational guidance. The evaluator shall configure a test peer with a lifetime that exceeds the lifetime of the TOE. The evaluator shall establish an SA between the TOE and the test peer, maintain the Phase 1 SA for 8 hours, and determine that once 8 hours has elapsed, a new Phase 2 SA is negotiated. The evaluator shall verify that the TOE initiates a Phase 2 negotiation.

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [*assignment: (one or more)*

number(s) of bits that is at least twice the security strength of the negotiated Diffie-Hellman group] bits.

Application Note: *For DH groups 19 and 20, the "x" value is the point multiplier for the generator point G.*

Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.9 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.

Assurance Activity

The evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating "x" (as defined in FCS_IPSEC_EXT.1.). The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of "x" meets the stipulations in the requirement.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonce used in [*selection: IKEv1, IKEv2*] exchanges of length [*selection:*

- *[assignment: security strength associated with the negotiated Diffie-Hellman group];*
- *at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash*].

Application Note: *The ST author must select the second option for nonce lengths if IKEv2 is also selected (as this is mandated in RFC 5996). The ST author may select either option for IKEv1.*

For the first option for nonce lengths, since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1. may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.

Because nonces may be exchanged before the DH group is negotiated, the nonce used should be large enough to support all TOE-chosen proposals in the exchange.

Assurance Activity

Test

- (conditional) If the first selection is chosen, the evaluator shall check to ensure that, for each DH group supported, the TSS describes the process for generating each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.
- (conditional) If the second selection is chosen, the evaluator shall check to ensure that, for each PRF hash supported, the TSS describes the process for generating each nonce. The evaluator shall verify that the TSS indicates that the random number generated that meets the requirements in this PP is used, and that the length of the nonces meet the stipulations in the requirement.

FCS_IPSEC_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [*selection: 19 (256-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), no other DH groups*].

Application Note: *The selection is used to specify additional DH groups supported. This applies to IKEv1 and IKEv2 exchanges. It should be noted that if any additional DH groups are specified, they must comply with the requirements (in terms of the ephemeral keys that are established) listed in FCS_CKM.1.*

Assurance Activity

The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer.

Test

For each supported DH group, the evaluator shall test to ensure that all supported IKE protocols can be successfully completed using that particular DH group.

FCS_IPSEC_EXT.1.12 The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [*selection: IKEv1 Phase 1, IKEv2 IKE_SA*] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [*selection: IKEv1 Phase 2, IKEv2 CHILD_SA*] connection.

Application Note: *The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. Obviously, the IKE version(s) chosen should be consistent not only in this element, but with other choices for other elements in this component. While it is acceptable for this capability to be configurable, the*

default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the AGD documentation) must enable this functionality.

Assurance Activity

The evaluator shall check that the TSS describes the potential strengths (in terms of the number of bits in the symmetric key) of the algorithms that are allowed for the IKE and ESP exchanges. The TSS shall also describe the checks that are done when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites to ensure that the strength (in terms of the number of bits of key in the symmetric algorithm) of the negotiated algorithm is less than or equal to that of the IKE SA this is protecting the negotiation.

Test

The evaluator simply follows the guidance to configure the TOE/platform to perform the following tests.

a) Test 1: This test shall be performed for each version of IKE supported. The evaluator shall successfully negotiate an IPsec connection using each of the supported algorithms and hash functions identified in the requirements.

b) Test 2: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an SA for ESP that selects an encryption algorithm with more strength than that being used for the IKE SA (i.e., symmetric algorithm with a key size larger than that being used for the IKE SA). Such attempts should fail.

c) Test 3: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an IKE SA using an algorithm that is not one of the supported algorithms and hash functions identified in the requirements. Such an attempt should fail.

d) Test 4: This test shall be performed for each version of IKE supported. The evaluator shall attempt to establish an SA for ESP (assumes the proper parameters where used to establish the IKE SA) that selects an encryption algorithm that is not identified in FCS_IPSEC_EXT.1.4. Such an attempt should fail.

FCS_IPSEC_EXT.1.13 The TSF shall ensure that all IKE protocols perform peer authentication using a [*selection: RSA, ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [*selection: Pre-shared Keys, no other method*].

Application Note: *At least one public-key-based Peer Authentication method is required in order to conform to this PP; one or more of the public key schemes is chosen by the ST author to reflect what is implemented. The ST author also ensures that appropriate FCS requirements reflecting the algorithms used (and key generation capabilities, if provided) are listed to support those methods. Note that the TSS will elaborate on the way in which these algorithms are to be used (for example,*

2409 specifies three authentication methods using public keys; each one supported will be described in the TSS).

Assurance Activity

TSS

The evaluator ensures that the TSS identifies RSA and/or ECDSA as being used to perform peer authentication. The description shall be consistent with the algorithms as specified in FCS_COP.1(2) Cryptographic Operations (for cryptographic signature).

If pre-shared keys are chosen in the selection, the evaluator shall check to ensure that the TSS describes how pre-shared keys are established and used in authentication of IPsec connections. The evaluator shall check that the operational guidance describes how pre-shared keys are to be generated and established. The description in the TSS and the operational guidance shall also indicate how pre-shared key establishment is accomplished for TOEs that can generate a pre-shared key as well as TOEs that simply use a pre-shared key.

Guidance

The evaluator ensures the operational guidance describes how to set up the TOE to use certificates with RSA and/or ECDSA signatures and public keys.

In order to construct the environment and configure the TOE for the following tests, the evaluator will ensure that the operational guidance describes how to configure the TOE to connect to a trusted CA, and ensure a valid certificate for that CA is loaded into the TOE and marked “trusted”.

Test

For efficiency sake, the testing that is performed may be combined with the testing for FIA_X509_EXT.1, FIA_X509_EXT.2 (for IPsec connections), and FCS_IPSEC_EXT.1.1. The following tests shall be repeated for each peer authentication selected in the FCS_IPSEC_EXT.1.1 selection above:

a) Test 1: The evaluator shall configure the TOE to use a private key and associated certificate signed by a trusted CA and shall establish an IPsec connection with the peer.

b) Test 2 [conditional]: The evaluator shall generate a pre-shared key off-TOE and use it, as indicated in the operational guidance, to establish an IPsec connection with the peer.

FCS_IPSEC_EXT.1.14 The TSF shall support peer identifiers of the following types: [*selection: IP address, Fully Qualified Domain Name (FQDN), user FQDN, Distinguished Name (DN)] and [selection: no other reference identifier type, [assignment: other supported reference identifier types]*].

Application Note: *The TOE must support at least one of the following identifier types: IP address, Fully Qualified Domain Name (FQDN), user FQDN, or Distinguished Name (DN). In the future, the TOE will be required to support all of these identifier types. The TOE is expected to support as many IP address formats (IPv4 and IPv6) as IP versions*

supported by the TOE in general. The ST author may assign additional supported identifier types in the second selection.

Assurance Activity

The assurance activities for this element are performed in conjunction with the assurance activities for the next element.

FCS_IPSEC_EXT.1.15 The TSF shall not establish an SA if the presented identifier does not match the configured reference identifier of the peer.

Application Note: *At this time, only the comparison between the presented identifier in the peer's certificate and the peer's reference identifier is mandated by the testing below. However, in the future, this requirement will address two aspects of the peer certificate validation: 1) comparison of the peer's ID payload to the peer's certificate which are both presented identifiers, as required by RFC 4945 and 2) verification that the peer identified by the ID payload and the certificate is the peer expected by the TOE (per the reference identifier). At that time, the TOE will be required to demonstrate both aspects (i.e. that the TOE enforces that the peer's ID payload matches the peer's certificate which both match configured peer reference identifiers).*

Excluding the DN identifier type (which is necessarily the Subject DN in the peer certificate), the TOE may support the identifier in either the Common Name or Subject Alternative Name (SAN) or both. If both are supported, the preferred logic is to compare the reference identifier to a presented SAN, and only if the peer's certificate does not contain a SAN, to fall back to a comparison against the Common Name. In the future, the TOE will be required to compare the reference identifier to the presented identifier in the SAN only, ignoring the Common Name.

The configuration of the peer reference identifier is addressed by FMT_SMF.1.1.

Assurance Activity

TSS

The evaluator shall ensure that the TSS describes how the TOE compares the peer's presented identifier to the reference identifier. This description shall include whether the certificate presented identifier is compared to the ID payload presented identifier, which field(s) of the certificate are used as the presented identifier (DN, Common Name, or SAN), and, if multiple fields are supported, the logical order comparison. If the ST author assigned an additional identifier type, the TSS description shall also include a description of that type and the method by which that type is compared to the peer's presented certificate.

Guidance

The evaluator shall ensure that the operational guidance includes the configuration of the reference identifier(s) for the peer.

Test

For each supported identifier type (excluding DNs), the evaluator shall repeat the following tests:

Test 1: For each field of the certificate supported for comparison, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the field in the peer's presented certificate and shall verify that the IKE authentication succeeds.

Test 2: For each field of the certificate support for comparison, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to not match the field in the peer's presented certificate and shall verify that the IKE authentication fails.

The following tests are conditional:

Test 3: (conditional) If, according to the TSS, the TOE supports both Common Name and SAN certificate fields and uses the preferred logic outlined in the Application Note, the tests above with the Common Name field shall be performed using peer certificates with no SAN extension. Additionally, the evaluator shall configure the peer's reference identifier on the TOE to not match the SAN in the peer's presented certificate but to match the Common Name in the peer's presented certificate, and verify that the IKE authentication fails.

Test 4: (conditional) If the TOE supports DN identifier types, the evaluator shall configure the peer's reference identifier on the TOE (per the administrative guidance) to match the subject DN in the peer's presented certificate and shall verify that the IKE authentication succeeds. To demonstrate a bit-wise comparison of the DN, the evaluator shall change a single bit in the DN (preferably, in an Object Identifier (OID) in the DN) and verify that the IKE authentication fails.

Test 5: (conditional) If the TOE supports both IPv4 and IPv6 and supports IP address identifier types, the evaluator must repeat test 1 and 2 with both IPv4 address identifiers and IPv6 identifiers. Additionally, the evaluator shall verify that the TOE verifies that the IP header matches the identifiers by setting the presented identifiers and the reference identifier with the same IP address that differs from the actual IP address of the peer in the IP headers and verifying that the IKE authentication fails.

Test 6: (conditional) If, according to the TSS, the TOE performs comparisons between the peer's ID payload and the peer's certificate, the evaluator shall repeat the following test for each combination of supported identifier types and supported certificate fields (as above). The evaluator shall configure the peer to present a different ID payload than the field in the peer's presented certificate and verify that the TOE fails to authenticate the IKE peer.

FCS_TLSC_EXT.1 TLS Client Protocol

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

- Mandatory Ciphersuites:
  - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- Optional Ciphersuites: [*selection:*
  - *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
  - *TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
  - *TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
  - *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
  - *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
  - *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_256_CBC_ SHA256 as defined in RFC 5246*
  - *TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246*
  - *TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246*
  - *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
  - *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
  - *TLS_ECDHE_ECDSA_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_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
  - *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
  - *no other ciphersuite*].

Application Note: *The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. The Suite B algorithms listed above (RFC 6460) are the preferred algorithms for implementation. TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246.*

These requirements will be revisited as new TLS versions are standardized by the IETF.

If any ciphersuites are selected using ECDHE, then FCS_TLSC_EXT.1.5 is required.

In a future version of this PP TLS v1.2 will be required for all TOEs.

Assurance Activity

TSS

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

Test

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

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

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

Test 4: The evaluator shall configure the server to select the TLS_NULL_WITH_NULL_NULL ciphersuite and verify that the client denies the connection.

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

a) Change the TLS version selected by the server in the Server Hello to a non-supported TLS version (for example 1.3 represented by the two bytes 03 04) and verify that the client rejects the connection.

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

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

d) Modify the signature block in the Server's Key Exchange handshake message, and verify that the client rejects the connection after receiving the Server Key Exchange message.

e) Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.

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

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

Application Note: *The rules for verification of identity are described in Section 6 of RFC 6125. The reference identifier is established by an authorized user, by configuration (e.g., configuring the name of an authentication server), or by an application (e.g., a parameter of an API) as described in the TSS. . Based on a singular reference identifier's source domain and application service type (e.g., HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.*

The preferred method for verification is the Subject Alternative Name using DNS names, URI names, or Service Names. Verification using the Common Name is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name or Subject Alternative name is discouraged as against best practices but may be implemented. Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the assurance activity.

Assurance Activity

TSS

The evaluator shall ensure that the TSS describes the client's method of establishing all reference identifiers from the administrator/application-configured reference identifier, including which types of reference identifiers are supported (e.g., Common Name, DNS Name, URI Name, Service Name, or other application-specific Subject Alternative Names) and whether IP addresses and wildcards are supported. The evaluator shall ensure that this description identifies whether and the manner in which certificate pinning is supported or used by the TOE.

Test

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

Test 1: The evaluator shall present a server certificate that does not contain an identifier in either the Subject Alternative Name (SAN) or Common Name (CN) that matches the reference identifier. The evaluator shall verify that the connection fails.

Test 2: The evaluator shall present a server certificate that contains a CN that matches the reference identifier, contains the SAN extension, but does not contain an identifier in the SAN that matches the reference identifier. The evaluator shall verify that the connection fails. The evaluator shall repeat this test for each supported SAN type.

Test 3: The evaluator shall present a server certificate that contains a CN that matches the reference identifier and does not contain the SAN extension. The evaluator shall verify that the connection succeeds.

Test 4: The evaluator shall present a server certificate that contains a CN that does not match the reference identifier but does contain an identifier in the SAN that matches. The evaluator shall verify that the connection succeeds.

Test 5: The evaluator shall perform the following wildcard tests with each supported type of reference identifier:

- The evaluator shall present a server certificate containing a wildcard that is not in the left-most label of the presented identifier (e.g., foo.*.example.com) and verify that the connection fails.
- The evaluator shall present a server certificate containing a wildcard in the left-most label (e.g., *.example.com). The evaluator shall configure the reference identifier with a single left-most label (e.g., foo.example.com) and verify that the connection succeeds. The evaluator shall configure the reference identifier without a left-most label as in the certificate (e.g., example.com) and verify that the connection fails. The evaluator shall configure the reference identifier with two left-most labels (e.g., bar.foo.example.come) and verify that the connection fails.

Test 6: [conditional] If URI or Service name reference identifiers are supported, the evaluator shall configure the DNS name and the service identifier. The evaluator shall present a server certificate containing the correct DNS name and service identifier in the URIName or SRVName fields of the SAN and verify that the connection succeeds. The evaluator shall repeat this test with the wrong service identifier (but correct DNS name) and verify that the connection fails.

Test 7: [conditional] If pinned certificates are supported the evaluator shall present a certificate that does not match the pinned certificate and verify that the connection fails.

FCS_TLSC_EXT.1.3 The TSF shall establish a trusted channel only if the peer certificate is valid.

Application Note: *Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity shall be tested in accordance with testing performed for FIA_X509_EXT.1.*

Assurance Activity

Test

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

FCS_TLSC_EXT.1.4 The TSF shall support mutual authentication using X.509v3 certificates.

Application Note: *If TLS is used for FTP_ITC.1, then this component is required.*

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.

Assurance Activity

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

Test

Test 1: The evaluator shall perform the following modification to the traffic:

a) Configure the server to require mutual authentication and then modify a byte in a CA field in the Server's Certificate Request handshake message. The modified CA field shall not be the CA used to sign the client's certificate. The evaluator shall verify the connection is unsuccessful.

FCS_TLSC_EXT.1.5 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [*selection: secp256r1, secp384r1, secp521r1*] and no other curves.

Application Note: *If ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, this component is required.*

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1(2) and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

Assurance Activity

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

Test

Test 1: The evaluator shall configure the server to perform an ECDHE key exchange in the TLS connection using a non-supported curve (for example P-192) and shall verify that the TOE disconnects after receiving the server's Key Exchange handshake message.

FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1 The TSF shall implement [*selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346), TLS 1.0 (RFC 2246)*] supporting the following ciphersuites:

- Mandatory Ciphersuites:
  - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- Optional Ciphersuites: [*selection:*
  - *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
  - *TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
  - *TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
  - *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
  - *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
  - *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_256_CBC_ SHA256 as defined in RFC 5246*
  - *TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246*
  - *TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246*
  - *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
  - *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
  - *TLS_ECDHE_ECDSA_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_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
  - *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
  - *no other ciphersuite].*

Application Note: *The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. The Suite B algorithms listed above (RFC 6460) are the preferred*

algorithms for implementation. TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246.

These requirements will be revisited as new TLS versions are standardized by the IETF.

If any ciphersuites are selected using ECDHE, then FCS_TLSS_EXT.1.5 is required.

In a future version of this PP TLS v1.2 will be required for all TOEs.

Assurance Activity

TSS

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

Guidance

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

Test

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

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

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

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

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

b) Modify the signature block in the Client's Key Exchange handshake message, and verify that the server rejects the client's Certificate Verify handshake message (if using mutual authentication) or that the server denies the client's Finished handshake message.

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

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

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

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and [*selection: TLS 1.0, TLS 1.1, no other TLS versions*].

Application Note: *All SSL versions shall be denied. Any TLS versions not selected in FCS_TLSS_EXT.1.1 should be selected here.*

Assurance Activity

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

Guidance

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

Tests

The evaluator shall send a Client Hello requesting a connection with version SSL 1.0 and verify that the server denies the connection. The evaluator shall repeat this test with SSL 2.0, SSL 3.0, TLS 1.0, and any selected TLS versions.

FCS_TLSS_EXT.1.3 The TSF shall generate key agreement parameters using RSA with key size 2048 bits and [*selection: 3072 bits, 4096 bits, no other size*] and [*selection: over NIST curves [selection: secp256r1, secp384r1] and no other curves; Diffie-Hellman parameters of size 2048 bits and [selection: 3072 bits, no other size]; no other*].

Application Note: *If the ST lists a DHE or ECDHE ciphersuite in FCS_TLSS_EXT.1.1, the ST must include the Diffie-Hellman or NIST curves selection in the requirement. FMT_SMF.1*

requires the configuration of the key agreement parameters in order to establish the security strength of the TLS connection.

Assurance Activity

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

Guidance

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

Test

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

FDP_ITT.1 Basic Internal Transfer Protection

FDP_ITT.1.1 The TSF shall prevent the [*disclosure, modification*] of user data when it is transmitted between physically separated parts of the TOE **through the use of [*selection, choose at least one of: IPsec, SSH, TLS, TLS/HTTPS*]**.[3]

Application Note: *This requirement ensures all communications between components of a distributed TOE is protected through the use of an encrypted communications channel. The data passed in this trusted communication channel are encrypted as defined by the protocol chosen in the first selection. The ST author chooses the mechanism(s) supported by the TOE, and then ensures the detailed requirements in Annex C corresponding to their selection are copied to the ST if not already present.*

If SSH is selected, the TOE is expected to conform to the Extended Package for Secure Shell.

Assurance Activity

Test

The evaluator shall examine the TSS to determine that the methods and protocols used to protect distributed TOE components are described. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE

[3]To refine this requirement, the phrase "[assignment: access control SFP(s) and/or information flow control SFP(s)] to" was removed and the phrase "through the use of [selection, choose at least one of: IPsec, SSH, TLS, TLS/HTTPS]" was added.

administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

The evaluator shall examine the operational guidance to ensure it contains instructions for establishing the communication paths for each supported method.

The evaluator shall perform the following tests:

- Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) communications method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test 2: The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext.

Further assurance activities are associated with the specific protocols.

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FIA_PSK_EXT.1 Pre-Shared Key Composition

The TOE may need to support pre-shared keys for use in the IPsec protocol (if it does, "Pre-shared Keys" will be selected as a peer authentication method in FCS_IPSEC_EXT.1.13) . There are two types of pre-shared key–-text-based (which are required) and bit-based (which are optional–-supported by the TOE, as specified in the requirements below. The first type is referred to as "text-based pre-shared keys", which refer to pre-shared keys that are entered by users as a string of characters from a standard character set, similar to a password. Such pre-shared keys must be conditioned so that the string of characters is transformed into a string of bits, which is then used as the key.

The second type is referred to as "bit-based pre-shared keys" (for lack of a standard term); this refers to keys that are either generated by the TSF on a command from the administrator, or input in "direct form" by an administrator. "Direct form" means that the input is used directly as the key, with no "conditioning" as was the case for text-based pre-shared keys. An example would be a string of hex digits that represent the bits that comprise the key.

The requirements below mandate that the TOE must support text-based pre-shared keys and optionally support bit-based pre-shared keys, although generation of the bit-based pre-shared keys may be done either by the TOE or in the operational environment.

FIA_PSK_EXT.1.1 The TSF shall be able to use pre-shared keys for IPsec.

FIA_PSK_EXT.1.2 The TSF shall be able to accept text-based pre-shared keys that:

- are 22 characters and [*selection: [assignment: other supported lengths], no other lengths*];

- composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”).

FIA_PSK_EXT.1.3 The TSF shall condition the text-based pre-shared keys by using [*selection: SHA-1, SHA-256, SHA-512, [assignment: method of conditioning text string]*] and be able to [*selection: use no other pre-shared keys; accept bit-based pre-shared keys; generate bit-based pre-shared keys using the random bit generator specified in FCS_RBG_EXT.1*].

Application Note: *For the length of the text-based pre-shared keys, a common length (22 characters) is required to help promote interoperability. If other lengths are supported they should be listed in the assignment; this assignment can also specify a range of values (e.g., "lengths from 5 to 55 characters") as well.*

In the second selection for FIA_PSK_EXT.1.3, the ST author fills in the method by which the text string entered by the administrator is “conditioned” into the bit string used as the key. This can be done by using one of the specified hash functions, or some other method through the assignment statement. If “bit-based pre-shared keys” is selected, the ST author specifies whether the TSF merely accepts bit-based pre-shared keys, or is capable of generating them. If it generates them, the requirement specified that they must be generated using the RBG specified by the requirements. If the use of bit-based pre-shared keys is not supported, the ST author chooses “use no other pre-shared keys”.

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure that it states that text-based pre-shared keys of 22 characters are supported, and that the TSS states the conditioning that takes place to transform the text-based pre-shared key from the key sequence entered by the user (e.g., ASCII representation) to the bit string used by IPsec, and that this conditioning is consistent with the first selection in the FIA_PSK_EXT.1.3 requirement. If the assignment is used to specify conditioning, the evaluator will confirm that the TSS describes this conditioning.

Guidance

The evaluator shall examine the operational guidance to determine that it provides guidance on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the merits of shorter or longer pre-shared keys. The guidance must specify the allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2.

If “bit-based pre-shared keys” is selected, the evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both). The evaluator shall also examine the TSS to

ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1.

Test

The evaluator shall perform the following tests:

- Test 1: The evaluator shall compose at least 15 pre-shared keys of 22 characters that cover all allowed characters in various combinations that conform to the operational guidance, and demonstrates that a successful protocol negotiation can be performed with each key.
- Test 2 [conditional]: If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length; the maximum length; and an invalid length. The minimum and maximum length tests should be successful, and the invalid length must be rejected by the TOE.
- Test 3 [conditional]: If the TOE supports bit-based pre-shared keys but does not generate such keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.
- Test 4 [conditional]: If the TOE supports bit-based pre-shared keys and does generate such keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_ITT.1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1 The TSF shall protect TSF data from [*disclosure, modification*] when it is transmitted between separate parts of the TOE **through the use of [*selection, choose at least one of: IPsec, SSH, TLS, TLS/HTTPS*]**.

Application Note: *This requirement ensures all communications between components of a distributed TOE is protected through the use of an encrypted communications channel. The data passed in this trusted communication channel are encrypted as defined the protocol chosen in the first selection. The ST author chooses the*

mechanism(s) supported by the TOE, and then ensures the detailed requirements in Annex C corresponding to their selection are copied to the ST if not already present.

If SSH is selected, the TOE is expected to conform to the Extended Package for Secure Shell.

Assurance Activity

TSS

The evaluator shall examine the TSS to determine that the methods and protocols used to protect distributed TOE components are described. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

Guidance

The evaluator shall examine the operational guidance to ensure it contains instructions for establishing the communication paths for each supported method.

Test

The evaluator shall perform the following tests:

- Test 1: The evaluators shall ensure that communications using each specified (in the operational guidance) communications method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
- Test 2: The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext.

Further assurance activities are associated with the specific protocols.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

B.8 Key Protection

Depending on the dependence of cryptographic support, various mechanisms to provide protection to secret and private keys are acceptable.

In FCS_CKM_EXT.1(1) and FCS_CKM_EXT.1(3) there are selections that include key encryption or combinations using KEKs. If any of the SFRs include these selections, entropy calculations are required to show that keys are adequately protected.

FCS_CKM_EXT.1(2) Extended: Key Generation Key Encryption Keys

FCS_CKM_EXT.1.1(2) The TSF shall be able to generate [*selection: asymmetric KEKs of [assignment: security strength greater than or equal to 112 bits*] security strength in accordance with FCS_CKM.1(1), [*selection: size 128-bit, 256-bit*] symmetric KEKS from

[selection:

- *an RBG that meets this profile (as specified in FCS_RBG_EXT.1),*
- *combined from KEKs in a way that preserves the effective entropy of each factor by [selection:*
  - *using an XOR operation,*
  - *concatenating the keys and using a key derivation function (KDF) in accordance with SP 800-108,*
  - *encrypting one key with another in accordance with FCS_COP.1(1) and using modes [selection: AES-CCM, AES-GCM, AES Key Wrap, AES Key Wrap with Padding]]].*

Application Note: *There are three major types of keys: asymmetric keys used by the TSF for signing, establishing secure channels, and TOE integrity, data encryption keys (DEKs) and key encryption keys (KEKs). The TSF may optionally generate subscriber keys. KEKs are used to protect any of these keys. When KEKs protect other keys they form a key hierarchy. When key hierarchies are used to protect keys generated via a mechanism than a validated RGB in accordance with FCS_RBG_EXT.1, FCS_CKM_EXT.7 in Annex C must be included.*

This requirement addresses the generation of KEKs used to protect other keys but not used to archive those keys.

The ST author can select asymmetric or symmetric KEKs (or both). If asymmetric KEKs are selected, the security strength corresponding to the modulus (per FCS_CKM.1(1) will be in assigned in the requirement in the ST. If symmetric generation is chosen, then the size of the symmetric key is as selected, and the method or methods of generating the symmetric KEKs also will need to be selected.

For the generation of symmetric KEKs, if any option but the RBG option is selected, FCS_CKM_EXT.7 in Annex C must be included.

Assurance Activity

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

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

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

FCS_CKM_EXT.7 Symmetric Key Generation for KEKs

FCS_CKM_EXT.7.1 The [*selection: TSF, 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 [selection: by a RBG in accordance with FCS_RBG_EXT.1., according to FCS_CKM.1.1(1)]

Application Note: *Either asymmetric or symmetric keys are allowed; the ST author makes the selection appropriate for the device. Symmetric keys must be of size 128 or 256 bits in order to correspond with FCS_COP.1(1). Asymmetric keys may be of any strength corresponding to FCS_CKM.1(1).*

The lack of a public/documented API for importing or exporting, when a private/undocumented API exists, is not sufficient to meet this requirement.

When TSF is selected in FCS_CKM_EXT.7.1, the RBG used to generate a REK may be a RBG native to a hardware key container that is within the TOE boundary or may be generated using an off-device RBG during manufacturing. If generated by an off-device RBG during manufacturing, the device manufacturer shall not be able to access a REK after the manufacturing process has been completed. If a hardware component in the Operational Environment stores the REK, the RBG may be resident in the component where the REK is stored, or in a separate component. The assurance activities for these cases differ.

Assurance Activity

The evaluator shall examine the TSS to determine that when a REK is supported by the TSF, the TSS includes a description of the protection

provided by the TSF for a REK, and that the TSS includes a description of the method of generation of a REK.

The evaluator shall verify that the description of the protection of a REK describes how any reading, import, and export of that REK is prevented. The evaluator shall verify that the TSS describes how encryption/decryption actions are isolated so as to prevent applications and system-level processes from reading the REK while allowing encryption/decryption by the key.

REK generated by the TOE:

If a REK is generated by the TOE, the TSS shall include a description of the generation mechanism including what triggers a generation, how the functionality described by FCS_RBG_EXT.1 is invoked, and whether a separate instance of the RBG is used for REK(s).

REK generated by an off-device RBG during TOE manufacturing:

If a TOE supported REK is generated by an off-device RBG during manufacturing, the TSS shall include evidence that the RBG used meets FCS_RBG_EXT.1.2. . In addition, the TSS shall describe the manufacturing process that prevents access to REKs.

If a REK is generated by a component of the OE, the evaluator shall confirm that the TSS describes the interface to the component, and verifies (via source code of the TSF) that the TSF calls the validated API of indicated component. If another component of the OE is used to generate the REK, that interface shall also be described.

Justification

The use of asymmetric keys in a key hierarchy had not previously been considered by the authors of the CA PP. An asymmetric encryption scheme can provide similar protection of keys as a symmetric encryption scheme.

FCS_CKM_EXT.8 Key Hierarchy Entropy

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

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

FCS_CKM_EXT.8.3 Key entropy for DEKs shall be [*selection: 128, 256*] bits in accordance with the sensitivity of the data encrypted.

Application Note: *KEKs may form key hierarchies, each rooted in a root encryption key (REK); a REK is considered a KEK. DEKs are used to protect data (e.g., subscriber PII). KEKs are used to protect other keys–- DEKs, other KEKs, and other types of keys stored by the user or applications. A REK is a special KEK that uses available hardware protections (e.g., trusted platform module (TPM) or external hardware cryptographic module) and is generated in accordance with FCS_RBG_EXT.1.*

Assurance Activity

TSS

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

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

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

Guidance

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

Test

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

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

FPT_SKY_EXT.2 Key Share Access

FPT_SKY_EXT.2.1 The [*selection: TSF, Operational Environment*] shall ensure that key shares generated in accordance with FCS_CKM_EXT.1(4) are accessible only to privileged users, and that each share is only accessible to a single privileged user as configured by an Administrator.

Application Note: *This SFR shall be included if "key sharing mechanisms in accordance with FCS_CKM_EXT.1(3), FCS_CKM_EXT.1(4), FCS_CKM_EXT.7, and FPT_SKY_EXT.2" is selected in FPT_SKY_EXT.1.1.*

Assurance Activity

TSS

The Evaluator shall review the user guidance and observe that instructions on how to establish key shares is provided.

Guidance

The evaluator shall assume the role of Administrator and attempt to establish two key shares for the same user and observe that the operation fails.

Test

The evaluator shall then establish two key shares for two different users as instructed in user guidance. The evaluator shall assuming the role of one of the users, attempt to access the share of the other, and observe that the operation fails.

The evaluator shall follow the user guidance to perform key export, taking on the role of each user, and observe that the key is successfully exported. The evaluator shall establish the user roles in a new system, transfer the key shares to the new system, assume each of the roles, recover the previously exported key, and issue a certificate under the new system, observing that the certificate is issued by properly recovered CA.

B.9 Auditable Events

For each of the selection-based requirements claimed by the TOE, the ST author shall include the associated auditable events to the claims made in FAU_GEN.1 and ensure that they are correctly generated as part of testing.

Table 6 – Auditable Events for Selection-Based Requirements

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_EXT.1	None.	None.	N/A	
FAU_GEN.1	None.	None.	N/A	
FAU_GEN.2	None.	None.	N/A	
FAU_SAR.1	None.	None.	N/A	
FAU_SAR.3	None.	None.	N/A	
FAU_SEL.1	All	None.	Normal	

	modifications to the audit configuration that occur while the audit collection functions are operating.			
FAU_STG.1(1)	None	None.	N/A	
FAU_STG.1(2)	None	None	N/A	
FAU_STG.4	None.	None.	N/A	
FAU_STG_EXT.1	None.	None.	N/A	
FAU_STG_EXT.2	None,	None.	N/A	
FCO_NRR_EXT.2	None.	None.	N/A	
FCS_CKM.1(1)	All occurrences of key generation for TOE related functions.	Success: public key generated	Normal	
FCS_CKM.1(2)	All occurrences of key generation for TOE related functions.	Success: public key generated	Normal	
FCS_CKM_EXT.1(1)	None.	None.	N/A	
FCS_CKM_EXT.1(2)	None.	None.	N/A	
FCS_CKM_EXT.1(3)	None	None.	N/A	
FCS_CKM_EXT.1(4)	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	
FCS_CKM_EXT.5	Detection of integrity violation for stored TSF data.	None.	Normal	
FCS_CKM_EXT.6	All key archival actions.	None.	Extended	
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	
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	
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA. Establishment/ Termination of an IPsec SA.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.	Normal	
FCS_RBG_EXT.1	None.	None.	N/A	
FCS_SSHC_EXT.1	Failure to establish an SSH session. Establishment/ Termination of an SSH session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and	Normal	

		failures.		
FCS_SSHS_EXT.1	Failure to establish an SSH session. Establishment/ Termination of an SSH session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.	Normal	
FCS_TLSC_EXT.1	Failure to establish a TLS Session. Establishment/ Termination of a TLS session.	Reason for failure. None	Normal	
FCS_TLSS_EXT.1	Failure to establish a TLS Session. Establishment/ Termination of a TLS session.	Reason for failure. None	Normal	
FDP_CRL_EXT.1	Failure to generate CRL.	None.	Normal	
FDP_ITT.1	None.	None.	N/A	
FDP_OCSP_EXT.1	Failure to generate certificate status information.	None.	Extended	
FDP_SDP_EXT.1	None.	None.	N/A	
FIA_AFL.1	The reaching of the threshold for the unsuccessful authentication attempts. The action taken.	None.	Normal	

	The re-enablement of disabled non-administrative accounts.			
FIA_CMC_EXT.1	CMC requests (generated or received) containing certificate requests or revocation requests. CMC responses issued.	Identifiers for all entities authenticating the request, including the entity providing client authentication for the CMC transport (if any). The submitted request. Any signed response.	Extended	
FIA_EST_EXT.1	EST requests (generated or received) containing certificate requests or revocation requests. EST responses issued.	Identifiers for all entities authenticating the request, including the entity providing client authentication for the EST transport (if any). The submitted request. Any signed response.	Extended	
FIA_PMG_EXT.1	None.	None.	N/A	
FIA_PSK_EXT.1	None.	None.	N/A	
FIA_UAU.7	None.	None.	N/A	
FPT_APW_EXT.1	None.	None.	N/A	
FPT_ITT.1	None.	None.	N/A	
FPT_SKY_EXT.2	Access control violations for users involved in key share establishment or control	None	Extended	
FPT_TST_EXT.2	Execution of this set of TSF integrity tests. Detected	For integrity violations, the identity of the object that caused the integrity violation.	Normal	

	integrity violations.			

C. Objective Requirements

As indicated in the introduction to this PP, the baseline requirements (those that must be performed by the TOE or its underlying platform) are contained in the body of this PP. There are additional requirements that specify security functionality that is desirable and these requirements are contained in this Annex. It is expected that these requirements will transition from objective requirements to baseline requirements in future versions of this PP.

At any time these may be included in the ST such that the TOE is still conformant to this PP.

C.1 Controlled Export

FCS_KSH_EXT.1 Key Sharing

FCS_KSH_EXT.1.1 The TSF shall [*selection: support, interface with the operational environment to support*] split knowledge procedures to enforce two-party control for the export of CA signing keys [*selection: no other data, [assignment: critical data or keys]*] necessary to resume CA functionality after TSF failure using key sharing mechanisms in accordance with FCS_CKM_EXT.1.1(3), FCS_CKM_EXT.1.2(3), FCS_CKM_EXT.7.1 and FPT_SKY_EXT.1.1(2).

Application Note: *This SFR, which mandates the use of key sharing to control the export of CA signing keys, is intended to replace FPT_SKY_EXT.1 in future versions of this PP.*

Assurance Activity

TSS

The evaluator shall examine the TSS to ensure it describes the restrictions placed on key shares generated in accordance with FCS_CKM_EXT.1(4) in accordance with this requirement.

Guidance

The evaluator shall examine the AGD guidance to ensure it contains instructions for configuring the TOE or Operational Environment to restrict access to the shares and limit each one to a single privileged user.

Test

The evaluator shall perform the following test:

- Test 1: The evaluator shall generate key shares that require two persons. The evaluator shall assume a single role and shall verify that access to the assigned share is possible but reconstitution of the original key is not. The evaluator shall then assume a second role and assign a key share to them, then verify that their actions together result in a reconstituted key.

Note that in order to perform this testing, it is acceptable to violate the operational guidance so that the same evaluator is simultaneously accessing

the TSF as two separate identities. Alternatively, this test can be performed by two testers.

Equivalency

Testing of the TOE may be performed on a subset of the platforms listed in the TOE's ST. Justification must be provided for those platforms that were excluded from testing.

D. Entropy Documentation and Assessment

The documentation of the entropy source should be detailed enough that, after reading, the evaluator will thoroughly understand the entropy source and why it can be relied upon to provide entropy. This documentation should include multiple detailed sections: design description, entropy justification, operating conditions, and health testing. This documentation is not required to be part of the TSS.

Design Description

Documentation shall include the design of the entropy source as a whole, including the interaction of all entropy source components. It will describe the operation of the entropy source to include how it works, how entropy is produced, and how unprocessed (raw) data can be obtained from within the entropy source for testing purposes. The documentation should walk through the entropy source design indicating where the random comes from, where it is passed next, any post-processing of the raw outputs (hash, XOR, etc.), if/where it is stored, and finally, how it is output from the entropy source. Any conditions placed on the process (e.g., blocking) should also be described in the entropy source design. Diagrams and examples are encouraged.

This design must also include a description of the content of the security boundary of the entropy source and a description of how the security boundary ensures that an adversary outside the boundary cannot affect the entropy rate.

Entropy Justification

There should be a technical argument for where the unpredictability in the source comes from and why there is confidence in the entropy source exhibiting probabilistic behavior (an explanation of the probability distribution and justification for that distribution given the particular source is one way to describe this). This argument will include a description of the expected entropy rate and explain how you ensure that sufficient entropy is going into the TOE randomizer seeding process. This discussion will be part of a justification for why the entropy source can be relied upon to produce bits with entropy.

Operating Conditions

Documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. It will clearly describe the measures that have been taken in the system design to ensure the entropy source continues to operate under those conditions. Similarly, documentation shall describe the conditions under which the entropy source is known to malfunction or become inconsistent. Methods used to detect failure or degradation of the source shall be included.

Health Testing

More specifically, all entropy source health tests and their rationale will be documented. This will include a description of the health tests, the rate and conditions under which each health test is performed (e.g., at startup, continuously, or on-demand), the expected results for each health test, and rationale indicating why each test is believed to be appropriate for detecting one or more failures in the entropy source.

E. References

Identifier	Title
[CC]	Common Criteria for Information Technology Security Evaluation – • Part 1: Introduction and General Model, CCMB-2012-09-001, Version 3.1 Revision 4, September 2012 [CC1] • Part 2: Security Functional Components, CCMB-2012-09-002, Version 3.1 Revision 4, September 2012 [CC2] • Part 3: Security Assurance Components, CCMB-2012-09-003, Version 3.1 Revision 4, September 2012 [CC3]
[CEM]	Common Methodology for Information Technology Security Evaluation, Version 3.1 Revision 4, September 2012
[IR7924]	Second Draft NIST IR 7924, Reference Certificate Policy, May 2014

F. 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